INAUGURAL-DISSERTATION

zur

Erlangung der Doktorwürde

der

Gesamtfakultät für Mathematik, Ingenieur- und Naturwissenschaften

der

Ruprecht-Karls-Universität

Heidelberg

vorgelegt von

Vinod Varma, Hridya, M.Sc.
aus Palakkad, Indien

Tag der mündlichen Prüfung: .....

Numerical Methods for Parameter Estimation in Dynamical Systems Using Measurements at Bifurcation Points

Betreuerin: Prof. Dr. Ekaterina Kostina

Abstract

Mathematical models play a crucial role in understanding the complex dynamics of real-world systems. One of the main challenges in mathematical modeling is the estimation of model parameters ensuring that model predictions align well with observed data. Traditional approaches to parameter estimation typically rely on measurements of observables over multiple points in time. However, obtaining such data can be challenging or even infeasible for many experimental systems, particularly those that operate on a fast timescale. For such experimental systems, we propose an alternative approach to parameter estimation that uses the values of applied external controls at bifurcation points, instead of time-series data to calibrate the mathematical models. Bifurcations are, in fact, a valuable source of information, particularly in systems that exhibit bistability, hysteresis, and oscillations.

Therefore, in this thesis, we formulate a constrained nonlinear least-squares problem to estimate model parameters by fitting the measured control values at bifurcation points to the corresponding theoretical predictions of the model. We solve this optimization problem using the generalized Gauss-Newton method with efficient structure-exploitation. To ensure reliable convergence, we combine optimization solvers with numerical continuation methods to create a robust numerical strategy for generating initial guesses for the optimization variables. Furthermore, we implement this parameter estimation framework in an open-source software package called `bifit`. This software enables researchers to easily apply, adapt, and extend the methods we develop in this thesis. Using `bifit`, we also demonstrate the effectiveness of our approach through four case studies across various fields. Finally, we adapt the standard optimal experimental design approach to our bifurcation-based framework, enabling researchers to strategically select new measurement points to minimize parameter uncertainty.

Overall, this thesis provides a new perspective on the kind of data that can be used for model calibration and lays the groundwork for further advancements in parameter estimation and experimental design using bifurcation points.

Zusammenfassung

Mathematische Modelle sind zentral für das Verständnis komplexer dynamischer Systeme. Eine große Herausforderung besteht dabei darin, die Modellparameter so zu bestimmen, dass die Vorhersagen mit experimentellen Daten übereinstimmen. Klassische Ansätze nutzen dafür Zeitreihendaten, doch solche Daten sind in vielen Experimenten schwer oder gar nicht zu erfassen – insbesondere bei sehr schnellen Prozessen.

Für solche Fälle schlagen wir einen alternativen Ansatz vor: Anstelle von Zeitreihen verwenden wir die Werte externer Steuergrößen an Bifurkationspunkten, um die Parameter mathematischer Modelle zu schätzen. Diese Werte liefern besonders in Systemen mit Bistabilität, Hysterese oder Oszillationen wertvolle Informationen.

Dazu formulieren wir ein nichtlineares Optimierungsproblem, das mithilfe des verallgemeinerten Gauss-Newton-Verfahrens effizient gelöst wird. Zur Generierung zuverlässiger Startwerte für die Optimierungsvariablen entwickeln wir eine robuste Strategie, die auf numerischen Fortsetzungsmethoden basiert. Wir haben unsere numerische Methode im Open-Source-Paket `bifit` implementiert, um eine einfache Anwendung und Erweiterung zu ermöglichen. Anhand von vier Fallstudien zeigen wir die Leistungsfähigkeit unseres Verfahrens. Außerdem übertragen wir Konzepte der optimalen Versuchsplanung auf unser bifurkationsbasiertes Framework, um weitere Messpunkte gezielt zur Verbesserung der Qualität der geschätzten Parameter auszuwählen.

Insgesamt bietet diese Arbeit einen neuen Zugang zur Parameterschätzung und optimalen Versuchsplanung auf Basis von Bifurkationsdaten und ebnet den Weg für weiterführende Entwicklungen in diesem Bereich.

Acknowledgements

I would like to take this opportunity to express my deepest gratitude to everyone who helped make this thesis possible.

Firstly, I would like to thank my supervisor, Prof. Dr. Ekaterina Kostina, for giving me the opportunity to work on this exciting topic. Your feedback has been invaluable in shaping my research and I really appreciate for all the support and guidance you have given me through the years.

Next, I would also like to thank Dr. Johannes Schlöder and Dr. Jürgen Gutekunst for the fruitful discussions and insightful inputs that helped guide this work. My sincere gratitude goes also to Michael, Ihno, Julian, and Horsch for proofreading parts of my thesis and providing valuable feedback. Thank you all so much for your time and effort.

In this first half of my research, I had the pleasure of working in collaboration with members of the Interdisciplinary Center for Scientific Computing and the University Hospital Mannheim as part of the SCIDATOS TP5 project to mathematically model the immunoregulatory processes in sepsis. I would like to express my gratitude to the Klaus Tschira Stiftung for funding my research during this period and the entire SCIDATOS TP5 team for the great working atmosphere. In particular, I would like to thank Dr. Maria Vittoria Barbarossa, who was my supervisor in this project, for her support and encouragement. I really appreciate all the time and effort you have invested in me and my work.

Next, I would like to thank all my friends and colleagues from the Numerical Optimization group, Simulation and Optimization group, Numerical Analysis and UQ group, Scientific Computing and Optimization group, and the Board Game group for all the friendly discussions, emotional support, and coffee breaks we have had together. You have all made my time in Heidelberg truly enjoyable and memorable.

My special thanks go also to Herta Fitzer, Ramona Ludwig and Jeannette Walsch for always being there to help me with any administrative questions I had. I would also like to express my gratitude to the Heidelberg Graduate School of Mathematical and Computational Methods for the Sciences (HGSMathComp) team for organizing

many helpful courses and events throughout my PhD journey. Thank you for helping me grow as a researcher and for providing me with the opportunity to connect with so many nice people.

Overall, my PhD journey has been a real rollercoaster ride with its many ups and downs, and I would not have made it this far without the love and support from my family and friends. In particular, I would like to start by expressing my deepest gratitude to my parents, my in-laws and my sister for their unconditional love, support, and encouragement through every step of my journey. You gave me the strength to keep going when things got tough, and I am most grateful for everything you have done for me. Alongside my family, I also have my amazing bunch of friends to thank. I really appreciate you guys for always being there for me — to celebrate all the good times and to help me through all the difficult times. Thank you all so much for being such a wonderful support system.

Last but not least, I would like to thank my husband, Michael, for being my pillar of strength and my biggest cheerleader throughout this journey. None of this would have been possible without you by my side and words cannot express how grateful I am for all your love and support. I am really lucky to have you in my life, and I look forward to the next chapter of our lives together.

This thesis is dedicated to Michael and to my parents. Thank you for everything!

Contents

Introduction

"Mathematical analysis and computer modelling are revealing to us that the shapes and processes we encounter in nature — the way that plants grow, the way that mountains erode or rivers flow, the way that snowflakes or islands achieve their shapes, the way that light plays on a surface, the way the milk folds and spins into your coffee as you stir it, the way that laughter sweeps through a crowd of people — all these things in their seemingly magical complexity can be described by the interaction of mathematical processes that are, if anything, even more magical in their simplicity. Shapes that we think of as random are in fact the products of complex shifting webs of numbers obeying simple rules. The very word "natural" that we have often taken to mean "unstructured" in fact describes shapes and processes that appear so unfathomably complex that we cannot consciously perceive the simple natural laws at work. They can all be described by numbers."

- Douglas Adams, Dirk Gently's Holistic Detective Agency

Mathematical models translate the interactions and dynamics of complex real-world systems into the language of mathematics. How does the zebra get its stripes? How do diseases spread in a population? How does an airplane fly in the sky? Should you confess to your crime or deny it? These are questions that can be studied with the help of mathematical modeling.

There are a wide variety of approaches available for mathematically modeling dynamical systems ranging from difference equations and differential equations to cellular automata and game theory. For any given problem, we can choose an appropriate modeling framework that best suits the nature of the dynamical system. The modeling framework that is of primary interest in our thesis is that of ordinary differential equations (ODEs).

A good mathematical representation of a real-world system can help us understand the mechanism underlying its complex dynamics, forecast future behavior and make informed decisions. In order to obtain an accurate representation, the parameters of the mathematical model need to be calibrated so that its dynamics matches at least the known behavior of the real-world system. This can be done, for example, in the study of an epidemic, by identifying the population of infected and recovered

patients over time and matching these quantities to the corresponding populations in the mathematical model. This process is known as parameter estimation.

One of the most commonly used approaches to parameter estimation is to take measurements at multiple time points and match them to their corresponding model predictions. The parameters that produce model predictions that lie closest to the observed data are then considered the best approximation of the true parameters. However, in some experimental systems — especially those operating on very fast time scales or those where measurements at multiple time points are impractical or expensive — it may be better to measure a different type of data.

An approach, which has received relatively little attention so far, is parameter estimation using measurements of bifurcation points or switching points. A bifurcation is said to occur when a small, continuous change in a system parameter leads to a sudden change in the qualitative behavior of the dynamical system. The parameter value at which this transition takes place is known as a bifurcation point. Bifurcations have been experimentally observed in a variety of real-world processes that exhibit bistability, hysteresis and oscillatory behavior.

Therefore, in this thesis, we explore how measurements of applied external controls at bifurcation points can be used to estimate the parameters of a mathematical model. Furthermore, we also discuss how the quality of the parameter estimates can be improved through optimal experimental design by strategically selecting new measurement points. To demonstrate the practical applicability of our parameter estimation approach, we present a series of numerical case studies and provide an open-source implementation of our parameter estimation framework.

Contributions of this thesis

The main contribution of this thesis is the development of a novel approach to parameter estimation inspired by and building upon the work of Schlöder and Bock [Sch87; SB83]. Specifically, we formulate a parameter estimation and optimal experimental design framework for systems exhibiting bistability through saddle-node bifurcations and oscillations through Hopf bifurcations. To numerically solve the parameter estimation problem, we develop a robust solution strategy with a structured approach to generating initial guesses for the optimization variables. We also present various fields of application where our parameter estimation framework can be effectively used and demonstrate this through several numerical case studies using synthetic data. Finally, we provide an open-source implementation of our

parameter estimation framework as a Python package called `bifit` to allow other researchers to easily apply our methods to their parameter estimation problems. These contributions are explained in more detail below:

A novel parameter estimation and optimal experimental design framework using measurements of bifurcation points

Bistable behavior and periodic oscillations have been observed in various experimental systems in biochemistry [DSH99; Ric+94; AHF90; Sem+16; Ozb+04], ecology [Jos+73; Fus+00; Cos+97; Den+97], neuroscience [Pat+99; Bin+06; LPS10], semiconductor lasers [Wie+03] and chemical engineering [Eln+06; GL87; SB87; HL87; ZBH01] under certain controlled environments. Motivated by these applications, in this thesis, we introduce a novel approach to parameter estimation — one that exploits the qualitative changes in these experimental systems to estimate the unknown parameters of their mathematical models.

This idea was inspired by the works of Schlöder and Bock [Sch87; SB83], who developed an efficient, structure-exploiting numerical method for solving high-dimensional parameter estimation problems and demonstrated their method with an unusual parameter estimation problem in one of their case studies. In order to estimate the parameters of a mathematical model of the Belousov-Zhabotinsky reaction using their numerical scheme, Schlöder and Bock used measurements of bifurcation points produced by Geiseler and Bar-Eli [GB81]. To the best of our knowledge, this was the earliest instance of bifurcation points being used for parameter estimation. However, since their primary focus was on exploiting the multi-experiment structure of parameter estimation problems to efficiently solve the high-dimensional optimization problem, the potential for establishing a broader framework for parameter estimation using bifurcation point measurements was not realized. While other independent studies have also explored the use of bifurcation points for parameter estimation, particularly in the field of chemical engineering [SB87; HL87; CT84], to the best of our knowledge, the idea has so far not been developed into a general framework. Therefore, one of the key contributions of this thesis is the development of a general parameter estimation framework that uses measurements of bifurcation points to calibrate mathematical models. Additionally, we also formulate an optimal experimental design problem to identify new values of the external controls at which bifurcation points can be measured to potentially improve the quality of the parameter estimates.

A robust numerical approach to generating initial guesses for the optimization variables in parameter estimation problems that use bifurcation point measurements

In order to estimate the parameters of a mathematical model using bifurcation point measurements, we need to solve a large-scale nonlinear least squares problem that combines measurements from multiple experiments. Since we use the generalized Gauss-Newton method [Boc87] with efficient structure-exploitation [Sch87] to solve this optimization problem, it is crucial to use reliable initial guesses for the optimization variables. However, this can be challenging, as we will see in Chapter 4, because the optimization variables include not just the unknown parameters of the mathematical model as in standard parameter estimation problems, but also the steady state and external control values corresponding to each measured bifurcation point. This means that the initial guesses for these additional optimization variables need to correspond to bifurcation points of the mathematical model and lie close to the measurements. Therefore, in this thesis, we develop a structured approach to generate reliable initial guesses that satisfy these requirements.

Our proposed numerical scheme is an extension and generalization of the one used for the Belousov-Zhabotinsky example by Schlöder and Bock [Sch87; SB83]. In particular, we develop a robust numerical strategy to generate initial guesses for the optimization variables in parameter estimation problems that use measurements of saddle-node or Hopf bifurcations. This involves identifying a steady state solution, computing a one-parameter bifurcation diagram, detecting the relevant bifurcation point and computing a two-parameter curve of bifurcation points that lie close to the measured bifurcation points. For computing the bifurcation diagrams, we consider two different numerical continuation methods — the standard pseudo-arclength continuation method [Kel86] and the deflated continuation method [FBB16] — with an adaptive step size strategy. This structured approach to generating initial guesses is crucial for the convergence of the optimization problem and the quality of the parameter estimates.

Numerical investigation of various case studies

Our parameter estimation framework, which uses bifurcation point measurements, is motivated by experimentally controllable systems across a wide range of applications such as biochemistry [DSH99; Ric+94; AHF90; Sem+16; Ozb+04], ecology [Jos+73; Fus+00; Cos+97; Den+97], neuroscience [Pat+99; Bin+06; LPS10],

semiconductor lasers [Wie+03] and chemical engineering [Eln+06; GL87; SB87; HL87; ZBH01]. Therefore, to illustrate the practical applicability of our approach, we numerically investigate some of these experimental systems as case studies and explore how their mathematical models can be calibrated using measurements of bifurcation points.

Specifically, we demonstrate our parameter estimation strategy using four case studies: two bistable systems, and two oscillatory systems. For bistable behavior, we examine a semiconductor laser system [WKL99] and an autocatalytic chemical reaction network from synthetic biology [Sem+16], both of which exhibit saddle-node bifurcations. For oscillatory dynamics, we investigate a peroxidase-oxidase reaction system [DOP79] and a predator-prey system [Fus+00], both of which exhibit Hopf bifurcations. In each case study, we generate artificial measurements by adding normally-distributed noise to a set of numerically computed bifurcation points and use our parameter estimation framework to infer the model parameters from these artificial measurements. To assess the robustness of our approach, we also examine the empirical convergence region of the parameter estimation problem by solving the problem using multiple random initial guesses for the model parameters.

Open-source implementation of our parameter estimation framework with *a posteriori* sensitivity analysis

In this thesis, we also develop and publish an open-source Python package called `bifit`, which provides a robust implementation of our parameter estimation framework along with *a posteriori* sensitivity analysis. This package allows users to easily provide a mathematical model, load measurement data and specify initial guesses to solve their multi-experiment parameter estimation problem.

A key feature of our software is its automated workflow. This means that, given the initial guesses for the model parameters, the solver can automatically compute a one-parameter bifurcation diagram, detect bifurcation points, and trace a curve of bifurcation points close to the experimentally measured points to generate reliable initial guesses for the parameter estimation problem. To achieve this, we have implemented both the pseudo-arclength method [Kel86] and the deflated continuation method [FBB16] with adaptive step size strategy for computing the bifurcation diagrams. Moreover, in order to solve the small optimization problems in the initial guess generation procedure, our software integrates standard nonlinear optimization solvers from the `scipy` package [Vir+20], in addition to our own implementation of

a generalized Gauss-Newton solver with an active-set strategy [Boc87] for nonlinear least squares problems. To efficiently handle large-scale parameter estimation problems that combine data from multiple experiments, our software uses the generalized Gauss-Newton method, with two options for solving the linear least squares problem at each step: our implementation of the structure-exploiting numerical scheme developed by Schlöder and Bock [Sch87; SB83] and the operator-splitting quadratic program (OSQP) solver provided by the `osqp` package [Ste+20].

Our software is designed to be user-friendly with extensive documentation, several examples, and minimal external dependencies. We believe that this can enable researchers from various fields to apply our framework to their own parameter estimation problems, even with limited prior experience in numerical optimization.

Thesis overview

This thesis is organized into three parts, comprising a total of ten chapters.

- **Part I** lays the theoretical groundwork, covering bifurcation theory, numerical bifurcation analysis, and nonlinear optimization.
- **Part II** introduces our novel parameter estimation framework including a discussion of *a posteriori* sensitivity analysis and optimal experimental design for bifurcation point measurements.
- **Part III** explores various applications, presents detailed case studies, and includes an overview of the open-source software package we have developed.

Part I: Theoretical Foundations

Chapter 1 provides an introduction to dynamical systems and bifurcation theory, including key definitions, linear stability analysis, and an overview of common one-parameter bifurcations. Chapter 2 covers numerical continuation methods for generating bifurcation diagrams, with a focus on the pseudo-arclength continuation method and the deflated continuation method, including our proposed modifications to these methods. Chapter 3 introduces nonlinear optimization, discussing fundamental concepts, numerical solvers for constrained optimization, and the structure-exploiting scheme proposed by Schlöder and Bock [Sch87; SB83] for parameter estimation problems.

Part II: Parameter Estimation and Optimal Experimental Design

Chapter 4 presents our parameter estimation framework for measurements of bifurcation points. This includes the motivation for our approach, a discussion of related works, formulation of the optimization problem and a detailed discussion of the numerical solution strategy. Chapter 5 focuses on *a posteriori* sensitivity analysis, covering the statistical motivation for the sensitivity analysis, formulation of the covariance matrix and definition of the confidence intervals for estimated parameters. Chapter 6 introduces an optimal experimental design approach for identifying new bifurcation points that can be measured to improve the uncertainty in the estimated parameters.

Part III: Numerical Implementation and Results

Chapter 7 explores practical applications of our parameter estimation framework in experimental systems exhibiting bistability and oscillations. Chapter 8 examines four case studies — including two bistable systems and two oscillatory systems — where we apply our framework to estimate model parameters from noisy data. Chapter 9 introduces our open-source software package, detailing its structure, implementation and how the software can be used. Finally, in Chapter 10, we conclude our discussion with a summary of our findings and an outlook on future research directions.

Part I

Theoretical Foundations

1 Fundamentals of Bifurcation Theory

In this chapter, we introduce some fundamental concepts from the qualitative theory of ordinary differential equations and bifurcations. We begin by learning about continuous-time dynamical systems and their basic components. We then study how the local stability of their steady state solutions can be analyzed to find bifurcations. Finally, we conclude the chapter with a brief overview of some of the most well-known one-parameter bifurcations of steady states.

For a more in-depth treatment of these topics, we refer the reader to standard textbooks on dynamical systems and bifurcation theory by Kuznetsov [Kuz23], Wiggins [Wig03] or Guckenheimer and Holmes [GH13]. The material presented in this chapter closely follows the book by Kuznetsov [Kuz23].

1.1 Introduction to dynamical systems

Consider the following autonomous system of ordinary differential equations that describes the time evolution of a state variable x:

$$\dot{x}(t) = f(x), \tag{1.1a}$$

$$x(t_0) = x_0. \tag{1.1b}$$

Here, the states $x \in \mathbb{R}^n$ lie in a continuous finite-dimensional state space and evolve continuously in time $t \in \mathbb{R}$. For the purpose of this thesis, we also assume that the right-hand side function $f : \mathbb{R}^n \to \mathbb{R}^n$ is sufficiently smooth, usually $\mathcal{C}^2$ or $\mathcal{C}^3$.

System (1.1) is an example of a continuous-time dynamical system. In order to determine the state $x(t)$ of this dynamical system at a given time t, we can define a map $\phi^t : \mathbb{R}^n \to \mathbb{R}^n$ called the *evolution operator* that transforms an initial state $x_0 \in \mathbb{R}^n$ into some state $x(t) \in \mathbb{R}^n$ at time t as follows:

$$x(t) = \phi^t x_0.$$

The family $\{\phi^t\}_{t\in\mathbb{R}}$ of evolution operators of a dynamical system is called its *flow*.

More broadly, we can define a dynamical system as follows:

Definition 1.1.1 : Dynamical System
A triple (X, T, ϕ^t), where X is the state space, T is the time set and $\phi^t : X \to X$ is a family of evolution operators parameterized by $t \in T$ and is called a *dynamical system* if it satisfies the following properties:

- $\phi^0 = \text{id}$, where id is the identity map on X with $\text{id}x = x$ for all $x \in X$,
- $\phi^{t_1+t_2} = \phi^{t_1} \circ \phi^{t_2}$ for all $t_1, t_2 \in T$, where $\circ$ denotes the composition of maps.

However, in this thesis, we will focus our attention on the case where the time set is $\mathbb{R}$ and the state space is a finite-dimensional metric space $\mathbb{R}^n$ as in the dynamical system (1.1). In order to study the solutions of this dynamical system, we can look at its orbits or trajectories in state space.

Definition 1.1.2 : Orbit or Trajectory
An *orbit* or *trajectory* of the dynamical system (1.1) starting at $x_0 \in \mathbb{R}^n$ is defined as an ordered subset of the state space $\mathbb{R}^n$ given by:

$$Or(x_0) = \{x(t) \in \mathbb{R}^n : x(t) = \phi^t x_0, \text{ for all } t \in \mathbb{R} \text{ such that } \phi^t x_0 \text{ defined } \}.$$

This means that an orbit $Or(x_0)$ is the set of all states that can be reached from the initial state x_0 by applying the evolution operator ϕ^t for all possible times $t \in \mathbb{R}$, and is a continuous curve in the state space parameterized by time and oriented in the direction of its increase.

One of the simplest orbits to study is the steady state.

Definition 1.1.3 : Steady State or Equilibrium Point
A point $\bar{x} \in \mathbb{R}^n$ is said to be a *steady state* or *equilibrium point* of the dynamical system (1.1) if $\phi^t \bar{x} = \bar{x}$ for all $t \in \mathbb{R}$. This is equivalent to the condition $f(\bar{x}) = 0$.

This means that an orbit starting at a steady state will remain there forever. Due to its simplicity, steady states are often the first objects of interest when studying a dynamical system's behavior.

Another more complex type of orbit that is also extensively studied is the periodic orbit or limit cycle.

Definition 1.1.4 : Periodic Orbit or Limit Cycle
Points $x(t) \in Or(x_0)$ are said to be in a *periodic orbit* or *limit cycle* if there exists some $T > 0$ such that $\phi^{t+T}x_0 = \phi^t x_0$ for all $t \in \mathbb{R}$. In the dynamical system (1.1), this means that $x(t) = x(t+T)$ for all $t \in \mathbb{R}$. The smallest value of T satisfying this property is called the *period* of the cycle.

This means that a periodic orbit is a closed curve in the state space such that the system starting at a point x_0 will return to the same point after a finite time T.

Both steady states and periodic orbits are part of a broader class of solutions known as invariant sets.

Definition 1.1.5 : Invariant Set
The set of all points $x \in S \subset \mathbb{R}^n$ in the state space of the dynamical system (1.1) is known as an *invariant set* if $x \in S$ implies that $\phi^t x \in S$ for all time $t \in \mathbb{R}$.

This means that any solution that starts within the invariant set S will remain in this set for all time.

Having introduced the basic components of a continuous-time dynamical system, we will now discuss how these tools can be used to study the qualitative behavior of the dynamical system. In general, it is very difficult to analytically study the behavior of a nonlinear dynamical system in its entire state space. Therefore, we will first focus on the local behavior of the system around its invariant sets.

1.2 Linear stability analysis

To study the asymptotic behavior of an invariant set, we need to analyze its stability. Broadly, we can understand an invariant set as being stable if a point starting close to it remains close to it for the remaining time. Conversely, an invariant set is said to be unstable if a point starting close to it moves further away from it over time. We can make these notions of stability more precise using the following definitions from Kuznetsov [Kuz23]:

Definition 1.2.1 : Lyapunov Stability
An invariant set S_0 is called *Lyapunov stable* if, for any sufficiently small neighborhood $U \supset S_0$, there exists a neighborhood $V \supset S_0$ such that $\phi^t x \in U$ for all $x \in V$ and all $t > 0$.

Definition 1.2.2 : Asymptotic Stability
An invariant set S_0 is said to be *asymptotically stable* if there exists a neighborhood $U_0 \supset S_0$ such that $\phi^t x \to S_0$ for all $x \in U_0$ as $t \to \infty$.

In other words, for any Lyapunov stable invariant set, there always exists a neighborhood around it such that the solution never leaves this neighborhood. Asymptotic stability, on the other hand, extends this notion further. An asymptotically stable invariant set has a neighborhood such that all points in this neighborhood will eventually converge to the invariant set itself, given enough time.

One of the simplest invariant sets to study is the steady state. The following theorem by Lyapunov [Lya92] translates the notions of stability that we have defined above to the steady states of the dynamical system (1.1):

Theorem 1.2.3 : Stability of steady states of differential equations
Consider the continuous-time dynamical system (1.1) with a steady state $\bar{x} \in \mathbb{R}^n$ such that $f(\bar{x}) = 0$. Let $Df(\bar{x})$ be the Jacobian matrix of $f(x)$ evaluated at the steady state. Then $\bar{x}$ is stable if all eigenvalues $\lambda_1, \ldots, \lambda_n$ of $Df(\bar{x})$ have negative real parts.

Proof. Consider the solutions of the dynamical system near the steady state $\bar{x}$:

$$x(t) = \bar{x} + \epsilon(t).$$

We can substitute this into the ODE (1.1) and take its first-order Taylor expansion around $\bar{x}$ to get the following linearized system:

$$\begin{aligned} \dot{x}(t) = \dot{\bar{x}} + \dot{\epsilon}(t) &= f(\bar{x}) + Df(\bar{x}) \cdot \epsilon + \mathcal{O}\left(\|\epsilon\|^2\right) \\ x(t_0) = x_0 &= \bar{x} + \epsilon(0) \end{aligned}$$

where Df is the derivative of the right-hand side function f with respect to the states x and $\|\cdot\|$ is any norm on $\mathbb{R}^n$. Since $\dot{\bar{x}} = 0$ and $f(\bar{x}) = 0$, this becomes:

$$\dot{\epsilon}(t) = Df(\bar{x}) \cdot \epsilon(t) + \mathcal{O}\left(\|\epsilon\|^2\right), \tag{1.2a}$$

$$\epsilon(t_0) = x_0 - \bar{x} =: \epsilon_0. \tag{1.2b}$$

We can then compute the solutions of this linear dynamical system to get:

$$\epsilon(t) = e^{Df(\bar{x})t}\epsilon_0.$$

It is now easy to see that the solution $\epsilon(t)$ will converge to zero only if all the eigenvalues of the Jacobian matrix $Df(\bar{x})$ have negative real parts. Conversely, if any of the eigenvalues have a positive real part, the solution will grow exponentially, leading to divergence from the equilibrium. □

Equations (1.2) represent the time-evolution of any small perturbation from the equilibrium point $\bar{x}$ in its local neighborhood. This means that we can study the stability of the steady state by determining how this perturbation evolves over time: does it converge to or diverge away from the equilibrium point? This is the basis of our linear stability analysis.

With this understanding, we can now introduce the fundamental theorem by Philip Hartman and David Grobman [Har60; Har63; Gro59; GH13]:

Theorem 1.2.4 : Hartman-Grobman
Consider the continuous-time dynamical system (1.1). If its Jacobian matrix $Df(\bar{x})$ has no zero or purely imaginary eigenvalues, then there is a homeomorphism $h : U \to \mathbb{R}^n$ defined on some neighborhood U of the steady state $\bar{x}$ that maps the local flow $\{\phi^t\}_{t\in\mathbb{R}}$ of the nonlinear vector field to the local flow $\{e^{Df(\bar{x})t}\}_{t\in\mathbb{R}}$ of its linearization:

$$h(x(t)) = e^{Df(\bar{x})t}h(x_0) \quad \forall x_0 \in U.$$

Proof. See [Gro59] and [Har63], or [Arn92]. □

Therefore, the linearized system provides a reliable approximation of the nonlinear dynamical system near the steady state, as long as none of the eigenvalues of the Jacobian matrix have zero real parts. A steady state that satisfies this condition is called *hyperbolic* or *non-degenerate*. However, if at least one eigenvalue has a zero real part, the Hartman-Grobman theorem no longer applies suggesting that the linearization does not provide enough information to determine the stability of the steady state. In such cases, the system may undergo a bifurcation — a qualitative change in its behavior.

1.3 One-parameter bifurcations of steady states

Before we learn more about bifurcations, we need to first establish a notion of what it means for two dynamical systems to have qualitatively similar behavior.

Definition 1.3.1 : Topological Equivalence
Two dynamical systems $\{\mathbb{R}^n, T, \phi^t\}$ and $\{\mathbb{R}^n, T, \psi^t\}$ are said to be *topologically equivalent*, if there exists a homeomorphism $h : \mathbb{R}^n \to \mathbb{R}^n$ that maps the orbits of one system onto orbits of the other system, preserving the direction of time.

This means that two dynamical systems are qualitatively similar if we can find a continuous transformation that maps the trajectories of one system to the other.

Consider now a dynamical system that depends on a parameter $\alpha \in \mathbb{R}$:

$$\dot{x}(t) = f(x, \alpha), \quad x(t_0) = x_0, \tag{1.3}$$

where $x \in \mathbb{R}^n$ and $\alpha \in \mathbb{R}$. The parameter α can be thought of as a control parameter that can be varied to change the behavior of the system. Then, a bifurcation is said to occur when a small change in the parameter α results in a sudden qualitative change in this system's behavior. More formally, we can define this as follows:

Definition 1.3.2 : Bifurcation Point
A *bifurcation* occurs when a variation in the parameter α of the dynamical system (1.3) produces a topologically nonequivalent phase portrait. The parameter value at which this occurs is then called a *bifurcation point*.

In this thesis, our primary focus will be on bifurcations of steady states. Since the analysis of such bifurcations is often done in the neighborhood of non-hyperbolic or degenerate steady states, these bifurcations are known as *local bifurcations*.

Without loss of generality, let us assume that the dynamical system (1.3) has a non-hyperbolic steady state $x = 0$ at $\alpha = 0$. We can now introduce the center manifold theorem.

Theorem 1.3.3 : Center manifold theorem
Consider a non-hyperbolic steady state $\bar{x} = 0$ of the dynamical system (1.3) at the parameter value $\bar{\alpha} = 0$. Then, there exists a locally defined smooth invariant manifold called the *center manifold* $W^c_{\text{loc}}(0)$ with the same dimension as and tangential to the *center eigenspace* E^c (span of all the eigenvectors with eigenvalues having zero real parts) of the linearized system at $\bar{x} = 0$.

Moreover, there is a neighborhood U of $\bar{x} = 0$ such that, if $\phi^t x \in U$ for all $t \geq 0$ ($t \leq 0$), then $\phi^t x \to W^c_{\text{loc}}(0)$ for $t \to \infty$ ($t \to -\infty$).

Proof. See [Car82] and [Van89]. □

This theorem tells us that the dynamics of a nonlinear dynamical system near the non-hyperbolic steady state can be analyzed by restricting our attention to the dynamics on its center manifold.

The steady state can become non-hyperbolic in one of two ways: either a simple real eigenvalue is zero (*fold bifurcation*) or a pair of simple complex eigenvalues reaches the imaginary axis (*Hopf bifurcation*). We will now look at these two types of bifurcations in more detail.

1.3.1 Saddle-node bifurcation

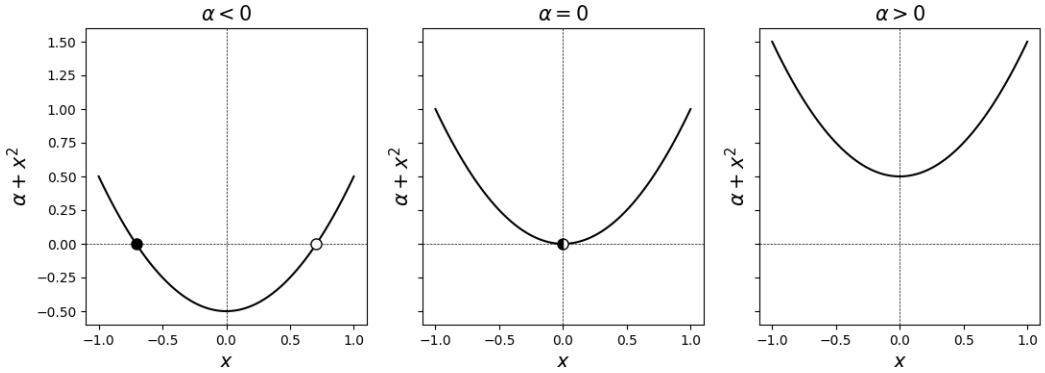


Fig. 1.1.: Visualization of the saddle-node bifurcation of the dynamical system (1.4) as the parameter α is varied (taken in modified form from [Kuz23]). The circles represent steady states, where the filled ones are stable, the empty ones are unstable, and the half-filled ones are saddle points.

Consider the following scalar dynamical system:

$$\dot{x}(t) = f(x, \alpha) = \alpha + x^2, \tag{1.4}$$

where $x \in \mathbb{R}$ and $\alpha \in \mathbb{R}$. It is easy to see here that the origin $(0, 0)$ is a bifurcation point of this system satisfying both the steady state condition $f(0, 0) = 0$ and the zero eigenvalue condition $\partial_x f(0, 0) = 0$. Furthermore, the set of all steady states is a parabola in the α-x plane:

$$\alpha + x^2 = 0 \quad \Longleftrightarrow \quad x = \pm\sqrt{-\alpha}.$$

Therefore, for $\alpha < 0$, this scalar system has two steady states, where one of them is stable and the other unstable. At $\alpha = 0$, the two steady states collide in a non-hyperbolic steady state $x = 0$ and annihilate each other. Then, for $\alpha > 0$, there are

no steady states. This type of bifurcation is called a *saddle-node or fold bifurcation*. We have visualized this bifurcation in Figure 1.1.

More generally, the saddle-node bifurcation of a higher dimensional nonlinear dynamical system (1.3) with a non-hyperbolic steady state $\bar{x} = 0$ at $\bar{\alpha} = 0$ can be studied by restricting our attention to its center manifold. Since the Jacobian matrix at the saddle-node bifurcation point has a simple zero eigenvalue, its center manifold is one-dimensional and locally topologically equivalent to the system (1.4), with a possible change of sign in the nonlinear term. For a more detailed explanation of this equivalence, we refer the reader to the book by Kuznetsov [Kuz23].

1.3.2 Hopf bifurcation

Consider the following planar dynamical system:

$$\dot{x_1}(t) = \alpha x_1 - x_2 - x_1(x_1^2 + x_2^2), \tag{1.5a}$$
$$\dot{x_2}(t) = x_1 + \alpha x_2 - x_2(x_1^2 + x_2^2), \tag{1.5b}$$

where $x_1, x_2 \in \mathbb{R}$ and $\alpha \in \mathbb{R}$. This system has a steady state at the origin $(0, 0)$ for all values of α with the Jacobian matrix $\partial_x f(0, 0)$ given by:

$$\partial_x f(0,0) = \begin{pmatrix} \alpha & -1 \\ 1 & \alpha \end{pmatrix}.$$

This matrix has eigenvalues $\lambda_1 = \alpha + i$ and $\lambda_2 = \alpha - i$. Since a Hopf bifurcation occurs when the Jacobian matrix has a pair of purely imaginary eigenvalues with all the other eigenvalues having non-zero real parts, we can see that this condition is satisfied when $\alpha = 0$.

Let us now transform this dynamical system to make the analysis easier. We introduce a complex variable $z = x_1 + ix_2$ which satisfies the following differential equation:

$$\begin{aligned} \dot{z}(t) = \dot{x_1}(t) + i\dot{x_2}(t) &= \alpha x_1 - x_2 - x_1(x_1^2 + x_2^2) + i\left(x_1 + \alpha x_2 - x_2(x_1^2 + x_2^2)\right), \\ &= \alpha(x_1 + ix_2) + i(x_1 + ix_2) - (x_1^2 + x_2^2)(x_1 + ix_2), \\ &= (\alpha + i)z - |z|^2 z. \end{aligned}$$

Using the representation $z = \rho e^{i\phi}$, we can rewrite this equation as:

$$\dot{z}(t) = \dot{\rho}e^{i\phi} + i\rho\dot{\phi}e^{i\phi} = \rho e^{i\phi}(\alpha + i - \rho^2).$$

This gives the following representation of system 1.5 in polar coordinates:

$$\dot{\rho} = \rho(\alpha - \rho^2), \tag{1.6a}$$
$$\dot{\phi} = 1. \tag{1.6b}$$

The first equation of this system represents the distance from the center and the second equation represents the speed of rotation. In this case, both these equations are decoupled.

From the first equation, we find that there is a steady state $\bar{\phi}_1 = 0$ that exists for all values of α. When $\alpha < 0$, this steady state is linearly stable. At the bifurcation point $\alpha = 0$, the steady state remains stable, but the rate of convergence is not exponential anymore. For $\alpha > 0$, the steady state then becomes linearly unstable. At this point, a new stable steady state $\bar{\rho}_2 = \sqrt{\alpha}$ is born. In the x_1-x_2 plane, this steady state represents a closed orbit or limit cycle with radius $\sqrt{\alpha}$. All orbits starting both inside or outside the cycle (except at the origin) tend to the cycle asymptotically as $t \to \infty$ and the system exhibits periodic oscillations around the origin. This type of bifurcation is called a *Hopf bifurcation*. In particular, this is a supercritical Hopf bifurcation that we have visualized in Figure 1.2.

If we now consider the dynamical system (1.5) with the opposite sign in the nonlinear terms, we get the following system:

$$\dot{x_1}(t) = \alpha x_1 - x_2 + x_1(x_1^2 + x_2^2), \tag{1.7a}$$
$$\dot{x_2}(t) = x_1 + \alpha x_2 + x_2(x_1^2 + x_2^2), \tag{1.7b}$$

Using similar transformations and analysis as before, we can show that this system has a steady state at the origin that exists for all values of α. This steady state has the same stability as the system (1.5) for all $\alpha \neq 0$. At the bifurcation point $\alpha = 0$, the steady state at the origin becomes unstable. Similarly, a limit cycle is born as the parameter α crosses zero from positive to negative values. However, this limit cycle is unstable. This means that all orbits starting inside or outside the cycle tend to diverge away from it as $t \to \infty$. This type of bifurcation is called a *subcritical Hopf bifurcation*, visualized in Figure 1.3.

We can generalize this analysis to higher dimensions using the center manifold theorem. Since the nonlinear dynamical system (1.3) with a non-hyperbolic steady

state $\bar{x} = 0$ at $\bar{\alpha} = 0$ exhibits a Hopf bifurcation when the Jacobian matrix $\partial_x f(0, 0)$ has a pair of purely imaginary eigenvalues, we can reduce our study of its local behavior by looking at its two-dimensional center manifold. This reduced system is locally topologically equivalent to the normal form of a Hopf bifurcation given by (1.4). For a more detailed explanation of normal forms, we refer the reader to the book by Kuznetsov [Kuz23].

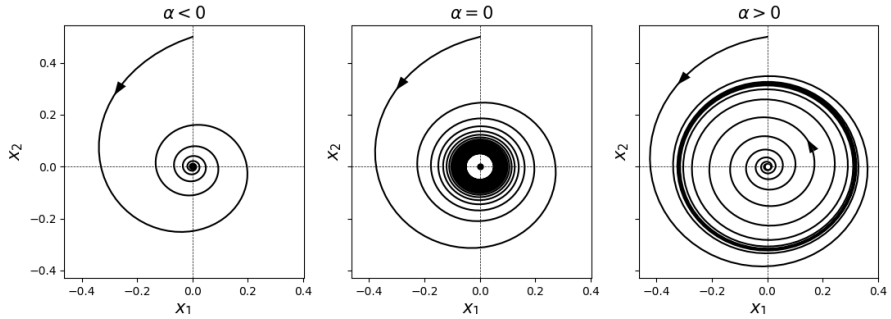


Fig. 1.2.: Visualization of a supercritical Hopf bifurcation of the dynamical system (1.5) as the parameter α is varied (taken in modified form from [Kuz23]). The circles at the origin represent steady states, where the filled ones are stable and the empty ones are unstable. The arrows indicate the direction of the flow.

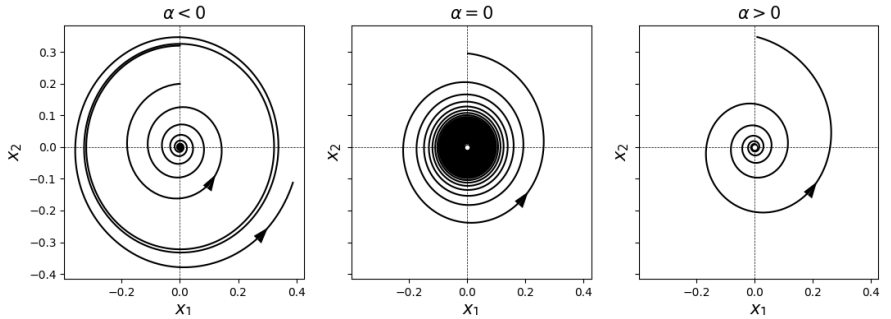


Fig. 1.3.: Visualization of a subcritical Hopf bifurcation of the dynamical system (1.7) as the parameter α is varied (taken in modified form from [Kuz23]). The circles at the origin represent steady states, where the filled ones are stable and the empty ones are unstable. The arrows indicate the direction of the flow.

2 Numerical Bifurcation Analysis

In this chapter, we explore numerical continuation methods for computing bifurcation diagrams. These methods solve systems of parameterized nonlinear equations to obtain a curve of solutions through continuous transformations from a known solution. We begin our discussion by outlining the basic idea of numerical continuation methods before introducing two key approaches for numerical bifurcation analysis: the pseudo-arclength continuation method and the deflated continuation method. In particular, we explain both methods in detail, present our adaptations of these methods and discuss some of their limitations.

Each section includes references for readers interested in a more detailed discussion of the presented numerical methods. For a more in-depth discussion of numerical continuation methods in general, we recommend the book by Allgower and Georg [AG03].

2.1 Motivation

The goal of numerical continuation methods is to find solutions of a system of nonlinear equations

$$f(x, \alpha) = 0$$

where $f : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$ is a sufficiently smooth mapping (usually $\mathcal{C}^2$) and $\alpha \in \mathbb{R}$ is a parameter. We do this by starting at a known solution (x_0, α_0) and perturbing this solution in small steps to trace the solution curve of this system. The implicit function theorem forms the foundation of these methods by guaranteeing the existence of a unique solution (x_1, α_1) in the neighborhood of (x_0, α_0), if the Jacobian matrix $\partial_x f(x, \alpha)$ is invertible at (x_0, α_0). Numerical continuation methods are used in bifurcation analysis to compute curves of steady state solutions, periodic orbits and bifurcation points.

A naive strategy for numerical continuation might be to perturb solutions along the natural parameter α, by stepping from a known solution x_0 at α_0 to find a

new solution at $\alpha_0 + \Delta\alpha_0$. However, this approach has a significant drawback: it fails at fold points where the solution curve reverses direction. We know from the implicit function theorem that a solution exists at $\alpha_0 + \Delta\alpha_0$ as long as the Jacobian matrix $\partial_x f(x_0, \alpha_0)$ remains invertible and the step size $\Delta\alpha_0$ is sufficiently small. However, at fold points, this condition fails as the Jacobian matrix becomes singular and the implicit function theorem no longer holds. Therefore, to address this issue, more sophisticated numerical continuation methods, such as pseudo-arclength continuation and deflated continuation, have been developed.

2.2 Pseudo-arclength continuation

Pseudo-arclength continuation is one of the most widely used numerical continuation methods for generating bifurcation diagrams. This method addresses the issue of Jacobian matrices becoming singular at fold points by parameterizing the solution curve using the approximate arclength instead of the natural parameter α.

To solve the system of nonlinear equations $f(x, \alpha) = 0$ for all $\alpha \in \mathbb{R}$, suppose we are given a known solution (x_0, α_0). We introduce a new parameter $s \in \mathbb{R}$, which approximates the arclength of the solution curve in the x-α plane. Without loss of generality, let us assume that $x(0) = x_0$ and $\alpha(0) = \alpha_0$. The solutions in the neighborhood of $s = 0$ are then given by:

$$f\left(x\left(s\right), \alpha\left(s\right)\right) = 0.$$

The numerical continuation process in the pseudo-arclength method follows a sequence of predictor and corrector steps. Starting from any known solution on the curve, a new point is first predicted using a predictor step. Since this predicted point may not lie exactly on the solution curve, Newton's method is then used to correct the prediction onto the nearest solution on the curve in the corrector step. In this way, a continuous solution curve can be traced out in the x-α plane as illustrated in Figure 2.1 for the one-dimensional case.

In the following, we will explain the predictor and corrector steps in more detail, and discuss possible modifications and limitations of this method. We have also presented the pseudo-arclength continuation method in the form of pseudocode in Algorithm 2.1. For a more detailed discussion of the theoretical background for this method, we refer the reader to the lecture notes by Keller [Kel86] and the book by Kuznetsov [Kuz23]. The content of this section is based on these references.

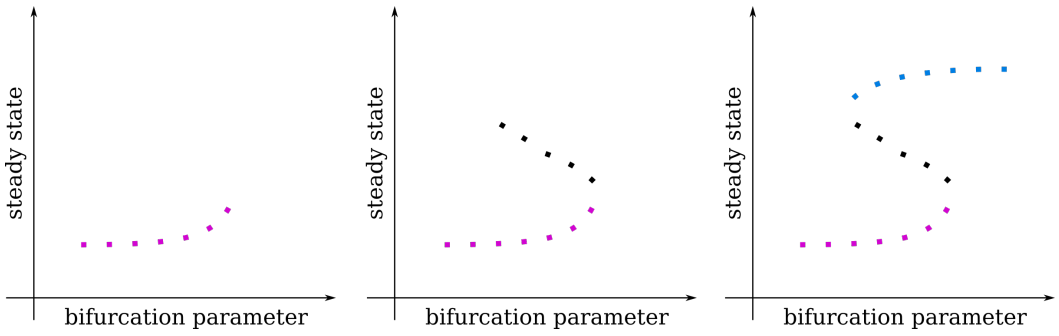


Fig. 2.1.: Visualization of the different stages (left to right) of computing a bifurcation diagram using the pseudo-arclength continuation method. The magenta and blue points represent two different stable steady state curves separated by the unstable manifold plotted in black.

2.2.1 Predictor step

In this step, we use a given solution (x_0, α_0) to compute the tangent vector $(\dot{x}_0, \dot{\alpha}_0)$ that can help us to predict the next solution. We can compute this unit tangent vector by solving the following linear system for the unknowns $(\dot{x}, \dot{\alpha})$:

$$\partial_x f(x_0, \alpha_0)\dot{x} + \partial_\alpha f(x_0, \alpha_0)\dot{\alpha} = 0, \tag{2.1a}$$

$$\|\dot{x}\|^2 + |\dot{\alpha}|^2 = 1. \tag{2.1b}$$

For this, let us first consider the case where $\partial_x f(x_0, \alpha_0)$ is non-singular. Then, we can re-write equation (2.1a) as:

$$\partial_x f(x_0, \alpha_0)\dot{x} = -\partial_\alpha f(x_0, \alpha_0)\dot{\alpha} \quad \Longleftrightarrow \quad \partial_x f(x_0, \alpha_0)u = -\partial_\alpha f(x_0, \alpha_0) \tag{2.2}$$

where $\dot{x} = uv$ and $\dot{\alpha} = v$. We can then solve the linear system (2.2) to compute u. To obtain v, we can then use the following relation from equation (2.1b):

$$v = \pm \frac{1}{\sqrt{1 + \|u\|^2}}.$$

The sign of v is chosen such that the previous tangent vectors and the current tangent vectors point in the same direction. This is equivalent to the following condition:

$$\dot{x}_{-1}^T \dot{x} + \dot{\alpha}_{-1}\dot{\alpha} > 0 \quad \Longleftrightarrow \quad v\left(\dot{x}_{-1}^T u + \dot{\alpha}_{-1}\right) > 0.$$

This ensures that the solutions are traced in the correct direction without the iterations getting stuck or re-tracing previously found solutions.

Now, let us consider the case where the Jacobian matrix $\partial_x f(x_0, \alpha_0)$ is almost singular with rank $(\partial_x f(x_0, \alpha_0)) = n - 1$. In this case, we use LU-decomposition with full pivoting on this matrix to get:

$$\partial_x f(x_0, \alpha_0) = PLUQ \equiv P \cdot \begin{pmatrix} \hat{L} & 0 \\ \hat{l}^T & 1 \end{pmatrix} \cdot \begin{pmatrix} \hat{U} & \hat{u} \\ 0 & \epsilon \end{pmatrix} \cdot Q$$

where $P \in \mathbb{R}^{n \times n}$ and $Q \in \mathbb{R}^{n \times n}$ are orthogonal permutation matrices, and L and U are lower and upper triangular matrices respectively. The matrices $\hat{L} \in \mathbb{R}^{(n-1)\times(n-1)}$ and $\hat{U} \in \mathbb{R}^{(n-1)\times(n-1)}$ are non-singular lower and upper triangular matrices respectively, $\hat{l} \in \mathbb{R}^{n-1}$ and $\hat{u} \in \mathbb{R}^{n-1}$ are vectors, and $\epsilon \in \mathbb{R}$ is an approximation to zero.

In order to solve the linear system in equation (2.2) using this decomposition, we define:

$$\begin{pmatrix} h \\ \eta \end{pmatrix} = -P^T \partial_\alpha f(x_0, \alpha_0) \quad \text{and} \quad \begin{pmatrix} \Psi \\ \psi \end{pmatrix} = Qu.$$

Using this, the linear system can be solved with forward and backward substitution as follows:

$$\begin{pmatrix} \hat{L} & 0 \\ \hat{l}^T & 1 \end{pmatrix} \begin{pmatrix} \Phi \\ \phi \end{pmatrix} = \begin{pmatrix} h \\ \eta \end{pmatrix} \quad \Longleftrightarrow \quad \begin{pmatrix} \Phi \\ \phi \end{pmatrix} = L^{-1} \begin{pmatrix} h \\ \eta \end{pmatrix}$$

$$\begin{pmatrix} \hat{U} & \hat{u} \\ 0 & \epsilon \end{pmatrix} \begin{pmatrix} \Psi \\ \psi \end{pmatrix} = \begin{pmatrix} \Phi \\ \phi \end{pmatrix} \quad \Longleftrightarrow \quad \begin{pmatrix} \Psi \\ \psi \end{pmatrix} = U^{-1} \begin{pmatrix} \Phi \\ \phi \end{pmatrix}$$

$$\begin{aligned}
&\hat{L}\Phi = h && \Longleftrightarrow && \Phi = \hat{L}^{-1} h && \in \mathbb{R}^{n-1} \\
&\hat{l}^T \Phi + \phi = \eta && \Longleftrightarrow && \phi = \eta - \hat{l}^T \Phi && \in \mathbb{R} \\
&\epsilon\psi = \phi && \Longleftrightarrow && \psi = \frac{\phi}{\epsilon} && \in \mathbb{R} \\
&\hat{U}\Psi + \hat{u}\psi = \Phi && \Longleftrightarrow && \Psi = \hat{U}^{-1}(\Phi - \hat{u}\psi) && \in \mathbb{R}^{n-1}
\end{aligned}$$

Once we have computed the unit tangent vector $(\dot{x}_0, \dot{\alpha}_0)$ at the given solution (x_0, α_0), we can predict a new point (x_p, α_p) close to the solution curve as follows:

$$\begin{aligned}
x_p &= x_0 + \sigma \cdot \Delta s \cdot \dot{x}_0, \\
\alpha_p &= \alpha_0 + \sigma \cdot \Delta s \cdot \dot{\alpha}_0.
\end{aligned}$$

Here, $\sigma \in \{-1, +1\}$ is a parameter that indicates the direction of the continuation, and Δs is the step size. The predicted point (x_p, α_p) is then used as an initial guess for the corrector step.

2.2.2 Corrector step

We can now use Newton's method to correct the predicted point (x_p, α_p) onto the solution curve. Since Newton's method can only be used on systems of nonlinear equations that have the same number of equations as unknowns, we introduce an additional equation to the original nonlinear problem to get the following augmented system:

$$f(x(s), \alpha(s)) = 0, \tag{2.3}$$
$$g\left(x(s), \alpha(s)\right) = 0, \tag{2.4}$$

where the function $g : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$ represents the normalization of the arclength parameter s. Since we have used a predictor that employs the tangent, we can define g as follows:

$$\dot{x}_0^T (x - x_0) + \dot{\alpha}_0(\alpha - \alpha_0) - \Delta s = 0.$$

This equation represents a plane, which is perpendicular to the unit tangent vector $(\dot{x}_0, \dot{\alpha}_0)$ and lying at a distance Δs from the point (x_0, α_0) on the solution path.

Therefore, the Newton iterations start at the point $(x^{(0)}, \alpha^{(0)}) := (x_p, \alpha_p)$ and proceed by taking the following steps for $j = 0, 1, 2, \ldots$ until convergence:

$$\begin{pmatrix} x^{(j+1)} \\ \alpha^{(j+1)} \end{pmatrix} = \begin{pmatrix} x^{(j)} \\ \alpha^{(j)} \end{pmatrix} - \begin{pmatrix} \partial_x f(x^{(j)}, \alpha^{(j)}) & \partial_\alpha f(x^{(j)}, \alpha^{(j)}) \\ \partial_x g(x^{(j)}, \alpha^{(j)}) & \partial_\alpha g(x^{(j)}, \alpha^{(j)}) \end{pmatrix}^{-1} \begin{pmatrix} f(x^{(j)}, \alpha^{(j)}) \\ g(x^{(j)}, \alpha^{(j)}) \end{pmatrix}.$$

It is important to note that the approximation of the arclength condition should be chosen in such a way that the Jacobian matrix in the Newton iterations always remains non-singular on the solution curve.

2.2.3 Additional features

Adaptive step size

When choosing the step size Δs, we need to make sure that it is neither too small nor too large so that we trace the solution curve efficiently. To achieve this, we use an adaptive step size, which may improve both the speed and accuracy of our numerical continuation. This means that, after each corrector step, we increase

the step size by a factor $\beta \in (1, \infty)$ if the Newton iterations converge very quickly[1]. Likewise, we decrease the step size by the factor $1/\beta$ if the Newton iterations fail to converge. Furthermore, in order to prevent the step sizes from getting arbitrarily small or large in this process, we also bound the step size between a given range $[\Delta s_{\min}, \Delta s_{\max}]$. This approach can also be found in the popular bifurcation software MATCONT [Gov+19].

Bifurcation detection

In order to detect bifurcation points as we continue along the solution curve, we can monitor the values of some test functions [Kuz23]. These test functions are smooth scalar functions that have regular zeros at the bifurcation points and depend on the type of bifurcation that needs to be detected.

For instance, we can use the following test function to detect a fold or saddle-node bifurcation point:

$$\rho_{SN}(x, \alpha) = \det\left(\partial_x f(x, \alpha)\right) = \prod_{i=1}^{n} \mu_i. \tag{2.5}$$

This function computes the product of all the eigenvalues μ_i of the Jacobian matrix $\partial_x f(x, \alpha)$ at each iterate. When a stable steady state loses its stability along the solution curve (or when an unstable steady state becomes stable) through a saddle-node bifurcation, the determinant of the Jacobian matrix changes signs as an eigenvalue changes its sign. We track this using the test function ρ_{SN} whose root corresponds to a saddle-node bifurcation point.

Another example is the following test function to detect Hopf bifurcation points:

$$\rho_H(x, \alpha) = \prod_{i>j} \left(\mu_i(x, \alpha) + \mu_j(x, \alpha)\right), \tag{2.6}$$

where μ_i and μ_j are the eigenvalues of the Jacobian matrix $\partial_x f(x, \alpha)$. This function changes sign when a pair of complex conjugate eigenvalues crosses the imaginary axis with non-zero imaginary part. However, caution must be taken when using this test function to detect Hopf bifurcation points because the function is also zero when the Jacobian matrix has at least two eigenvalues such that $\mu_1 = -\mu_2$.

[1] In our implementation, we use the factor $\beta = 2$.

Algorithm 2.1: Pseudo-arclength continuation method tracing data points

Input: augmented system $(f(x,\alpha), g(x,\alpha))^T$
Input: given data points $\eta_1 < \cdots < \eta_M$ (optional)
Input: known solution (x_0, α_0)
Input: initial direction $\sigma \in \{-1, +1\}$
Input: initial step size $\Delta s > 0$
Input: step size range $[\Delta s_{\min}, \Delta s_{\max}]$
Input: parameter range $[\alpha_{\min}, \alpha_{\max}]$
Input: maximum number of iterations $\overline{N}$

for $i = 0$ to $\overline{N}$ **do**
  if $\alpha_i \notin [\alpha_{\min}, \alpha_{\max}]$ **then**
    break
  compute derivatives $\partial_x f(x_i, \alpha_i)$ and $\partial_\alpha f(x_i, \alpha_i)$
  compute unit tangents $\dot{x}_i$ and $\dot{\alpha}_i$ by solving the linear equations (2.1)
  if $i > 0$ **then**
    $\sigma \leftarrow \text{sign}\left(\dot{x}_{i-1}^T \dot{x}_i + \dot{\alpha}_{i-1}\dot{\alpha}_i\right)$
  success $\leftarrow$ False
  while success is False and $\Delta s \geq \Delta s_{\min}$ **do**
    trace data points (if any):
      switch σ **do**
        case -1: find $\eta \in \{\eta_1, \ldots, \eta_M\}$ such that $\alpha_i > \eta > \alpha_i - \Delta s \cdot \dot{\alpha}_i$
        case $+1$: find $\eta \in \{\eta_1, \ldots, \eta_M\}$ such that $\alpha_i < \eta < \alpha_i + \Delta s \cdot \dot{\alpha}_i$
      $\Delta s \leftarrow \frac{\eta - \alpha_i}{\sigma \cdot \dot{\alpha}_i}$
    predictor step:
      $x_p \leftarrow x_i + \sigma \cdot \Delta s \cdot \dot{x}_i$
      $\alpha_p \leftarrow \alpha_i + \sigma \cdot \Delta s \cdot \dot{\alpha}_i$
    corrector step:
      compute derivatives $\partial_x g(x_i, \alpha_i)$ and $\partial_\alpha g(x_i, \alpha_i)$
      use Newton's method to solve the augmented system
      if solution (x^*, α^*) found **then**
        success $\leftarrow$ True
        $x_{i+1} \leftarrow x^*$
        $\alpha_{i+1} \leftarrow \alpha^*$
        if fast Newton convergence and $\beta \cdot \Delta s \leq \Delta s_{\max}$ **then**
          $\Delta s \leftarrow \beta \cdot \Delta s$
      else
        $\Delta s \leftarrow \frac{1}{\beta} \cdot \Delta s$

Output: solution curve (x_i, α_i) for $i = 0, 1, \ldots, N$ where $N \leq \overline{N}$

Tracing given sequence of data points

Suppose we are given a set of data points $\eta_1 < \cdots < \eta_M$ with $M \in \mathbb{N}$, and we need to trace the solution curve such that the parameters $(\alpha_i)_{i=0,1,\ldots}$ pass through these data points[2]. For this, we need to adapt the predictor and corrector steps of the pseudo-arclength method. Here we present our heuristic solution to this problem.

For each iterate (x_i, α_i) of the pseudo-arclength method, we can check in the predictor step if any of the given data points $\eta_1 < \eta_2 < \cdots < \eta_M$ lie between the current parameter value α_i and the predicted parameter value α_{i+1}. This can be done by checking for all $j \in \{1, 2, \ldots, M\}$ if:

$$\alpha_i < \eta_j < \alpha_i + \Delta s \cdot \dot{\alpha}_i \quad \text{or} \quad \alpha_i > \eta_j > \alpha_i - \Delta s \cdot \dot{\alpha}_i.$$

If we find such a data point η_j, we then reduce the step size Δs so that the predicted parameter α_{i+1} is exactly equal to the data point η_j instead:

$$\Delta s = \frac{\eta_j - \alpha_i}{\sigma \cdot \dot{\alpha}_i}.$$

In this case, instead of the augmented system (2.3), we simply solve the equation $f(x, \alpha_{i+1}) = 0$ to find a solution x_{i+1} in the corrector step. This ensures that α_{i+1} remains at the data point η_j, and we find a solution on the curve for every data point.

2.2.4 Limitations

One of the main drawbacks of the pseudo-arclength continuation method is that it can only trace solution curves that are continuously connected to the initial known solution. As a result, the algorithm cannot detect solution branches that are disconnected from each other. This means that, in order to obtain a complete picture of the solution landscape, we need to repeat the algorithm multiple times with different starting points. However, in practice, it is generally difficult to know *a priori*, if there are disconnected solution curves to be found and, if yes, which initial points can us lead us there. Therefore, we need a different approach to address this problem.

[2] We will see in Section 4.4 why we need this.

2.3 Deflated continuation

An alternative approach to numerical bifurcation analysis is the deflated continuation method introduced in 2016 by Farrell et al. [FBB16]. This method is designed to not only find continuously connected branches of solutions from an initial point, but to also discover new disconnected solution branches.

To understand how this method works, we need to first introduce the notion of deflation. Consider once again the system of nonlinear equations $f(x, \alpha) = 0$. We now additionally define a new function F as follows:

$$F(x) \equiv f(x, \alpha^*) = 0, \tag{2.7}$$

where $F : \mathbb{R}^n \to \mathbb{R}^n$ is a smooth function and the parameter $\alpha^* \in \mathbb{R}$ is a fixed parameter.

Suppose we know $m > 0$ roots $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$ of this system of nonlinear equations (2.7). Then we can find other unknown roots of this system by modifying the problem $F(x) = 0$ such that the roots of F are preserved, but Newton's method can no longer converge to the known roots. This is the main idea behind deflation introduced for the n-dimensional case by Brown and Gearhart in 1970 [BG71].

In order to force the Newton iterations to converge to an unknown solution, we apply a continuously differentiable deflation operator $M(x, x^*) : \mathbb{R}^n \to \mathbb{R}^n$ to the system of nonlinear equations $F(x) = 0$. This operator satisfies two important properties:

$$\forall x \neq x^* : M(x, x^*)F(x) = 0 \quad \Longleftrightarrow \quad x = x^*, \tag{2.8a}$$

$$\lim_{x \to x^*} \inf \|M(x, x^*)F(x)\| > 0. \tag{2.8b}$$

Property (2.8a) ensures that the solutions are preserved and property (2.8b) ensures the deflated residual does not converge to zero for any sequence of Newton iterates x that approach a known root x^*.

Farrell et al. [FBB16] then used this principle of deflation to develop a new method for computing bifurcation diagrams that combines the classical continuation approach to extend known solutions and the deflation technique to find new solution branches. Therefore, starting at a given solution, the deflated continuation method alternates between a deflation step and a continuation step to trace the solution curve as illustrated in Figure 2.2.

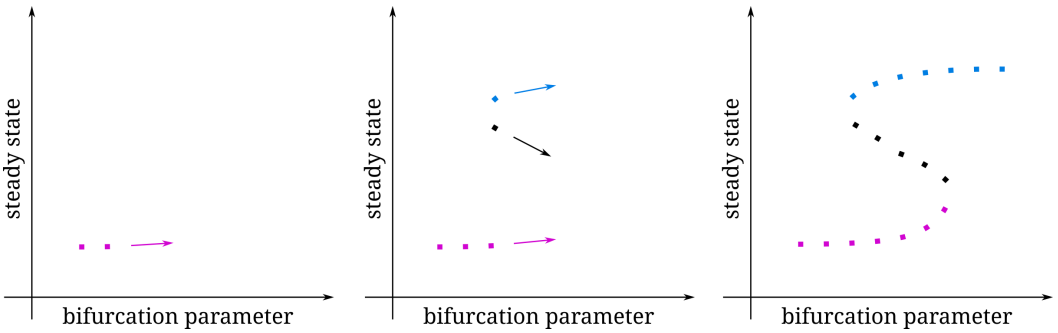


Fig. 2.2.: Visualization of the different stages (left to right) of computing a bifurcation diagram using the deflated continuation method. The magenta and blue points represent two different stable steady state curves separated by the unstable manifold plotted in black.

In this thesis, we have also adapted the basic algorithm introduced by Farrell et al. to make it more robust, allow adaptive step size and trace given data points. Therefore, in the remainder of this section, we will present the deflation and continuation steps of this method in more detail, explain our proposed modifications and then discuss some limitations of this method. Finally, a pseudocode for the deflated continuation method is also presented in Algorithm 2.2.

2.3.1 Deflation step

Suppose we have the set of known solutions $S(\alpha^*) = \{x^{(1)}, \ldots, x^{(m)}\}$ of the system of nonlinear equations (2.7) at the parameter value α^*. The goal of the deflation step is to find other solutions of this equation by deflating out the known solutions. This means that we use Newton's method to solve the following nonlinear equation:

$$\tilde{F}(x) \equiv M(x, x^{(m)})M(x, x^{(m-1)}) \ldots M(x, x^{(1)})F(x) = 0.$$

Then every time a new solution $x^{(m+1)}$ is found, it is included in the set of known solutions $S(\alpha^*)$ and deflated out. This step is repeated until Newton's method can no longer find new solutions.

To deflate out a known solution x^*, Farrell et al. [FBB16] propose, for example, the following shifted deflation operator:

$$M(x, x^*) = \left(\frac{1}{\|x - x^*\|_p} + s \right) \mathbb{I},$$

where $\mathbb{I}$ is the identity matrix, $s \in \mathbb{R}$ is the shift and $p \in \mathbb{N}_+$ is the power. Farrell et al. use the values $p = 2$ and $s = 1$ in all their examples [FBB16]. In our implementation of the deflated continuation method, we will use the same deflation operator.

2.3.2 Continuation step

In the basic algorithm, the continuation step takes fixed steps $\Delta\alpha$ in a given direction $\sigma \in \{-1, +1\}$ until the bounds $[\alpha_{\min}, \alpha_{\max}]$ of the parameter α are reached. This means that, if we have a set of solutions $S(\alpha^*) = \{x^{(1)}, \ldots, x^{(m)}\}$ at the parameter value α^*, then we can use the continuation step to extend these solutions to the next parameter value $\alpha^* + \sigma \cdot \Delta\alpha$. We do this by using predictor and corrector steps similar to the pseudo-arclength continuation method.

Given a solution x^* at the parameter value α^*, the predictor step is given by:

$$\begin{aligned} x_p &= x^* - \partial_x f(x^*, \alpha^*)^{-1} \cdot \partial_\alpha f(x^*, \alpha^*) \cdot \sigma \cdot \Delta\alpha, \\ \alpha_p &= \alpha^* + \sigma \cdot \Delta\alpha. \end{aligned}$$

The predicted solution x_p is then corrected using Newton's method by solving the nonlinear equation $F(x) = f(x, \alpha_p) = 0$ starting at the initial guess x_p.

This process is repeated for all points in $S(\alpha^*)$ to obtain the set of solutions $S(\alpha^* + \sigma \cdot \Delta\alpha)$ at the next parameter value $\alpha^* + \sigma \cdot \Delta\alpha \equiv \alpha_p$.

2.3.3 Additional features

Adaptive step size

Since the basic algorithm uses a fixed step size $\Delta\alpha$ in the continuation step, it may encounter convergence problems in the Newton corrector, if this step size is chosen too large. To address this issue, we iteratively reduce the step size by a factor $1/\beta$ where $\beta \in \mathbb{N}_+$. We then attempt the continuation step for each reduced step size until we get convergence in the Newton corrector. If it takes us $k > 0$ attempts to successfully continue the solutions from α^* to $\alpha^* + \sigma \cdot \beta^{-k} \cdot \Delta\alpha$, then we perform the continuation step k times with the reduced step size $\beta^{-k} \cdot \Delta\alpha$ until the target parameter value $\alpha^* + \sigma \cdot \Delta\alpha$ is reached.

Bifurcation detection

In order to detect saddle-node bifurcations or fold points from the steady-state curve produced by the deflated continuation method, we do not need to track the determinant or eigenvalues of the Jacobian matrix $\partial_x f(x^*, \alpha^*)$ at each solution point (x^*, α^*) like we did in the pseudo-arclength method. Since the deflated continuation method essentially tracks the number of solutions found at each parameter value, we can simply check for changes in the number of solutions to identify a saddle-node bifurcation. Therefore, when the number of solutions increase or decrease by two as we continue from parameter α^* to $\alpha^* + \sigma \cdot \Delta\alpha$, we conclude that a saddle-node bifurcation lies between these two parameter values. We can also verify this by checking the condition ρ_{SN} from equation (2.5) at the two parameter values.

On the other hand, since Hopf bifurcation points do not result in a change in the number of steady states on the solution curve, we need to track the eigenvalues of the Jacobian matrix $\partial_x f(x^*, \alpha^*)$ as in the test function ρ_H from equation (2.6) in order to identify Hopf bifurcation points.

Tracing given sequence of data points

Suppose we are given a set of data points $\eta_1 < \cdots < \eta_M$ with $M \in \mathbb{N}$, and we need to trace the solution curve such that the parameters $(\alpha_i)_{i=0,1,\ldots}$ pass through these data points[3]. We can do this in the deflated continuation method by simply adjusting the step size before proceeding from the deflation step to the continuation step.

Therefore, for each iterate (x^*, α^*), we check before the continuation step if any of the given data points $\eta_1 < \eta_2 < \cdots < \eta_M$ lie between the current parameter value α^* and the predicted parameter value $\alpha^* + \sigma \cdot \Delta\alpha$. This can be done by checking for all $j \in \{1, 2, \ldots, M\}$ if:

$$\alpha^* < \eta_j < \alpha^* + \Delta\alpha \quad \text{or} \quad \alpha^* > \eta_j > \alpha^* - \Delta\alpha.$$

If we find such a data point η_j, we reduce the step size $\Delta\alpha$ so that the predicted parameter value $\alpha^* + \sigma \cdot \Delta\alpha$ is exactly equal to the data point η_j instead:

$$\Delta\alpha \leftarrow \frac{\eta_j - \alpha_i}{\sigma}.$$

The remaining steps are then carried out as usual. However, since this step size adaptation can potentially make the steps arbitrarily small, we also regularly check

[3] We will see in Section 4.4 why we need this.

before each continuation step if the step size is smaller than a given threshold and increase the step size if necessary[4].

Perturbed initialization

The system of nonlinear equations $f(x, \alpha) = 0$ may sometimes have trivial solutions $\bar{x}$ such that $f(\bar{x}, \alpha) = 0$ for all $\alpha \in \mathbb{R}$. Since the basic algorithm [FBB16] uses the solutions found at a parameter value α^* as initial guesses for the deflation step at the next parameter value $\alpha^* + \sigma \cdot \Delta\alpha$, the deflated residuals will evaluate to $0/0$ at the trivial solutions causing the deflation step to fail. To address this issue, Farrell et al. [FBB16] suggest excluding such trivial solutions a priori from the set of known solutions $S(\alpha^*)$. However, in our implementation of this method for parameter estimation (see Chapter 4 and Chapter 9), we would like to automatically generate the bifurcation diagram with minimal a priori information. Therefore, we address this issue by adding a small perturbation $\epsilon \in \mathcal{N}(0, \Sigma)$ to each solution in $S(\alpha^*)$ in the deflation step. Here, $\Sigma \in \mathbb{R}^{n \times n}$ is assumed to be a symmetric positive-definite matrix. We choose the perturbation small enough such that the sufficient conditions for convergence of Newton's method with the deflated function are still satisfied [FBB16].

2.3.4 Limitations

Although the deflated continuation method offers a powerful alternative to the pseudo-arclength continuation method, it also has some limitations. In particular, if there exists a solution x^* at α^* such that $f(\tau x^*, \alpha^*) = 0$ for all $\tau \in \mathbb{R}$, then the deflation operator treats each solution τx^* as distinct, resulting in a spurious explosion of solutions in the deflation step. Therefore, such solutions need to be accounted for and excluded using additional conditions. Furthermore, since the deflated continuation method relies on the Newton's method to find new solution branches from the known solutions, it does not guarantee that all disconnected solution branches will be found. In fact, the deflation step may fail to converge to a solution, when the previously found solutions that are used as initial guesses lie far away from the unknown solutions.

[4]For better readability, we have not accounted for this in the pseudocode in Algorithm 2.2.

Algorithm 2.2: Deflated continuation method tracing data points

Input: system of nonlinear equations $f(x, \alpha)$
Input: given data points $\eta_1 < \cdots < \eta_M$ (optional)
Input: deflation operator $M(x; x^*) := s + \frac{1}{\|x - x^*\|_p^2}$
Input: known solution (x_0, α_0)
Input: direction $\sigma \in \{-1, +1\}$
Input: step size $\Delta\alpha > 0$
Input: parameter range $[\alpha_{\min}, \alpha_{\max}]$

$i = 0$
$S(\alpha_0) \leftarrow \{x_0\}$
$F(.) \leftarrow f(\cdot, \alpha_0)$
while $\alpha_i \in [\alpha_{\min}, \alpha_{\max}]$ **do**
  $T \leftarrow S(\alpha_i)$
  deflation step:
    for $y \in T$ **do**
      repeat
        $y_0 \leftarrow y + \epsilon$ where $\epsilon \in \mathcal{N}(0, \Sigma)$
        apply Newton's method to F with y_0 as initial guess
        if solution y^* found **then**
          $S(\alpha_i) \leftarrow S(\alpha_i) \cup \{y^*\}$
          $F(.) \leftarrow M(\cdot; y^*)F(\cdot)$
      until Newton's method fails to converge
  trace data points (if any):
    switch σ **do**
      case -1: find $\eta \in \{\eta_1, \ldots, \eta_M\}$ such that $\alpha_i > \eta > \alpha_i - \Delta\alpha$
      case $+1$: find $\eta \in \{\eta_1, \ldots, \eta_M\}$ such that $\alpha_i < \eta < \alpha_i + \Delta\alpha$
    $\Delta\alpha \leftarrow \frac{\eta - \alpha_i}{\sigma}$
  continuation step:
    $\alpha_{i+1} \leftarrow \alpha_i + \sigma \cdot \Delta\alpha$
    $S(\alpha_{i+1}) \leftarrow \varnothing$
    $F(.) \leftarrow f(\cdot, \alpha_{i+1})$
    for $y \in S(\alpha_i)$ **do**
      $(\tilde{y}, \tilde{\alpha}) \leftarrow (y, \alpha_i)$
      while $\Delta\alpha > \epsilon > 0$ **do**
        use predictor-corrector step from $(\tilde{y}, \tilde{\alpha})$ to solve $f(\cdot, \tilde{\alpha} + \sigma\Delta\alpha) = 0$
        if solution y^* found **then**
          if $\tilde{\alpha} + \sigma \cdot \Delta\alpha == \alpha_{i+1}$ **then**
            $S(\alpha_{i+1}) \leftarrow S(\alpha_{i+1}) \cup \{y^*\}$
            $F(.) \leftarrow M(\cdot; y^*)F(\cdot)$
            break
          else
            $(\tilde{y}, \tilde{\alpha}) \leftarrow (y^*, \tilde{\alpha} + \sigma \cdot \Delta\alpha)$
        else
          $\Delta\alpha \leftarrow \frac{1}{\beta}\Delta\alpha$
    $i \leftarrow i + 1$

3 Nonlinear Optimization

In this chapter, we provide a brief introduction to the theory of nonlinear optimization and present some numerical methods for solving nonlinear optimization problems. We start by introducing the general form of constrained optimization problems, with a special emphasis on nonlinear least-squares problems, and introduce the relevant terminology and optimality conditions for the solutions. We then move on to the numerical techniques. In particular, we present the sequential quadratic programming (SQP) method and the generalized Gauss-Newton method for solving constrained nonlinear optimization problems, together with a discussion of various globalization strategies. Finally, we examine how the block structure in the Jacobian matrix of a large-scale nonlinear least-squares problems can be exploited to efficiently solve the optimization problem.

For a deeper insight into the topic, we refer the reader to standard textbooks in nonlinear optimization by Nocedal and Wright [NW06], Gill, Murray and Wright [GMW19] or Fletcher [Fle00]. The general introduction to the theory of nonlinear optimization and the section on the SQP method in this chapter closely follow the book by Nocedal and Wright [NW06]. The sections on the generalized Gauss-Newton method and its structure-exploiting variant are based on the work of Bock [Boc87] and Schlöder [Sch87].

3.1 Basic concepts

Optimization is an important mathematical tool used in various fields of science and engineering to find the best possible solution to a given problem. This generally involves the minimization or maximization of some performance indicator (called the *objective function*) to find characteristics or *optimization variables* of the system that provide the best performance under certain restrictions or constraints. When the objective function or some of the constraints are nonlinear, we get a constrained nonlinear optimization problem. In this section, we briefly introduce some terminology and theorems for constrained nonlinear optimization problems.

3.1.1 Constrained optimization

Consider the general form of a constrained nonlinear optimization problem:

Problem 3.1.1 : Nonlinear Optimization Problem (NLP)
Find a solution $x \in \mathbb{R}^n$ to the optimization problem:

$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{s.t.} \quad & g(x) = 0, \\ & h(x) \geq 0, \end{aligned}$$

where $f : \mathbb{R}^n \to \mathbb{R}^{m_f}$, $g : \mathbb{R}^n \to \mathbb{R}^{m_g}$, $h : \mathbb{R}^n \to \mathbb{R}^{m_h}$. All the functions are assumed to be sufficiently smooth, usually $\mathcal{C}^2$ or $\mathcal{C}^3$.

Before we study the solutions of this optimization problem, let us first identify which points in the domain $\mathbb{R}^n$ are actually feasible.

Definition 3.1.2 : Feasible Point
A point $x \in \mathbb{R}^n$ that satisfies all the equality and inequality constraints of the NLP 3.1.1 is called a *feasible point*. The set Ω of all such feasible points is called the *feasible set*. This means that:

$$\Omega = \{x \in \mathbb{R}^n : g(x) = 0 \text{ and } h(x) \geq 0\}.$$

This means that solutions of the nonlinear optimization problem are chosen from the feasible set. A solution can be classified into two types depending upon the scope of our search:

Definition 3.1.3 : Local Minimizer
A feasible point $x^* \in \Omega$ is called a *local solution* or *local minimizer* of the NLP 3.1.1 if there exists a neighborhood $\mathcal{N}$ of x^* such that $f(x^*) \leq f(x)$ for all $x \in (\Omega \cap \mathcal{N})$.

Definition 3.1.4 : Global Minimizer
A feasible point $x^* \in \Omega$ is called a *global solution* or *global minimizer* of the NLP 3.1.1 if $f(x^*) \leq f(x)$ for all $x \in \Omega$.

Before we state the optimality conditions for a solution to the nonlinear optimization problem, we also need to introduce the concepts of Lagrangian functions, Lagrange multipliers, and constraint qualifications.

Definition 3.1.5 : Lagrangian Function
The *Lagrangian function* for the NLP 3.1.1 is defined as:

$$\mathcal{L}(x, \mu, \nu) = f(x) - \mu^T g(x) - \nu^T h(x)$$

where $\mu \in \mathbb{R}^{m_g}$ and $\nu \in \mathbb{R}^{m_h}$ are called the *Lagrange multipliers* of the equality and inequality constraints respectively.

Definition 3.1.6 : Active Set
At any feasible point $x \in \Omega$, the inequality constraint $h_i(x)$ is said to be *active* if $h_i(x) = 0$ and *inactive* if $h_i(x) > 0$. The set of all equality constraints and the inequality constraints that are active at x is called the *active set* $\mathcal{A}(x)$, i.e.,

$$\mathcal{A}(x) = \{i \in \{1, \ldots, m_g\} : g_i(x) = 0\} \cup \{i \in \{1, \ldots, m_h\} h_i(x) = 0\}.$$

Definition 3.1.7 : Linearly Independent Constraint Qualification
Linearly independent constraint qualification (LICQ) holds at a feasible point $x \in \Omega$ if and only if the set of all active constraint gradients is linearly independent. In this case, all vectors $\nabla g_i(x)$ for $i \in \{1, 2, \ldots, m_g\}$ and $\nabla h_i(x)$ for $i \in \mathcal{A}(x)$ are linearly independent.

This allows us to now formulate the necessary and sufficient optimality conditions that characterize a solution of the NLP 3.1.1.

Theorem 3.1.8 : First-Order Necessary Optimality Conditions
Assume that $x^* \in \Omega$ is a local minimizer of the NLP 3.1.1 and that LICQ holds at x^*. Then, there exist Lagrange multipliers $\mu^* \in \mathbb{R}^{m_g}$ and $\nu^* \in \mathbb{R}^{m_h}$ such that:

$$\begin{aligned}
\nabla f(x^*) - \nabla g(x^*)\mu^* - \nabla h(x^*)\nu^* &= 0, && (3.1)\\
g(x^*) &= 0, && (3.2)\\
h(x^*) &\geq 0, && (3.3)\\
\nu^* &\geq 0, && (3.4)\\
\nu_i^* h_i(x^*) &= 0, \quad \forall i = 1, \ldots, m_h. && (3.5)
\end{aligned}$$

Proof. See Chapter 12 in [NW06]. □

The first-order necessary optimality conditions are also known as *Karush-Kuhn-Tucker conditions* or *KKT conditions* after the mathematicians William Karush, Harold

William Kuhn and Albert William Tucker who formulated them. Any point $x^* \in \Omega$ that satisfies the KKT conditions is called a *KKT point* or *stationary point*.

Furthermore, equations (3.5) are known as *complementarity conditions* because they suggest that the inequality constraint $h_i(x^*)$ is either active at the solution x^*, i.e., $h_i(x^*) = 0$, or its corresponding Lagrange multiplier $\nu_i^* = 0$. It can also happen that both $h_i(x^*) = 0$ and $\nu_i^* = 0$ hold true.

Definition 3.1.9 : Linearized Feasible Directions
Given a feasible point $x \in \Omega$ and the active set $\mathcal{A}(x)$, the set of *linearized feasible directions* $\mathcal{F}(x)$ is given by

$$\mathcal{F}(x) = \left\{ w \in \mathbb{R}^n : \nabla g_i(x)^T w = 0 \; \forall i = 1, \ldots, m_g, \; \nabla h_i(x)^T w \geq 0 \; \forall i \in \mathcal{A}(x) \right\}.$$

The KKT conditions tell us that moving from a local minimizer x^* along a linearized feasible direction $w \in \mathcal{F}(x^*)$ would either increase the first-order approximation of the objective function $w^T \nabla f(x^*) > 0$ or keep it constant $w^T \nabla f(x^*) = 0$. In the latter case, we do not have enough information from the first-order approximation to determine whether moving along w will increase or decrease the objective function. Therefore, we need to additionally look at the second-order derivatives of the Lagrangian function to get a better understanding of the solution.

Definition 3.1.10 : Critical Cone
For a KKT point x^* with Lagrange multipliers μ^* and ν^*, the *critical cone* $\mathcal{C}(x^*, \mu^*, \nu^*)$ is defined as:

$$w \in \mathcal{C}(x^*, \mu^*, \nu^*) \iff \begin{cases} \nabla g_i(x^*)^T w = 0, & \\ \nabla h_i(x^*)^T w = 0, & \forall i \in \mathcal{A}(x^*) \text{ with } \nu_i^* > 0, \\ \nabla h_i(x^*)^T w \geq 0, & \forall i \in \mathcal{A}(x^*) \text{ with } \nu_i^* = 0. \end{cases}$$

Theorem 3.1.11 : Second-Order Necessary Conditions
Let x^* be a local minimizer of the NLP 3.1.1 and let LICQ be satisfied. If x^* is a KKT point with Lagrange multipliers μ^* and ν^*, then:

$$w^T \nabla^2_{xx} \mathcal{L}(x^*, \mu^*, \nu^*) w \geq 0 \quad \forall w \in \mathcal{C}(x^*, \mu^*, \nu^*).$$

Proof. See Chapter 12 in [NW06]. □

So far, we have discussed the necessary conditions that every local minimizer needs to satisfy. Now let us look at the sufficient conditions that any feasible point needs to satisfy in order to be a local minimizer.

Theorem 3.1.12 : Second-Order Sufficient Conditions
Let $x^* \in \Omega$ be a feasible KKT point of the NLP 3.1.1 with Lagrange multipliers μ^* and ν^*. Suppose that this point satisfies:

$$w^T \nabla^2_{xx} \mathcal{L}(x^*, \mu^*, \nu^*) w > 0 \quad \forall w \in \mathcal{C}(x^*, \mu^*, \nu^*), w \neq 0.$$

Then x^* is a strict local minimizer of the NLP.

Proof. See Chapter 12 in [NW06]. □

Notice that the second-order sufficient conditions do not require LICQ to hold at the solution, but they do require strict positive-definiteness of the Hessian of the Lagrangian function on the critical cone.

3.1.2 Least-squares optimization

Nonlinear least-squares optimization problems are a special class of optimization problems where the objective function is given by the sum of squares of residuals. The general form of such an optimization problem is given by:

Problem 3.1.13 : Nonlinear Least-Squares Optimization Problem
Find a solution $x \in \mathbb{R}^n$ to the optimization problem:

$$\begin{aligned} \min_x \quad & \frac{1}{2} \|f(x)\|_2^2 \\ \text{s.t.} \quad & g(x) = 0, \\ & h(x) \geq 0, \end{aligned}$$

where $f : \mathbb{R}^n \to \mathbb{R}^{m_f}$ computes the residual vector, and $g : \mathbb{R}^n \to \mathbb{R}^{m_g}$ and $h : \mathbb{R}^n \to \mathbb{R}^{m_h}$ are constraints. All the functions are assumed to be sufficiently smooth, usually $\mathcal{C}^2$ or $\mathcal{C}^3$.

We encounter optimization problems of this form in data fitting and parameter estimation where the goal is usually to minimize the distance between measurements

and model predictions under certain constraints. In fact, the parameter estimation problem that we present in Section 4.3 is a nonlinear least-squares problem with multi-experiment structure.

3.2 Numerical solution methods

Let us now look at some numerical techniques for solving nonlinear optimization problems. In this section, we present a brief introduction to the sequential quadratic programming (SQP) approach for equality-constrained nonlinear optimization problems and the generalized Gauss-Newton method for equality-constrained nonlinear least-squares problems. Since these methods are only guaranteed to converge in a local neighborhood of the true solution, we also discuss globalization strategies that can expand their area of convergence.

For more details on these methods and their extensions to inequality-constrained optimization problems, we refer the reader to the works of Nocedal and Wright [NW06] and Bock [Boc87]. In particular, for inequality-constrained problems, the interior-point algorithm and active set strategy may be of interest.

3.2.1 Sequential quadratic programming

The sequential quadratic programming (SQP) approach is one of the most effective and well-known methods for solving constrained nonlinear optimization problems [NW06]. The main idea behind the SQP method is to apply Newton's method to solve the KKT-conditions of the nonlinear optimization problem:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & g(x) = 0. \end{aligned}$$

This means that the SQP algorithm starts with an initial guess $\left(x^{(0)}, \mu^{(0)}\right)$ and computes the next iterates $\left(x^{(k)}, \mu^{(k)}\right)$, $k = 1, 2, \ldots$ by solving a sequence of quadratic programming (QP) sub-problems given by:

$$\begin{aligned} \min_{\Delta x \in \mathbb{R}^n} \quad & F(x^{(k)})\Delta x + \frac{1}{2}\Delta x^T \nabla^2_{xx} \mathcal{L}(x^{(k)}, \mu^{(k)})\Delta x \\ \text{s.t.} \quad & g(x^{(k)}) + G(x^{(k)})\Delta x = 0. \end{aligned}$$

Here, $F(x^{(k)}) \in \mathbb{R}^{m_f \times n}$ and $G(x^{(k)}) \in \mathbb{R}^{m_g \times n}$ are the Jacobian matrices of the objective function and the equality constraints respectively evaluated at the current iterate $x^{(k)}$. The Lagrange multipliers of the equality constraints at the current iterate are represented by $\mu^{(k)}$.

If LICQ holds and the Hessian of the Lagrangian $\nabla^2_{xx}\mathcal{L}(x, \mu)$ is positive definite on the tangent space of the constraints, then the QP has a unique solution. This solution then provides a search direction $\Delta x^{(k)}$ that is used to update the current iterate $x^{(k)}$ as follows:

$$x^{(k+1)} = x^{(k)} + \Delta x^{(k)}.$$

However, since the Hessian matrix of the Lagrangian is often difficult to compute in practice, the QP sub-problem is usually solved in many applications by using a quasi-Newton approximation such as the BFGS or SR1 update [NW06].

3.2.2 Generalized Gauss-Newton method

The generalized Gauss-Newton method, introduced by Bock [Boc87], is an extension of the standard Gauss-Newton method to constrained nonlinear least-squares problems of the following form:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & \frac{1}{2} \|f(x)\|_2^2 \\ \text{s.t.} \quad & g(x) = 0. \end{aligned}$$

It is an iterative algorithm that successively solves in each step a linearization of the constrained nonlinear least-squares problem. This means that the method starts at an initial guess $x^{(0)} \in \mathbb{R}^n$ and computes the next iterates $x^{(k)}$ for $k = 1, 2, \ldots$ by iteratively solving the following linearized least-squares problem:

$$\begin{aligned} \min_{\Delta x \in \mathbb{R}^n} \quad & \frac{1}{2} \left\| f(x^{(k)}) + F(x^{(k)}) \Delta x \right\|^2 \\ \text{s.t.} \quad & g(x^{(k)}) + G(x^{(k)}) \Delta x = 0 \end{aligned}$$

where $F(x^{(k)}) \in \mathbb{R}^{m_f \times n}$ and $G(x^{(k)}) \in \mathbb{R}^{m_g \times n}$ are the Jacobian matrices of the objective function and the equality constraints respectively evaluated at the current iterate $x^{(k)}$.

This linearized least-squares problem has a unique strict minimizer Δx^* as long as the Jacobian matrices satisfy the following regularity assumptions [Boc87]:

$$\text{[CQ] rank } G(x) = m_g$$
$$\text{[PD] rank } J(x) = n$$

where $J(x)$ is the full Jacobian matrix given by:

$$J(x) := \begin{pmatrix} F(x)^T & G(x)^T \end{pmatrix}^T \in \mathbb{R}^{(m_f+m_g)\times n}.$$

The first condition [CQ], known as the constraint qualification condition, ensures that the equality constraints are linearly independent. The second condition [PD], also known as the positive-definiteness condition, ensures that the combined Jacobian matrix of the objective function and the equality constraints has full column-rank.

The regularity assumptions allow us to define a linear solution operator $J^+(x)$ which takes the form:

$$J^+(x) = \begin{pmatrix} \mathbb{I} & 0 \end{pmatrix} \begin{pmatrix} F(x)^T F(x) & G(x)^T \\ G(x) & 0 \end{pmatrix}^{-1} \begin{pmatrix} F(x)^T & 0 \\ 0 & \mathbb{I} \end{pmatrix} \in \mathbb{R}^{n\times(m_f+m_g)}.$$

Using this operator, the solution Δx^* of the linearized least-squares problem is given by:

$$\Delta x^* = -J(x)^+ \phi(x)$$

where $\phi(x) = \begin{pmatrix} f(x)^T & g(x)^T \end{pmatrix}^T$.

The linear solution operator J^+ is a generalization of the Moore-Penrose pseudo-inverse, satisfying the condition $J^+ = J^+ J J^+$, and is called a *generalized inverse* [Boc87].

Thus, the solution Δx^* computed at each iteration k provides a new search direction $\Delta x^{(k)} := \Delta x^*$ for the generalized Gauss-Newton solver. The algorithm then takes a step in this direction as follows:

$$x^{(k+1)} = x^{(k)} + \Delta x^{(k)}.$$

The generalized Gauss-Newton method can be seen as a special case of the SQP method with $F(x)^T F(x)$ taken as an approximation of the Hessian of the Lagrangian [Jan15].

The logic of this method is summarized as a pseudocode in Algorithm 3.1.

Algorithm 3.1: Generalized Gauss-Newton Method

Input: objective function f and its Jacobian F
Input: equality constraints g and its Jacobian G
Input: initial guess $x^{(0)}$
Input: maximum number of iterations K

for $k = 0, 1, 2, \ldots, K-1$ **do**
  if $x^{(k)}$ satisfies convergence criteria **then**
    solution $x^* \leftarrow x^{(k)}$
    return x^*
  find the solution $\Delta x^{(k)}$ to the linearized least-squares problem:

$$\begin{aligned} \min_{\Delta x} \quad & \frac{1}{2} \left\| f(x^{(k)}) + F(x^{(k)})\Delta x \right\|^2 \\ \text{s.t.} \quad & g(x^{(k)}) + G(x^{(k)})\Delta x = 0 \end{aligned}$$

  determine step size $\alpha \in (0, 1]$ by globalization strategy
  $x^{(k+1)} \leftarrow x^{(k)} + \alpha \Delta x^{(k)}$

Output: solution x^*

3.2.3 Globalization strategies

The full-step SQP and generalized Gauss-Newton methods outlined above are only guaranteed to converge in a small local neighborhood of the true solution [NW06; Boc87]. Therefore, in order to increase the area of convergence, we need to use damped iterations to allow shorter steps as follows:

$$x^{(k+1)} = x^{(k)} + \alpha \Delta x^{(k)}, \quad \alpha \in (0, 1].$$

The step size α determines how far to go in the new search direction so that the solution is "best" approached.

In the case of an unconstrained optimization problem, we approach the solution by minimizing the objective function. This means that the step size is chosen such that a descent is ensured: $f(x^{(k+1)}) < f(x^{(k)})$.

On the other hand, for constrained optimization problems, we need to find a balance between minimizing the objective function and satisfying the constraints. This can be done, for instance, by defining a *merit function* that quantifies the progress in the

objective function and the constraints. A popular choice for the merit function is the l_1-penalty function defined as:

$$\xi(x) = f(x) + \sum_{i=1}^{m_g} \mu_i \left| g_i(x) \right|,$$

where $\mu_i \in \mathbb{R}_+$ is a positive penalty parameter corresponding to the ith equality constraint. A *line search strategy* then chooses a step size that minimizes the merit function that measures the progress in both the objective function and the constraints.

In the following, we present three different line search strategies: exact line search, Armijo backtracking, and restrictive monotonicity test. For more details on the former two, we refer the reader to the book by Nocedal and Wright [NW06]. For more details on the restrictive monotonicity test, we refer to the original paper by Bock et al. [BKS00] and the Master's thesis by Schrot [Sch19].

Exact line search

The theoretically "ideal" line search strategy for computing the step size would be to find the global minimizer of the merit function:

$$\alpha^* = \min_{\alpha} \xi(x^{(k)} + \alpha \Delta x^{(k)}).$$

This is called *exact line search*. However, this approach is generally not used in practice because it is computationally expensive and may lead to very small step sizes close to the solution.

Armijo backtracking line search

Armijo backtracking line search is an inexact line search strategy that is widely used in practice because it can find a good approximation of the global minimizer with low computational costs. This strategy requires that the step length α always produces a sufficient decrease in the merit function, which can be measured by tracking the following condition:

$$\xi(x + \alpha \Delta x) \leq \xi(x) + c\alpha \nabla \xi(x)^T \Delta x$$

where $c \in (0, 1)$ is some constant. This is called the *Armijo condition*. In practice, the constant c is usually chosen to be small, e.g., $c = 10^{-2}$ or $c = 10^{-4}$.

However, since the Armijo condition is satisfied for all sufficiently small values of α, we need to also find a way to prevent steps that are too short to ensure reasonable progress. This is done by the *backtracking approach* where the step length is initialized to be one and then reduced by a factor $0 < \rho < 1$ for a finite number of steps until the Armijo condition is satisfied. In practice, the contraction factor ρ can also be chosen dynamically in each iteration.

Restrictive monotonicity test

An alternative globalization strategy is the restrictive monotonicity test proposed by Bock et al. [BKS00] for the generalized Gauss-Newton method. This approach is a step size criterion based on the so-called *natural level function* defined as:

$$\xi(x) = \left\| J^{+}(x)\phi(x) \right\|_2^2.$$

The *restrictive monotonicity test* is then given by:

$$\|J(x)^{-1}F(x + t\Delta x)\| \leq (1 - t + t^2\omega_1(t)\|\Delta x\|)\|J(x)^{-1}F(x)\|$$

where:

$$\omega_1(t) = \sup_{0 \leq s \leq t} \frac{\|J(x + s\Delta x) - J(x)\|}{\|\Delta x\|}.$$

This condition ensures that the iterations remain within a region where the Jacobian matrix offers a valid approximation of the local shape to prevent very small step sizes and 2-cycles that are usually found with classical merit functions in ill-conditioned problems. Although there exists no global convergence proof for this approach so far, it has shown good convergence behavior in various practical applications [BKS00; Sch19].

3.3 Efficient structure exploitation

As we will later see in Section 4.3, parameter estimation problems that combine measurements from multiple independent experiments often lead to large-scale nonlinear least-squares problems of the following form:

Problem 3.3.1 : Multi-experiment nonlinear least-squares problem (MENLP)
Find solutions $x := (x_1^{\text{loc}}, x_2^{\text{loc}}, \ldots, x_N^{\text{loc}}, x^{\text{glb}}) \in \mathbb{R}^n$ to the optimization problem:

$$\begin{aligned} \min \quad & \sum_{i=1}^{N} \frac{1}{2} \left\| f(x_i^{\text{loc}}, x^{\text{glb}}) \right\|_2^2 \\ \text{s.t.} \quad & g(x_i^{\text{loc}}, x^{\text{glb}}) = 0 \qquad i = 1, \ldots, N \end{aligned}$$

where the functions f and g are defined as:

$$\begin{aligned} f &: \mathbb{R}^{n_i^{\text{loc}}} \times \mathbb{R}^{n^{\text{glb}}} \to \mathbb{R}^{m_i^{\text{obj}}} \in \mathcal{C}^2, \\ g &: \mathbb{R}^{n_i^{\text{loc}}} \times \mathbb{R}^{n^{\text{glb}}} \to \mathbb{R}^{m_i^{\text{cons}}} \in \mathcal{C}^2. \end{aligned}$$

Here, we refer to the variables $x_i^{\text{loc}} \in \mathbb{R}^{n_i^{\text{loc}}}$ as the *local optimization variables* for experiments $i = 1, \ldots, N$ because their values can be different in each experiment. Likewise, we refer to the variables $x^{\text{glb}} \in \mathbb{R}^{n^{\text{glb}}}$ as the *global optimization variables* because their values remain the same for all experiments.

In order to solve this large-scale optimization problem efficiently, Schlöder and Bock [Sch87; SB83] adapted the generalized Gauss-Newton method to exploit the block structure of the Jacobian matrix in the MENLP 3.3.1 at each iterate. Therefore, in this section, we will explain in detail how this can be done numerically. Since the numerical method was described only very concisely in the original paper, our goal is to provide a more detailed and complete description of the method similar to [Nat14].

Before we begin, let us define the following notation:

$$\begin{aligned} n^{\text{loc}} &:= n_1^{\text{loc}} + \cdots + n_N^{\text{loc}}, \\ m^{\text{cons}} &:= m_1^{\text{cons}} + \cdots + m_N^{\text{cons}}, \\ m^{\text{obj}} &:= m_1^{\text{obj}} + \cdots + m_N^{\text{obj}}. \end{aligned}$$

Now, to solve the MENLP 3.3.1 using the generalized Gauss-Newton method, we start at an initial guess for the optimization variables and then iteratively solve the following linearized least-squares problem in each step to compute the next iterates:

Problem 3.3.2 : Multi-experiment linearized least-squares problem (MELLP) Find solutions $\Delta x := (\Delta x_1^{\text{loc}}, \Delta x_2^{\text{loc}}, \ldots, \Delta x_N^{\text{loc}}, \Delta x^{\text{glb}}) \in \mathbb{R}^n$ to the following optimization problem:

$$\begin{aligned} \min \quad & \sum_{i=1}^{N} \frac{1}{2} \left\| f_i(x_i^{\text{loc}}, x^{\text{glb}}) + \begin{bmatrix} F_i^{\text{loc}} & F_i^{\text{glb}} \end{bmatrix} \begin{bmatrix} \Delta x_i^{\text{loc}} \\ \Delta x^{\text{glb}} \end{bmatrix} \right\|_2^2 \\ \text{s.t.} \quad & g_i(x_i^{\text{loc}}, x^{\text{glb}}) + \begin{bmatrix} G_i^{\text{loc}} & G_i^{\text{glb}} \end{bmatrix} \begin{bmatrix} \Delta x_i^{\text{loc}} \\ \Delta x^{\text{glb}} \end{bmatrix} = 0, \quad i = 1, \ldots, N. \end{aligned}$$

Here, the matrices $F_i^{\text{loc}}, F_i^{\text{glb}}, G_i^{\text{loc}}, G_i^{\text{glb}}$ are defined as:

$$\begin{aligned} F_i^{\text{loc}} &\equiv F_i^{\text{loc}}(x_i^{\text{loc}}, x^{\text{glb}}) := \frac{\partial f_i}{\partial x_i^{\text{loc}}} \in \mathbb{R}^{m_i^{\text{obj}} \times n_i^{\text{loc}}}, \\ F_i^{\text{glb}} &\equiv F_i^{\text{glb}}(x_i^{\text{loc}}, x^{\text{glb}}) := \frac{\partial f_i}{\partial x^{\text{glb}}} \in \mathbb{R}^{m_i^{\text{obj}} \times n^{\text{glb}}}, \\ G_i^{\text{loc}} &\equiv G_i^{\text{loc}}(x_i^{\text{loc}}, x^{\text{glb}}) := \frac{\partial g_i}{\partial x_i^{\text{loc}}} \in \mathbb{R}^{m_i^{\text{cons}} \times n_i^{\text{loc}}}, \\ G_i^{\text{glb}} &\equiv G_i^{\text{glb}}(x_i^{\text{loc}}, x^{\text{glb}}) := \frac{\partial g_i}{\partial x^{\text{glb}}} \in \mathbb{R}^{m_i^{\text{cons}} \times n^{\text{glb}}}. \end{aligned}$$

As we already saw in Section 3.2.2, this linearized least-squares problem has a unique solution when the regularity assumptions [CQ] and [PD] are satisfied. This solution can then be computed as $\Delta x = -J^+(x) \cdot \phi(x)$ where $J^+(x)$ is the generalized inverse of the Jacobian matrix $J(x)$ and $\phi(x)$ represents the functions evaluated at the current iterate x. If we further define the notation $x_i := (x_i^{\text{loc}}, x^{\text{glb}})$ for all $i = 1, \ldots, N$, we can write the Jacobian matrix $J(x)$ and the function $\phi(x)$ as follows:

$$J(x) := \overbrace{\qquad\qquad}^{n^{\text{loc}}+n^{\text{glb}} \text{ columns}} \begin{bmatrix} G_1^{\text{loc}} & 0 & 0 & \cdots & 0 & G_1^{\text{glb}} \\ F_1^{\text{loc}} & 0 & 0 & \cdots & 0 & F_1^{\text{glb}} \\ 0 & G_2^{\text{loc}} & 0 & \cdots & 0 & G_2^{\text{glb}} \\ 0 & F_2^{\text{loc}} & 0 & \cdots & 0 & F_2^{\text{glb}} \\ \vdots & & \ddots & & \vdots & \vdots \\ 0 & \cdots & \cdots & \cdots & G_N^{\text{loc}} & G_N^{\text{glb}} \\ 0 & \cdots & \cdots & \cdots & F_N^{\text{loc}} & F_N^{\text{glb}} \end{bmatrix} \Bigg\updownarrow m^{\text{cons}}+m^{\text{obj}} \text{ rows}$$

$$\phi(x) := \begin{bmatrix} g_1(x_1) \\ f_1(x_1) \\ g_2(x_2) \\ f_2(x_2) \\ \vdots \\ g_N(x_N) \\ f_N(x_N) \end{bmatrix} \Bigg\updownarrow m^{\text{cons}}+m^{\text{obj}} \text{ rows}.$$

When we assume that [CQ] holds true, this means that the constraints are linearly independent and the following matrix with $m^{\text{cons}} \leq n^{\text{loc}} + n^{\text{glb}}$ has full row-rank:

$$G = \overbrace{\begin{bmatrix} G_1^{\text{loc}} & 0 & 0 & \cdots & 0 & G_1^{\text{glb}} \\ 0 & G_2^{\text{loc}} & 0 & \cdots & 0 & G_2^{\text{glb}} \\ \vdots & & \ddots & & \vdots & \vdots \\ 0 & \cdots & \cdots & \cdots & G_N^{\text{loc}} & G_N^{\text{glb}} \end{bmatrix}}^{n^{\text{loc}}+n^{\text{glb}} \text{ columns}} \Bigg\updownarrow \begin{matrix} \\ \\ \\ \\ m^{\text{cons}} \text{ rows} \end{matrix}.$$

Furthermore, when we assume that Jacobian matrix $J(x)$ satisfies [PD], this means that the columns of $J(x)$ are linearly independent. This is only possible if the number of columns of this matrix is not more than the number of rows. In this case, we have $m^{\text{obj}} + m^{\text{cons}} \geq n^{\text{loc}} + n^{\text{glb}}$.

Now, in order to efficiently compute Δx, we will exploit the block structure of the Jacobian matrix and turn the sparse matrix $J(x)$ into upper triangular form in three steps as follows:

Step I: Transform local sub-matrices into upper triangular form

The Jacobian matrix $J(x)$ is made up of dense blocks of the form

$$J_i^{\text{loc}} := \begin{matrix} \\ \begin{bmatrix} G_i^{\text{loc}} \\ F_i^{\text{loc}} \end{bmatrix} \\ \\ \end{matrix} \begin{matrix} n_i^{\text{loc}} \\ \\ m_i^{\text{cons}} \\ m_i^{\text{obj}} \end{matrix} \quad \forall i = 1, \ldots, N.$$

We will now transform each of these blocks into upper triangular form as follows:

① Use QR decomposition with column pivoting on the matrix $G_i^{\text{loc}} \in \mathbb{R}^{m_i^{\text{cons}} \times n_i^{\text{loc}}}$:

$$\tilde{G}_i^{\text{loc}} := G_i^{\text{loc}} P_i^{\text{loc}} = U_i R_{i1}$$

where:

- $P_i^{\text{loc}} \in \mathbb{R}^{n_i^{\text{loc}} \times n_i^{\text{loc}}}$ is a permutation matrix,
- $U_i \in \mathbb{R}^{m_i^{\text{cons}} \times m_i^{\text{cons}}}$ is an orthogonal matrix,
- $R_{i1} \in \mathbb{R}^{m_i^{\text{cons}} \times n_i^{\text{loc}}}$ is an upper triangular matrix.

The matrix R_{i1} can be further distinguished into three cases:

a) If $m_i^{\text{cons}} > n_i^{\text{loc}}$, then $R_{i1} := \begin{matrix} n_i^{\text{loc}} \\ \left[\begin{matrix} R_{i11} \\ 0 \end{matrix}\right] \begin{matrix} n_i^{\text{loc}} \\ m_i^{\text{cons}} - n_i^{\text{loc}} \end{matrix} \end{matrix}$.

b) If $m_i^{\text{cons}} < n_i^{\text{loc}}$, then $R_{i1} := \begin{matrix} m_i^{\text{cons}} & n_i^{\text{loc}} - m_i^{\text{cons}} \\ \left[\begin{matrix} R_{i11} & R_{i12} \end{matrix}\right] & m_i^{\text{cons}} \end{matrix}$.

c) If $m_i^{\text{cons}} = n_i^{\text{loc}}$, then $R_{i1} := \begin{matrix} n_i^{\text{loc}} \\ \left[R_{i11}\right] \end{matrix} m_i^{\text{cons}}$.

In all three cases, R_{i11} is an invertible square upper triangular matrix.

(2) Permute the columns of the matrix $F_i^{\text{loc}} \in \mathbb{R}^{m_i^{\text{obj}} \times n_i^{\text{loc}}}$ to match G_i^{loc}:

$$\tilde{F}_i^{\text{loc}} := F_i^{\text{loc}} P_i^{\text{loc}}.$$

(3) Compute the orthogonal matrix L_i corresponding to the columns of F_i^{loc} that have already been resolved by QR decomposition in the previous step:

a) If $m_i^{\text{cons}} \geq n_i^{\text{loc}}$, then $L_i := \tilde{F}_i^{\text{loc}} R_{i11}^{-1} \in \mathbb{R}^{m_i^{\text{obj}} \times n_i^{\text{loc}}}$.

b) If $m_i^{\text{cons}} < n_i^{\text{loc}}$, then $L_i := \tilde{F}_{i1}^{\text{loc}} R_{i11}^{-1} \in \mathbb{R}^{m_i^{\text{obj}} \times m_i^{\text{cons}}}$ where:

$$\tilde{F}_i^{\text{loc}} := \begin{matrix} m_i^{\text{cons}} & n_i^{\text{loc}} - m_i^{\text{cons}} \\ \left[\begin{matrix} \tilde{F}_{i1}^{\text{loc}} & \tilde{F}_{i2}^{\text{loc}} \end{matrix}\right] & m_i^{\text{obj}} \end{matrix}.$$

(4) If there are any columns of $\tilde{F}_i^{\text{loc}}$ that remain unresolved after step (3), use QR decomposition on these columns. This will be necessary only in the case where $m_i^{\text{cons}} < n_i^{\text{loc}}$. In this case, we get:

$$\tilde{F}_{i2}^{\text{loc}} - L_i R_{i12} = Q_i R_{i2} = Q_i \begin{matrix} n_i^{\text{loc}} - m_i^{\text{cons}} \\ \left[\begin{matrix} R_{i21} \\ 0 \end{matrix}\right] \begin{matrix} n_i^{\text{loc}} - m_i^{\text{cons}} \\ m_i^{\text{obj}} + m_i^{\text{cons}} - n_i^{\text{loc}} \end{matrix} \end{matrix}$$

where $Q_i \in \mathbb{R}^{m_i^{\text{obj}} \times m_i^{\text{obj}}}$ is an orthogonal matrix and $R_{i2} \in \mathbb{R}^{m_i^{\text{obj}} \times \left(n_i^{\text{loc}} - m_i^{\text{cons}}\right)}$ is a square upper triangular matrix.

⑤ Combine the results from steps ① to ④ to get the following decomposition:

a) If $m_i^{\text{cons}} > n_i^{\text{loc}}$, then:

$$J_i^{\text{loc}} P_i^{\text{loc}} = \begin{bmatrix} \tilde{G}_i^{\text{loc}} \\ \tilde{F}_i^{\text{loc}} \end{bmatrix} = \begin{matrix} m_i^{\text{cons}} \\ m_i^{\text{obj}} \end{matrix} \overbrace{\left[\begin{array}{cc} U_i & 0 \\ L_i & 0 \end{array}\right.}^{\begin{matrix} m_i^{\text{cons}} & m_i^{\text{obj}} \end{matrix}} \overbrace{\left.\begin{array}{c} \\ \mathbb{I} \end{array}\right]}^{\begin{matrix} \\ m_i^{\text{obj}} \end{matrix}} \overbrace{\begin{bmatrix} R_{i11} \\ 0 \end{bmatrix}}^{n_i^{\text{loc}}} \begin{matrix} n_i^{\text{loc}} \\ m_i^{\text{obj}} + m_i^{\text{cons}} - n_i^{\text{loc}} \end{matrix}.$$

b) If $m_i^{\text{cons}} < n_i^{\text{loc}}$, then:

$$J_i^{\text{loc}} P_i^{\text{loc}} = \begin{bmatrix} \tilde{G}_i^{\text{loc}} \\ \tilde{F}_i^{\text{loc}} \end{bmatrix} = \begin{matrix} m_i^{\text{cons}} \\ m_i^{\text{obj}} \end{matrix} \overbrace{\left[\begin{array}{cc} U_i & 0 \\ L_i & \mathbb{I} \end{array}\right]}^{\begin{matrix} m_i^{\text{cons}} & m_i^{\text{obj}} \end{matrix}} \overbrace{\left[\begin{array}{cc} R_{i11} & R_{i12} \\ 0 & R_{i21} \\ 0 & 0 \end{array}\right]}^{\begin{matrix} m_i^{\text{cons}} & n_i^{\text{loc}} - m_i^{\text{cons}} \end{matrix}} \begin{matrix} m_i^{\text{cons}} \\ n_i^{\text{loc}} - m_i^{\text{cons}} \\ m_i^{\text{obj}} + m_i^{\text{cons}} - n_i^{\text{loc}} \end{matrix}.$$

c) If $m_i^{\text{cons}} = n_i^{\text{loc}}$, then:

$$J_i^{\text{loc}} P_i^{\text{loc}} = \begin{bmatrix} \tilde{G}_i^{\text{loc}} \\ \tilde{F}_i^{\text{loc}} \end{bmatrix} = \begin{matrix} m_i^{\text{cons}} \\ m_i^{\text{obj}} \end{matrix} \overbrace{\left[\begin{array}{cc} U_i & 0 \\ L_i & \mathbb{I} \end{array}\right]}^{\begin{matrix} m_i^{\text{cons}} & m_i^{\text{obj}} \end{matrix}} \overbrace{\begin{bmatrix} R_{i11} \\ 0 \end{bmatrix}}^{n_i^{\text{loc}}} \begin{matrix} m_i^{\text{cons}} \\ m_i^{\text{obj}} \end{matrix}.$$

In this way, for every experiment $i = 1, \ldots, N$, we can decompose the local submatrix J_i^{loc} into an orthogonal matrix $V_i \in \mathbb{R}^{(m_i^{\text{cons}} + m_i^{\text{obj}}) \times (m_i^{\text{cons}} + m_i^{\text{obj}})}$ and an upper triangular matrix $R_i \in \mathbb{R}^{(m_i^{\text{cons}} + m_i^{\text{obj}}) \times n_i^{\text{loc}}}$ to get:

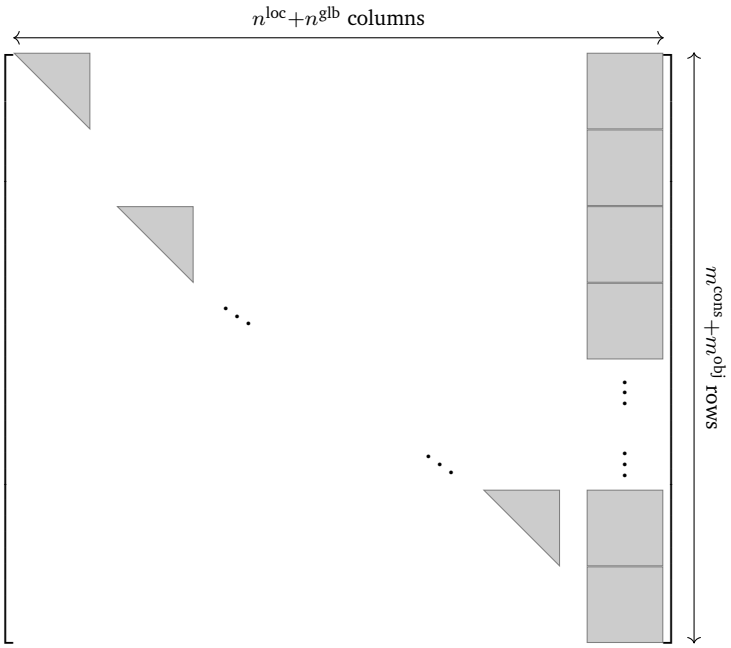
$$J_i^{\text{loc}} P_i^{\text{loc}} = V_i R_i, \quad \forall i = 1, \ldots, N.$$

Combining all the orthogonal matrices V_i into a large diagonal invertible block matrix $V \in \mathbb{R}^{(m^{\text{cons}} + m^{\text{obj}}) \times (m^{\text{cons}} + m^{\text{obj}})}$ and combining all the permutation matrices P_i^{loc} into a large diagonal invertible block matrix $P_{c1} \in \mathbb{R}^{n \times n}$, we get:

$$V = \begin{bmatrix} V_1 & & & \\ & V_2 & & \\ & & \ddots & \\ & & & V_N \end{bmatrix}, \qquad P_{c1} = \begin{bmatrix} P_1^{\text{loc}} & & & & \\ & P_2^{\text{loc}} & & & \\ & & \ddots & & \\ & & & P_N^{\text{loc}} & \\ & & & & \mathbb{I}_{n^{\text{glb}}} \end{bmatrix}.$$

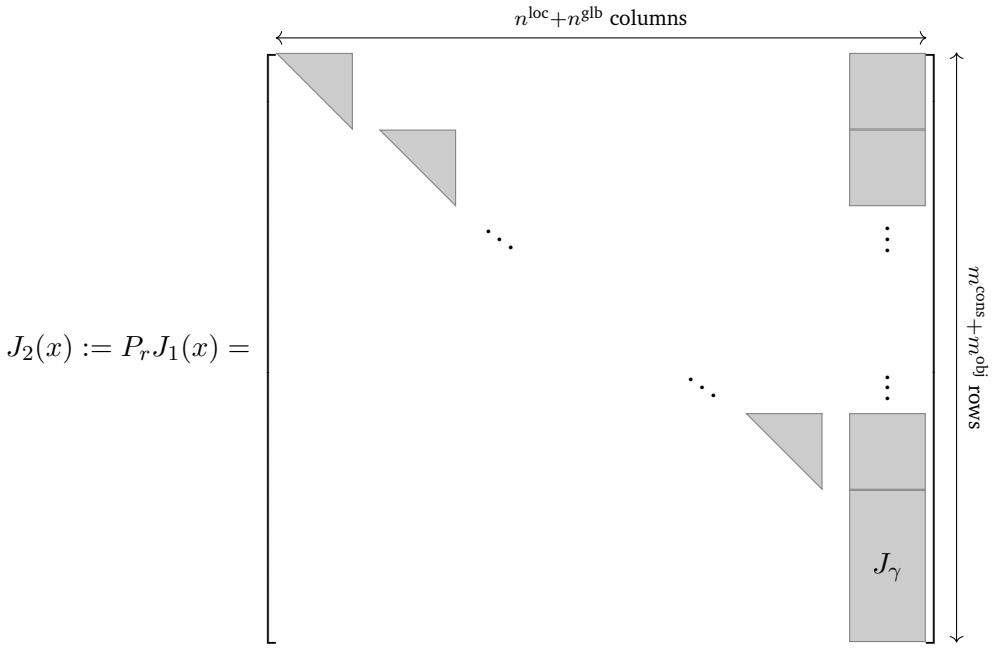
This allows us to transform the Jacobian matrix $J(x)$ into the following structure:

$$J_1(x) := V^{-1} J(x) P_{c1} =$$



Step II: Move the almost empty rows to the bottom

From the output of Step I, we see that the first n^{loc} columns of the transformed Jacobian matrix $J_1(x)$ almost have an upper triangular structure, broken only by the rows that are non-zero in the last n^{glb} columns of the matrix. So, in this step, we will define a permutation matrix $P_r \in \mathbb{R}^{(m^{\text{cons}}+m^{\text{obj}})\times(m^{\text{cons}}+m^{\text{obj}})}$ that rearranges the rows of $J_1(x)$ such that all the small local upper triangular matrices are diagonally stacked together to form a large upper triangular matrix. This means that we move the rows of $J_1(x)$ that are only non-zero in the last n^{glb} columns to the bottom of the matrix to get the following structure:



Now let us denote the dense sub-matrix formed by the last few rows, where the only non-zero entries are in the last n^{glb} columns as $J_\gamma \in \mathbb{R}^{\left(m^{\text{obj}}+m^{\text{cons}}-n^{\text{loc}}\right)\times n^{\text{glb}}}$. Then, we define the permutation matrix P_r such that the first few rows of J_γ correspond to the constraints and the remaining rows of J_γ correspond to the least-squares terms.

Step III: Transform J_γ into upper triangular form

The transformation of the Jacobian matrix $J(x)$ into upper triangular form is almost complete. All that remains now is to transform J_γ also into upper triangular form.

For this, we once again use QR decomposition with column pivoting as follows:

$$J_\gamma P_\gamma = U_\gamma R = U_\gamma \begin{bmatrix} R_\gamma \\ 0 \end{bmatrix} \begin{matrix} n^{\text{glb}} \\ m^{\text{obj}}+m^{\text{cons}}-n^{\text{loc}}-n^{\text{glb}} \end{matrix}$$

where:

- $P_\gamma \in \mathbb{R}^{n^{\text{glb}}\times n^{\text{glb}}}$ is a permutation matrix,
- $U_\gamma \in \mathbb{R}^{\left(m^{\text{obj}}+m^{\text{cons}}-n^{\text{loc}}\right)\times\left(m^{\text{obj}}+m^{\text{cons}}-n^{\text{loc}}\right)}$ is an orthogonal matrix, and
- $R \in \mathbb{R}^{\left(m^{\text{obj}}+m^{\text{cons}}-n^{\text{loc}}\right)\times n^{\text{glb}}}$ is a square upper triangular matrix.

Using the decomposition, we can then define a permutation matrix $P_{c2} \in \mathbb{R}^{n \times n}$ and an orthogonal matrix $T \in \mathbb{R}$ such that:

$$P_{c2} := \begin{bmatrix} \overset{n^{\text{loc}}}{\mathbb{I}} & \overset{n^{\text{glb}}}{0} \\ 0 & P_\gamma \end{bmatrix} \begin{matrix} n^{\text{loc}} \\ n^{\text{glb}} \end{matrix} \quad \text{and} \quad T := \begin{bmatrix} \overset{n^{\text{loc}}}{\mathbb{I}} & \overset{n^{\text{glb}}}{0} \\ 0 & U_\gamma \end{bmatrix} \begin{matrix} n^{\text{loc}} \\ n^{\text{glb}} \end{matrix}.$$

Then, we can transform the Jacobian matrix J_2 from Step II into an upper triangular matrix $\tilde{J}$ as follows:

$$\tilde{J}(x) := T^{-1} J_2(x).$$

Together, the complete transformation of the Jacobian matrix $J(x)$ into upper triangular form can be expressed as:

$$\tilde{J}(x) = T^{-1} P_r V^{-1} J P_{c1} P_{c2}$$

The same transformations are also applied to the function values $\phi(x)$ to get:

$$\tilde{\phi}(x) := T^{-1} P_r V^{-1} \phi(x).$$

Compute solution Δx of the linearized least-squares problem

Using the transformations from the previous steps, we can write the Jacobian matrix and function values as follows:

(a) If $m^{\text{cons}} \geq n^{\text{loc}}$, then:

$$\tilde{J}(x) = \begin{bmatrix} R_{111} & & & \tilde{S}_{11} \\ & R_{211} & & \tilde{S}_{21} \\ & & \ddots & \vdots \\ & & R_{N11} & \tilde{S}_{N1} \\ & & & R_\gamma \\ & & & 0 \end{bmatrix}, \qquad \tilde{\phi}(x) = \begin{bmatrix} \tilde{\phi}_{11} \\ \tilde{\phi}_{21} \\ \vdots \\ \tilde{\phi}_{N1} \\ \tilde{\phi}_{\gamma 1} \\ \tilde{\phi}_{\gamma 2} \end{bmatrix}.$$

(b) If $m^{\text{cons}} < n^{\text{loc}}$, then:

$$\tilde{J}(x) = \begin{bmatrix} R_{111} & R_{112} & & & & & & \tilde{S}_{11} \\ 0 & R_{122} & & & & & & \tilde{S}_{12} \\ & & R_{211} & R_{212} & & & & \tilde{S}_{21} \\ & & 0 & R_{222} & & & & \tilde{S}_{22} \\ & & & & \ddots & & & \vdots \\ & & & & & R_{N11} & R_{N12} & \tilde{S}_{N1} \\ & & & & & 0 & R_{N22} & \tilde{S}_{N2} \\ & & & & & & & R_\gamma \\ & & & & & & & 0 \end{bmatrix}, \qquad \tilde{\phi}(x) = \begin{bmatrix} \tilde{\phi}_{11} \\ \tilde{\phi}_{12} \\ \tilde{\phi}_{21} \\ \tilde{\phi}_{22} \\ \vdots \\ \tilde{\phi}_{N1} \\ \tilde{\phi}_{N2} \\ \tilde{\phi}_{\gamma 1} \\ \tilde{\phi}_{\gamma 2} \end{bmatrix}.$$

This allows us to express the solution Δx of the MELLP 3.3.2 as follows:

$$\Delta x^{\text{glb}} = -P_\gamma R_\gamma^{-1} \tilde{\phi}_{\gamma 1}$$

$$\Delta x_i^{\text{loc}} = \begin{cases} -P_i^{\text{loc}} R_{i11}^{-1} \left(\tilde{S}_{i1} \Delta x^{\text{glb}} + \tilde{\phi}_{i1} \right) & \text{if } m^{\text{cons}} > n^{\text{loc}} \\ -P_i^{\text{loc}} \begin{bmatrix} R_{i11} & R_{i12} \\ 0 & R_{i22} \end{bmatrix}^{-1} \left(\begin{bmatrix} \tilde{S}_{i1} \\ \tilde{S}_{i2} \end{bmatrix} \Delta x^{\text{glb}} + \begin{bmatrix} \tilde{\phi}_{i1} \\ \tilde{\phi}_{i2} \end{bmatrix} \right) & \text{if } m^{\text{cons}} \leq n^{\text{loc}} \end{cases} \qquad \forall i = 1, \ldots, N.$$

3.4 Derivative computation

The numerical optimization methods that we consider in this thesis require the computation of derivatives. Although, in theory, these derivatives can be computed analytically, this is usually tedious, error-prone, and unnecessary in practice. Therefore, in this section, we discuss some standard approaches for accurately and efficiently computing derivatives in derivative-based optimization methods.

Finite difference approximation

One of the most straightforward ways to approximate the derivatives is to use finite differences. This approach is based on the definition of the derivative as the limit of the difference quotient. The finite difference approximation of a function

$f \in \mathcal{C}^2 : \mathbb{R}^n \to \mathbb{R}$ at a point $x \in \mathbb{R}^n$ can be computed using the forward, backward, or central difference formula. The *forward difference formula* is given by

$$\frac{df}{dx_i}(x) \approx \frac{f(x + he_i) - f(x)}{h},$$

the *backward difference formula* is given by

$$\frac{df}{dx_i}(x) \approx \frac{f(x) - f(x - he_i)}{h},$$

and the *central difference formula* is given by

$$\frac{df}{dx_i}(x) \approx \frac{f(x + he_i) - f(x - he_i)}{2h},$$

where h is a small perturbation and e_i is a vector of zeros with a one at the ith component for some $i \in \{1, \ldots, n\}$. The accuracy of this approximation depends on the choice of h. If h is too large, the approximation is inaccurate, and if h is too small, the approximation is sensitive to round-off errors. Therefore, the optimal choice of h is usually problem-dependent.

Although the finite difference approximation is simple to implement, it can be computationally expensive. Therefore, we next introduce a more efficient and accurate approach for computing the derivatives called automatic differentiation.

Automatic differentiation

Automatic differentiation, also known as *algorithmic differentiation* (AD), is a method for computing the exact derivatives of a function — up to machine precision — by applying the chain rule of calculus repeatedly. The key insight behind AD is that functions can be decomposed into elementary operations, each with a known derivative. This means that the derivative of the composite function can be obtained by systematically applying the chain rule to these operations.

For example, let us consider the task of computing the Jacobian matrix for a function $f \in \mathcal{C}^2 : \mathbb{R}^n \to \mathbb{R}^m$ at a point $x \in \mathbb{R}^n$. AD offers two primary modes for this purpose: the forward mode and the reverse mode. In the forward mode, the Jacobian matrix is constructed column-wise by computing all the partial derivatives of the function f with respect to one input variable x_i (for $i = 1, \ldots, n$). This means that the forward mode propagates the derivative information from the inputs to the outputs, making it particularly efficient for functions with few inputs and many outputs. In contrast, in the reverse mode, the derivatives are constructed row-wise by computing

the partial derivatives of each function component f_j with respect to all the input variables x (for $j = 1, \ldots, m$). This means that the reverse mode propagates the derivative information from the outputs to the inputs, making it particularly efficient for functions with many inputs and few outputs.

For a more detailed discussion on AD, we refer the reader to the book by Griewank and Walther [GW08].

Part II

Parameter Estimation and Optimum Experimental Design

4 Parameter Estimation with Bifurcation Points

In this chapter, we introduce our general framework for parameter estimation using measurements of bifurcation points. We begin our discussion by outlining the motivation for our approach and presenting related previous works. We then formulate the parameter estimation problem as a nonlinear least squares problem with a focus on experimental systems that show bistability or oscillations. Finally, we also present a numerical strategy for solving the parameter estimation problem with a detailed discussion on how suitable initial guesses can be obtained to improve convergence.

4.1 Motivation

Traditionally, parameter estimation methods rely on time-series data, where a function of some state variables, known as observables, are measured at multiple time points to calibrate the mathematical models. However, this approach can be difficult to use when the dynamical system operates on a much faster timescale than the sampling rate of our measurements. In some cases, it may even be impractical or prohibitively expensive to measure the observables at multiple time points. These challenges prompt the question: what alternative measurements can we use to calibrate our models?

This thesis was inspired by one of the case studies in the work of Schlöder and Bock [SB83; Sch87] from the 1980s. Although the focus of their research at the time was on developing an efficient numerical method to solve a multi-experiment parameter estimation problem, one of the examples that they used to illustrate the performance of their method was particularly interesting for a different reason. In a case study, Schlöder and Bock tried to estimate the parameters of the Noyes-Field-Thompson model [NFT71] that describes the Belousov-Zhabotinsky chemical reaction [Bel59; Zha64]. The measurement data that they used for this purpose were the externally controlled inflow concentrations of the chemicals at switching points (or saddle-node bifurcation points). To the best of our knowledge, this was one of the earliest

attempts at using bifurcation points to estimate model parameters. While their work emphasized the efficiency gains from structure-exploitation in the multi-experiment framework of the generalized Gauss-Newton solver, it did not explore the broader potential of bifurcation point measurements in parameter estimation. Therefore, our work adapts and extends their case study to establish a general framework for parameter estimation using bifurcation point measurements.

The idea of experimentally measuring bifurcation points is not new in itself. In fact, there are numerous experiments in fields such as biochemistry, chemical engineering, ecology and semiconductor lasers where qualitative changes in system behavior due to saddle-node or Hopf bifurcations have been observed and recorded[1]. Therefore, these applications serve as the motivation for our parameter estimation framework, which is based on the assumption that external controls can be manipulated to induce qualitative changes in system dynamics, and that the control values at the observed bifurcation points can be used as measurement data.

4.2 Related works

So far, we have not found any published work that offers a general framework for parameter estimation using bifurcation points as proposed in this thesis. Nevertheless, there are a few related studies that have explored the use of bifurcation theory in inverse problems from a different perspective. In this section, we would like to provide a brief overview of some of these works.

In the field of chemical engineering, there are a variety of experiments with variable input-controls that allow the detection of bifurcation points. Although we will discuss these experiments in further detail in Chapter 7, in this section, we would like to present some studies that have used measurements of bifurcation points detected with these experiments to calibrate their mathematical models. For example, Harold and Luss [HL87] used measurements of the surrounding gas temperature and the pellet temperature at bifurcation points to estimate the parameters of a mathematical model for ethane oxidation in a single Pt/Al_2O_3 pellet. In another study, Shanks and Bailey [SB87] used steady state, bifurcation, and frequency data to fit two different models of carbon-monoxide oxidation over supported silver. They estimated the parameters over multiple stages starting with steady state measurements, followed by step-response experiments. In the final step, they also included the measured values for Hopf bifurcation points and the frequency of the corresponding limit

[1] We will discuss some of these applications in more detail in Chapter 7.

cycles as the gains to the CO and O_2 mass flow controllers were experimentally varied.

For systems where it is possible to measure the steady state concentrations of the species, these concentrations can be tracked for varying values of an external control to obtain a one-parameter bifurcation diagram. This approach was used in the specific context of bistable biochemical networks by Otero-Muras et al. [OYS14]. They used Monte-Carlo methods to estimate parameters of chemical reaction network models with saddle-node bifurcation points from their dose-response curves. In a similar vein, Cedersund et al. [CK05] have discussed how prior knowledge about the location of Hopf bifurcation points can provide additional information to the time-series data used in parameter estimation problems.

We have so far presented studies that have used some information about bifurcations to calibrate mathematical models. On a related note, there have also been studies that estimate parameters of a mathematical model with the aim of controlling the location of the bifurcation. For instance, in order to prevent the static voltage collapse caused by saddle-node bifurcations in power systems, Canizares et al. [Can98] developed a method to estimate the parameters of a mathematical model such that the distance to the saddle-node bifurcation point is maximized. Similarly, Boulle et al. [BFR23] developed a numerical approach to determine parameters of a mathematical model to control various properties of Hopf bifurcation points such as its location or the frequency of the limit cycle.

4.3 Problem formulation

Let us now formulate the parameter estimation problem as a constrained nonlinear least-squares problem. We will start by introducing the general formulation of this problem that employs measurements of steady-state bifurcation points to fit a mathematical model. Then, we will look at how the parameter estimation problem can be specifically formulated for two different types of bifurcations: saddle-node bifurcations and Hopf bifurcations.

4.3.1 General formulation

Consider a system of ordinary differential equations given by:

$$\begin{aligned} \dot{y}(t) &= \psi(y(t), q, p), \quad \psi \in \mathcal{C}^2 \\ y(0) &= y_0(q, p) \end{aligned} \tag{4.1}$$

where $t \in \mathbb{R}$ represents time, $y \in \mathbb{R}^{n_y}$ the model states, $q \in \mathbb{R}^{n_q}$ the applied external controls, and $p \in \mathbb{R}^{n_p}$ the model parameters. We assume that this system is autonomous with a right-hand side function ψ that may depend explicitly on the current model state, the applied external controls, and the model parameters. The initial state y_0 may also depend on the applied controls and the model parameters. Together equations (4.1) form a mathematical model that is assumed to describe a real-world dynamical system.

The goal of our parameter estimation problem is to estimate the values of the parameters p with the help of measured bifurcation points. This means that our measurement data consist of the values of the applied external controls at which we experimentally observe a sudden qualitative change in the system dynamics[2].

Suppose we have two external controls that we can vary in our experiments. Then, we can run M experiments, where, in each experiment $i = 1, \ldots, M$, one of the two controls is kept fixed at a value $\tilde{q}_{2_i}$, while the other control is varied until a bifurcation is observed at some value $\tilde{q}_{1_i}$. This results in a set of $N \geq M$ pairs of control values $(\tilde{q}_{1_i}, \tilde{q}_{2_i})_{i=1,\ldots,N}$ that correspond to experimentally observed bifurcation points. We will then use this set of bifurcation points as the measurement data to estimate the parameters of the mathematical model.

As with any measurements in real-world systems, the values of the applied external controls are subject to measurement errors and, therefore, not known exactly. As a result, for every pair of measured controls $i = 1, \ldots, N$, we assume that the measurement pair $(\tilde{q}_{1_i}, \tilde{q}_{2_i})$ is affected by independent, additive normally distributed measurement errors with mean zero and standard deviation $(\sigma_{1_i}, \sigma_{2_i})$. This means that the measured values of the bifurcation points are then given by:

$$\begin{aligned} \tilde{q}_{1_i} &= \bar{q}_{1_i} + \epsilon_{1_i}, \quad \epsilon_{1_i} \sim \mathcal{N}(0, \sigma_{1_i}), \\ \tilde{q}_{2_i} &= \bar{q}_{2_i} + \epsilon_{2_i}, \quad \epsilon_{2_i} \sim \mathcal{N}(0, \sigma_{2_i}), \end{aligned}$$

where $\bar{q}_{1_i}$ and $\bar{q}_{2_i}$ are the true bifurcation points, and ϵ_{1_i} and ϵ_{2_i} are the measurement errors. We can estimate the standard deviation $(\sigma_{1_i}, \sigma_{2_i})$ for each $i = 1, \ldots, N$ by

[2] We will explain how this can be done specifically for saddle-node and Hopf bifurcations in more detail in the next sections.

repeating the experiment multiple times and calculating the standard deviation of the measured values.

Once we have the measurements, our objective is to determine the parameter values $p \in \mathbb{R}^{n_p}$ that theoretically predict a set of bifurcation points aligning as closely as possible with the measured bifurcation points. This parameter estimation problem can be formulated as a constrained nonlinear least-squares problem as follows:

Problem 4.3.1 : Parameter Estimation using Bifurcation Points
Find the values of the model parameters $p \in \mathbb{R}^{n_p}$ and also, for $i = 1, \ldots, N$, the steady states $y_i \in \mathbb{R}^{n_y}$, controls $q_i \in \mathbb{R}^2$ and additional variables $z_i \in \mathbb{R}^{n_z}$ such that they solve the following optimization problem:

$$\min \quad \sum_{i=1}^{N} \left(\frac{\tilde{q}_{1_i} - q_{1_i}}{\sigma_{1_i}} \right)^2 + \left(\frac{\tilde{q}_{2_i} - q_{2_i}}{\sigma_{2_i}} \right)^2$$

$$\text{s.t.} \quad \begin{cases} \psi(y_i, q_i, p) = 0 \\ \zeta(y_i, q_i, z_i, p) = 0 \end{cases} \qquad \forall\, i = 1, \ldots, N,$$

where $\psi \in \mathcal{C}^2$ is the right-hand side function of the mathematical model (4.1) and $\zeta \in \mathcal{C}^2$ is a function that characterizes its bifurcation point.

In theory, the problem formulation could be easily extended to three or more controls, where all but one control are kept fixed while the free control is varied until a bifurcation is observed. However, most of the experiments we have seen in the literature (see Chapter 7) only track two external controls. Therefore, for the sake of simplicity, we will only consider the case of two external controls in this thesis.

The choice of the additional variables z_i, for $i = 1, \ldots, N$, and the function ζ in problem 4.3.1 depend upon the type of bifurcation that is being measured. In Section 1.3, we introduced two different types of bifurcations: saddle-node bifurcations and Hopf bifurcations. Therefore, in the following sections, we will look at how this parameter estimation problem can be specifically formulated for these two types of bifurcations.

4.3.2 Saddle-node bifurcations

Before formulating the parameter estimation problem using saddle-node bifurcation points, let us first look at how saddle-node bifurcations can be experimentally observed and measured.

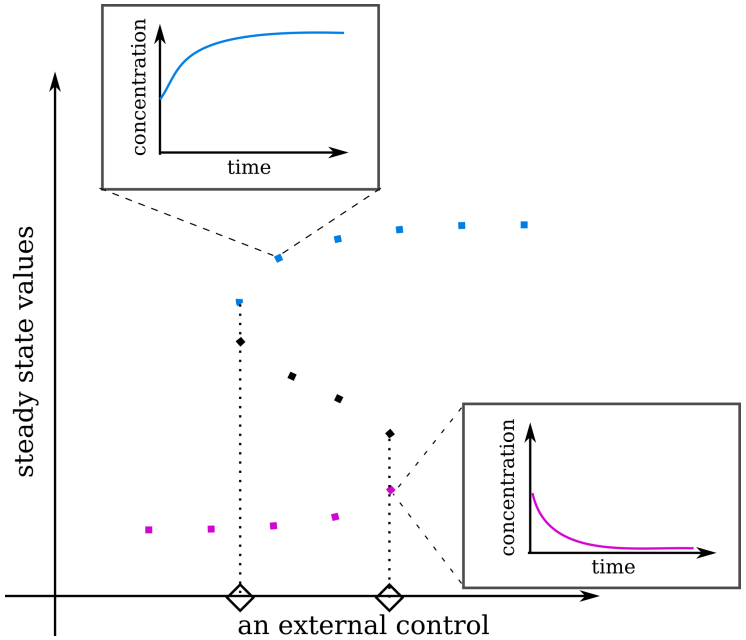


Fig. 4.1.: A schematic diagram showing how a pair of saddle-node bifurcation points can be experimentally measured in bistable systems. The blue points and pink points represent the two stable steady state curves. The black points represent the unstable manifold. The black diamonds represent the two measured values of the applied external control at the two bifurcation points. The other control is kept fixed throughout.

Consider an experimental setup for a dynamical system that can exhibit bistable behavior. We assume that we have two external controls that can be experimentally varied. We begin by keeping both the controls fixed and letting the system reach a stable equilibrium. Suppose our initial stable equilibrium is represented by the pink points in Figure 4.1. Now, we can continuously vary one of the controls such that we let the experimental system settle into a stable equilibrium for each value of this control. For small continuous changes in the control value, we expect only small changes in the stable equilibrium. We can see this in Figure 4.1, where, as the value of the applied control is increased, the system initially continues to settle into the pink steady state. However, as we vary the value of the applied control, at some point, we may observe a sudden change in the qualitative dynamics of the system and notice that the experimental system settles to a different stable equilibrium. This new stable equilibrium is illustrated in Figure 4.1 by the blue points. Now, if we

continue varying the applied control along the same direction, we will see that the experimental system continues to settle to this new stable equilibrium. This means that we have experimentally observed a saddle-node bifurcation point. The values of the applied controls at which this bifurcation was detected gives us a measurement pair, which is represented by the black diamond on the right in Figure 4.1. In order to obtain more such measurement pairs, we can fix the other external control at a different value and repeat this process.

Suppose we have measured a set of N pairs of external control values $(\tilde{q}_{1_i}, \tilde{q}_{2_i})_{i=1,\dots,N}$ at which saddle-node bifurcations were detected. In order to estimate the parameter values $p \in \mathbb{R}^{n_p}$ using these measurements, we formulate the following parameter estimation problem:

Problem 4.3.2 : Parameter Estimation using Saddle-Node Bifurcation Points
Find the values of the model parameters $p \in \mathbb{R}^{n_p}$, and for $i = 1, \dots, N$, the steady states $y_i \in \mathbb{R}^{n_y}$, applied controls $q_i \in \mathbb{R}^2$, and additional variables $h_i \in \mathbb{R}^{n_y}$ such that they solve the following optimization problem:

$$\begin{aligned} \min \quad & \sum_{i=1}^{N} \left(\frac{\tilde{q}_{1_i} - q_{1_i}}{\sigma_{1_i}} \right)^2 + \left(\frac{\tilde{q}_{2_i} - q_{2_i}}{\sigma_{2_i}} \right)^2 \\ \text{s.t.} \quad & \begin{cases} \psi(y_i, q_i, p) = 0, \\ \dfrac{\partial \psi(y_i, q_i, p)}{\partial y} h_i = 0, \\ h_i^T h_i - 1 = 0, \end{cases} \qquad \forall\, i = 1, \dots, N. \end{aligned}$$

As we saw in Section 1.3.1, a saddle-node bifurcation point is characterized by the condition that the Jacobian matrix $\dfrac{\partial}{\partial y}\psi(y, q, p)$ of the right-hand side function ψ has a zero eigenvalue. Therefore, the additional variables h_i correspond to the eigenvector of the Jacobian matrix for each $i = 1, \dots, N$.

4.3.3 Hopf bifurcations

Let us now consider an experimental system that can exhibit oscillations through Hopf bifurcations. As we already saw in Section 1.3.2, Hopf bifurcations can occur in one of two forms: subcritical and supercritical. In this thesis, we will focus our attention on supercritical Hopf bifurcations that result in stable oscillations.

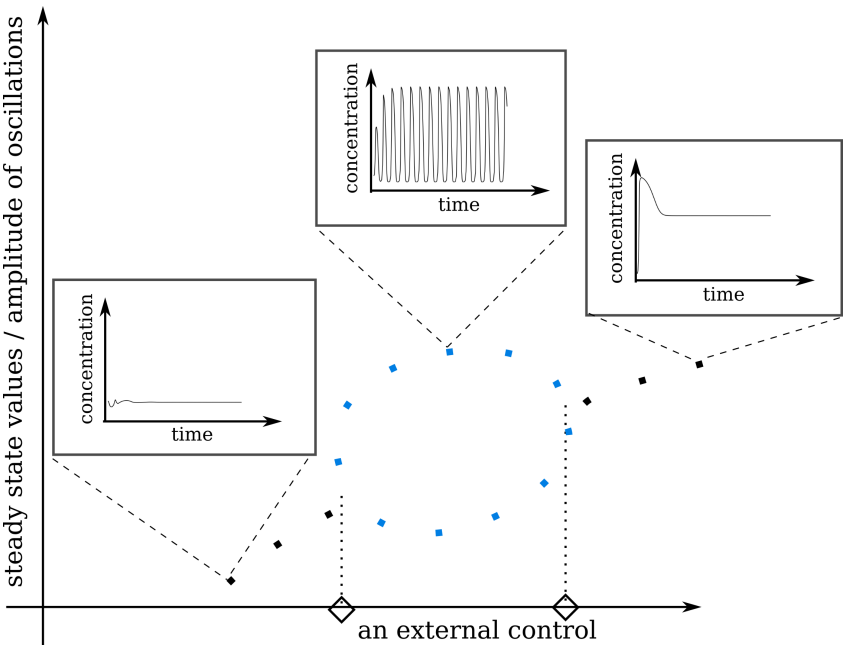


Fig. 4.2.: A schematic diagram showing how a pair of Hopf bifurcation points can be experimentally measured in oscillatory systems. The black dots represent the stable steady state curve and the blue dots represent the peaks and valleys of the stable limit cycles. The black diamonds represent the two measured values of the applied external control at the two bifurcation points. The other control is kept fixed throughout.

Suppose we have an experimental setup for an oscillatory system with two external controls that can be experimentally varied. We begin by keeping both the controls fixed and letting the system reach a stable outcome. This may be either a stable equilibrium or stable oscillations. Let us assume that our initial point is a stable equilibrium represented by the black dot on the bottom-left of Figure 4.2. Now, we can continuously vary one of the controls such that we let the experimental system settle into a stable equilibrium for each value of this control. For small continuous changes in the control value, we expect only small changes in the outcome. We can see this in Figure 4.2, where, as the value of the applied control is increased, the system initially continues to settle into the black equilibrium point. However, as we vary the value of the applied control, at some point, we may observe a sudden change in the qualitative dynamics of the system and notice that the experimental system no longer settles into a stable equilibrium. Instead, the system may now exhibit periodic oscillations. These oscillations are represented by the blue dots in Figure 4.2. Furthermore, if we continue to vary the applied control along the same direction, we will see that the experimental system continues to exhibit periodic oscillations with varying amplitudes. This means that we have experimentally crossed a supercritical Hopf bifurcation point. The values of the applied controls

at which this bifurcation was detected are then recorded as a measurement pair, represented by the black diamond on the left in Figure 4.2. In order to obtain more such measurement pairs, we can simply fix the other external control at a different value and repeat this process.

Suppose we have measured a set of N pairs of external control values $(\tilde{q}_{1_i}, \tilde{q}_{2_i})_{i=1,\dots,N}$ at which Hopf bifurcations were detected. In order to estimate the parameter values $p \in \mathbb{R}^{n_p}$ of the mathematical model using these measurements, we formulate the following parameter estimation problem:

Problem 4.3.3 : Parameter Estimation using Hopf Bifurcation Points
Find the values of the model parameters $p \in \mathbb{R}^{n_p}$ and, for $i = 1, \dots, N$, the steady states $y_i \in \mathbb{R}^{n_y}$, controls $q_i \in \mathbb{R}^2$, and additional variables $v_i, w_i \in \mathbb{R}^{n_y}$ and $\mu_i \in \mathbb{R}$ such that they solve the following optimization problem:

$$\min \quad \sum_{i=1}^{N} \left(\frac{\tilde{q}_{1_i} - q_{1_i}}{\sigma_{1_i}} \right)^2 + \left(\frac{\tilde{q}_{2_i} - q_{2_i}}{\sigma_{2_i}} \right)^2$$

$$\text{s.t.} \quad \begin{cases} \psi(y_i, q_i, p) = 0 \\ \dfrac{\partial \psi(y_i, q_i, p)}{\partial y} v_i + \mu_i w_i = 0 \\ \dfrac{\partial \psi(y_i, q_i, p)}{\partial y} w_i - \mu_i v_i = 0 \\ v_i^T v_i + w_i^T w_i - 1 = 0 \\ v_i^T w_i = 0 \end{cases} \qquad \forall\, i = 1, \dots, N.$$

Here, we have used the Griewank-Reddien formulation [GR83; MM02] to characterize the Hopf bifurcation point. We know from Section 1.3.2 that a Hopf bifurcation point is characterized by the condition that the Jacobian matrix $\dfrac{\partial}{\partial y}\psi(y, q, p)$ of the right-hand side function ψ has a pair of complex conjugate eigenvalues with zero real part. Therefore, the additional variables v_i and w_i correspond to the real and imaginary parts of the eigenvector of the Jacobian matrix at the bifurcation point for each $i = 1, \dots, N$, and the variables μ_i are the imaginary parts of the corresponding eigenvalue.

4.4 Numerical solution strategy

Having formulated the parameter estimation problem as an optimization problem, we can now discuss how to solve it numerically. We will start by translating the notation of the parameter estimation problem into that of the multi-experiment nonlinear least-squares problem presented in Section 3.3. We will then discuss how the generalized Gauss-Newton solver described in Section 3.2 can be used to solve the nonlinear least-squares problem. Finally, we will also present our numerical strategy to generate suitable initial guesses that can improve the convergence of the solver.

4.4.1 Nonlinear least-squares solver

The parameter estimation problems presented in Section 4.3 are equality-constrained, multi-experiment nonlinear least-squares problems (MENLPs). We have already seen problems of this type in Section 3.3. Therefore, in order to use the numerical methods presented in Section 3.3, we need to first translate the parameter estimation problem into the notation of the MENLP 3.3.1. Table 4.1 shows how we can reformulate the optimization variables, objective function and constraints of the parameter estimation problems 4.3.2 and 4.3.3 accordingly.

We also presented in Section 3.2 the generalized Gauss-Newton method as a numerical solver for nonlinear least-squares problems. Furthermore, we described in Section 3.3 an efficient adaptation of the generalized Gauss-Newton method that exploits the block structure of the Jacobian matrix in multi-experiment nonlinear least-squares problems. This is the numerical method we will use in this thesis to solve the parameter estimation problems.

Since the choice of initial guesses for the optimization variables plays a crucial role in the convergence of the generalized Gauss-Newton method, we have developed a robust numerical strategy to generate suitable initial guesses. This is especially important in our parameter estimation problems, where the initial guesses must also satisfy the steady state and bifurcation conditions for every experiment. Therefore, in the following sections, we will describe the general strategy in detail and then present the specific steps for saddle-node and Hopf bifurcations.

	Saddle-Node Bifurcation	Hopf Bifurcation
$x_i^{\text{loc}} =$	$\begin{pmatrix} y_i \\ q_i \\ h_i \end{pmatrix} \in \mathbb{R}^{2n_y+2}$	$\begin{pmatrix} y_i \\ q_i \\ v_i \\ w_i \\ \mu_i \end{pmatrix} \in \mathbb{R}^{3n_y+3}$
$x^{\text{glb}} =$	$p \in \mathbb{R}^{n_p}$	$p \in \mathbb{R}^{n_p}$
$f(x_i^{\text{loc}}, x^{\text{glb}}) =$	$\begin{pmatrix} q_{1_i} - \tilde{q}_{1_i} \\ q_{2_i} - \tilde{q}_{2_i} \end{pmatrix} \in \mathbb{R}^2$	$\begin{pmatrix} q_{1_i} - \tilde{q}_{1_i} \\ q_{2_i} - \tilde{q}_{2_i} \end{pmatrix} \in \mathbb{R}^2$
$g(x_i^{\text{loc}}, x^{\text{glb}}) =$	$\begin{pmatrix} \psi_i(y_i, q_i, p) \\ \dfrac{\partial \psi_i(y_i, q_i, p)}{\partial y} h_i \\ h_i^T h_i - 1 \end{pmatrix} \in \mathbb{R}^{2n_y+1}$	$\begin{pmatrix} \psi_i(y_i, q_i, p) \\ \dfrac{\partial \psi_i(y_i, q_i, p)}{\partial y} v_i + \mu_i w_i \\ \dfrac{\partial \psi_i(y_i, q_i, p)}{\partial y} w_i - \mu_i v_i \\ v_i^T v_i + w_i^T w_i - 1 \\ v_i^T w_i \end{pmatrix} \in \mathbb{R}^{3n_y+2}$

Tab. 4.1.: Reformulation of the optimization variables and functions of the parameter estimation problems for saddle-node and Hopf bifurcations presented in Section 4.3 using the notation of the multi-experiment nonlinear least-squares problem presented in Section 3.3. The optimization variables x_i^{loc}, the objective function $f(x_i^{\text{loc}}, x^{\text{glb}})$ and the equality constraints $g(x_i^{\text{loc}}, x^{\text{glb}})$ are defined for each measurement $i = 1, \ldots, N$. The optimization variables x^{glb} are common to all the experiments.

4.4.2 Initial guess generation

The optimization variables in our parameter estimation problem 4.3.1 are the steady states $y_i \in \mathbb{R}^{n_y}$, the applied controls $q_i \in \mathbb{R}^2$ and the additional variables $z_i \in \mathbb{R}^{n_z}$ corresponding to each $i = 1, \ldots, N$ and the model parameters $p \in \mathbb{R}^{n_p}$, which are common to all the experiments.

For the model parameters $p \in \mathbb{R}^{n_p}$, a reasonable initial guess can be obtained from expert knowledge based on literature or empirical tests. Let us call this initial guess for the parameters as $\bar{p} \in \mathbb{R}^{n_p}$.

In order to then find suitable initial guesses for the remaining optimization variables, we propose a robust numerical strategy that can produce a curve of bifurcation points lying close to the measurement data. The general idea of this strategy is outlined in the following steps:

Step 1: Solve the steady state condition to find an initial steady state solution.

Step 2: Vary one control in a given range to compute a curve of steady states.

Step 3: Detect bifurcation points on the curve and select one for continuation.

Step 4: Continue the selected bifurcation point varying both controls to obtain a curve of bifurcation points that trace the measured control values.

Step 5: Find, for each pair of measured controls, a corresponding bifurcation point to use as initial guess.

This procedure was adapted from the approach used by Schlöder and Bock [Sch87; SB83] in their case study of the Belousov-Zhabotinsky reaction. In particular, we have not only abstracted their approach, but also generalized it to work for both saddle-node and Hopf bifurcations. Moreover, we also use two different types of numerical continuation methods for computing the bifurcation diagrams: the pseudo-arclength method (see Section 2.2) and the deflated continuation method (see Section 2.3). While the pseudo-arclength method is the traditional approach to numerical continuation, the deflated continuation method offers a more robust approach for computing the steady-state curve. This is because the deflated continuation method enables us to find solution branches that are not continuously connected to the initial point. Additionally, we have also developed a reliable heuristic to adapt the step size for both numerical continuation methods, so that one of the two controls in the curve of bifurcation points passes through the measured values of the control.

We will now elaborate on each step in the initial guess generation procedure for the two types of bifurcations we are interested in: saddle-node and Hopf bifurcations.

Saddle-Node Bifurcations

We need an initial guess for the steady states $y_i \in \mathbb{R}^{n_y}$, applied controls $(q_{1_i}, q_{2_i}) \in \mathbb{R}^2$ and eigenvectors $h_i \in \mathbb{R}^{n_y}$ for each $i = 1, \ldots, N$.

For the applied controls, we start by selecting a pair of measured controls $(\underline{q_1}, \underline{q_2}) \equiv (\tilde{q}_{1_i}, \tilde{q}_{2_i})$ from some $i \in \{1, \ldots, N\}$. With the model parameters initialized to $\bar{p}$, we then proceed through the following steps (also illustrated in Figure 4.3) to determine the initial guesses for all the remaining optimization variables:

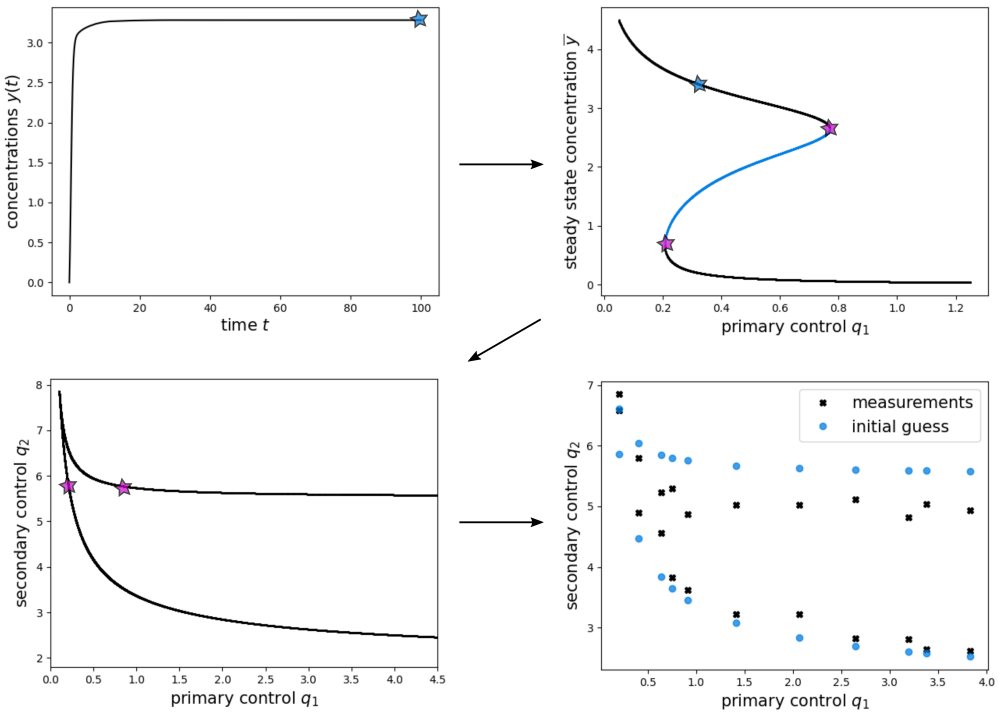


Fig. 4.3.: A schematic diagram showing the steps involved in generating initial guesses for the parameter estimation problem using measurements of saddle-node bifurcation points. First, a steady state solution of the dynamical system is identified by simulation and optimization as visualized in the top-left plot. The blue star marks the steady state solution used as a starting point for numerical continuation. Then, a curve of steady states is drawn using the deflated continuation method to obtain a one-parameter bifurcation diagram. This is visualized in the top-right plot where the black points represent stable steady states and the blue points represent unstable steady states. Saddle-node bifurcation points identified on this curve are indicated with pink stars. Either one of the detected bifurcation points could be numerically continued to draw a curve of bifurcation points as visualized in the bottom-left plot. This gives us the initial guesses needed for solving the parameter estimation problem. The bottom-right plot shows the initial guesses for the applied controls, marked by blue circles, and their corresponding measurement values, marked by black crosses.

Step 1: Solve the steady state condition to find an initial steady state

In order to compute a curve of steady state solutions, we need to first find an initial steady state. We can do this by using Newton's method to solve the following steady state condition for $y \in \mathbb{R}^{n_y}$:

$$\psi(y, \underline{q_1}, \underline{q_2}, \bar{p}) = 0.$$

As an initial guess for the Newton solver, we use the state $y(t_{\text{end}}) \in \mathbb{R}^{n_y}$ obtained by solving the following initial value problem from a random initial state $\underline{y} \in \mathbb{R}^{n_y}$ until time $t_{\text{end}} \in \mathbb{R}_+$:

$$\begin{aligned} \dot{y}(t) &= \psi\left(y(t), \underline{q_1}, \underline{q_2}, \bar{p}\right), \\ y(0) &= \underline{y}. \end{aligned}$$

This step is visualized in the top-left plot of Figure 4.3 where the solution of the initial value problem is plotted, and the steady state $\underline{\underline{y}} \in \mathbb{R}^{n_y}$ is marked with a blue star.

Step 2: Vary one control in a given range to compute a curve of steady states

Once we have a steady state solution $\underline{\underline{y}} \in \mathbb{R}^{n_y}$, we can use numerical continuation to compute a curve of steady states by varying the values of a control parameter. Let us call this control the *primary control* and without loss of generality, we can assume that it is $q_1 \in \mathbb{R}$. The other control, which we will call the *secondary control*, will then be kept fixed at the value $\underline{q_2} \in \mathbb{R}$ in this step.

To compute the steady-state curve starting from the known solution $\underline{\underline{y}}$, we use the deflated continuation method presented in Section 2.3. This method offers the advantage that solution branches that are not continuously connected to $\underline{\underline{y}}$ may also be found.

This step is visualized in the top-right plot of Figure 4.3 where the black points represent stable steady states and the blue points represent unstable steady states. The blue star on this curve marks the initial steady state $\underline{\underline{y}}$.

Step 3: Detect bifurcation points on the curve and select one for continuation

From the steady-state curve computed in Step 2, we can now identify all the saddle-node bifurcation points as described in Section 2.3.3. If we find multiple bifurcation points on this curve, we select one of them for continuation.

Suppose we select the bifurcation point with the steady state $\underline{\underline{\underline{y}}} \in \mathbb{R}^{n_y}$ and primary control $\underline{\underline{\underline{q_1}}} \in \mathbb{R}$ from the steady-state curve. This point serves as a crude approximation of a saddle-node bifurcation point. Now, in order to get a better approximation, we solve the following constrained nonlinear optimization problem:

$$\begin{aligned}\left(\underline{\underline{\underline{y}}}, \underline{\underline{\underline{q_1}}}, \underline{\underline{\underline{h}}}\right) = \quad & \min_{y,q_1,h} \quad \frac{\partial \psi(y, q_1, \underline{q_2}, \bar{p})}{\partial y} h \\ & \text{s.t.} \quad \psi(y, q_1, \underline{q_2}, \bar{p}) = 0, \\ & \qquad h^T h - 1 = 0.\end{aligned}$$

Here, $\underline{\underline{\underline{y}}}$ is taken as the initial guess for the steady state y and $\underline{\underline{\underline{q_1}}}$ as the initial guess for the primary control q_1. Furthermore, the eigenvector of the Jacobian matrix at this bifurcation point is taken as the initial guess for the additional variable $h \in \mathbb{R}^{n_y}$.

The bifurcation points detected in this step are visualized as pink stars in the top-right and bottom-left plots of Figure 4.3.

Step 4: Continue the curve of bifurcation points along measured controls

We have identified a saddle-node bifurcation point with steady state values $\underline{\underline{\underline{y}}} \in \mathbb{R}^{n_y}$, primary control $\underline{\underline{\underline{q_1}}} \in \mathbb{R}$, secondary control $\underline{q_2} \in \mathbb{R}$, and additional variables $\underline{\underline{\underline{h}}} \in \mathbb{R}^{n_y}$ in Step 3. We can now use this point as the initial known solution in a numerical continuation method to compute a curve of saddle-node bifurcation points. This means that we solve the following system of nonlinear equations for varying values of the primary control $q_1 \in \mathbb{R}$ to find steady states $y \in \mathbb{R}^{n_y}$, controls $(q_1, q_2) \in \mathbb{R}^2$ and additional variables $h \in \mathbb{R}^{n_y}$:

$$\begin{aligned}\frac{\partial \psi(y, q_1, q_2, \bar{p})}{\partial y} h &= 0 \\ \psi(y, q_1, q_2, p_0) &= 0 \\ h^T h - 1 &= 0.\end{aligned}$$

Note that the secondary control q_2, which has been kept fixed to the value $\underline{q_2}$ so far, is now free to be estimated.

For this step, we may use either the deflated continuation method (see Section 2.3) or the pseudo-arclength continuation method (see Section 2.2) to compute the curve of bifurcation points. Moreover, since we would like to find bifurcation points that lie close to the measurement data, we use the heuristic described in sections 2.2 and 2.3 to adapt the step size and make this possible.

This step is visualized in the bottom-left plot of Figure 4.3 where the black points on the curve are saddle-node bifurcation points. The pink stars in this plot correspond to the bifurcation points detected in Step 3.

Step 5: Find, for each measurement data, a bifurcation point to use as initial guess

Once we have computed a curve of bifurcation points that passes through (or lies close to) the measured values of a control, we can select, for each pair of measured controls $(\tilde{q}_{1_i}, \tilde{q}_{2_i})$ for $i = 1, \ldots, N$, a corresponding matching solution from the bifurcation curve. Therefore, we obtain a subset $\left(\bar{x}_i^{\text{loc}}\right)_{i=1,\ldots,N}$ of solutions from the curve of bifurcation points such that:

$$\bar{x}_i^{\text{loc}} = \left(\bar{y}_i, \bar{q}_{1_i}, \bar{q}_{2_i}, \bar{h}_i\right) \in \mathbb{R}^{2n_y+2} \quad \text{for} \quad i = 1, \ldots, N.$$

We select this subset such that $\bar{q}_{1_i} \approx \tilde{q}_{1_j}$ for some $j \in \{1, \ldots, N\}$. If there are measurements that do not match any solutions on the curve of bifurcation points, we use a greedy approach to select a bifurcation point on this curve that lies closest to the measurement.

Finally, we can define $\bar{x}^{\text{glb}} = \bar{p}$ to get, together with the solutions $\bar{x}_i^{\text{loc}}$ for $i = 1, \ldots, N$, the initial guesses for all the optimization variables of the multi-experiment parameter estimation problem 4.3.2.

This step is visualized in the bottom-right plot of Figure 4.3 where the blue circles are the saddle-node bifurcation points that form the initial guesses for the multi-experiment parameter estimation problem and the black crosses are the measurement data.

Hopf Bifurcations

We need an initial guess for the steady states $y_i \in \mathbb{R}^{n_y}$, applied controls $(q_{1_i}, q_{2_i}) \in \mathbb{R}^2$ and additional variables $(v_i, w_i, \mu_i) \in \mathbb{R}^{2n_y+1}$ for each $i = 1, \ldots, N$.

For the applied controls, we start by selecting a pair of measured controls $(\underline{q_1}, \underline{q_2}) \equiv (\tilde{q}_{1_i}, \tilde{q}_{2_i})$ from some $i \in \{1, \ldots, N\}$. With the model parameters initialized to $\bar{p}$, we then proceed through the following steps (also illustrated in Figure 4.4) to obtain the initial guesses for all the remaining optimization variables :

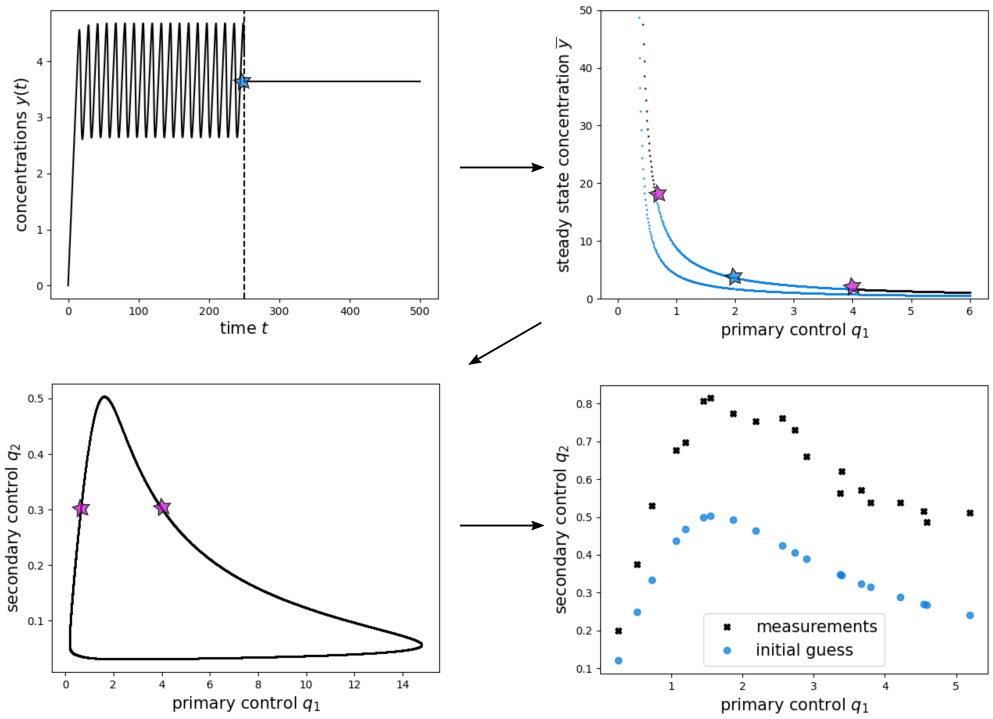


Fig. 4.4.: A schematic diagram showing the steps involved in generating initial guesses for the parameter estimation problem using measurements of Hopf bifurcation points. In the first step visualized in the top-left plot, we solve the initial value problem until some time t_{end} marked by the vertical dotted line. Since this produced stable oscillations in this case, we use a point on the periodic orbit as the initial guess for solving the steady state condition. The unstable steady state identified in this process is indicated with the blue star and the equilibrium visualized by simulating the system from the dotted line starting at the unstable steady state. Using the deflated continuation method, we can then draw a curve of steady states starting from the unstable steady state to get a one-parameter bifurcation diagram. This is visualized in the top-right plot where the black plots indicate stable steady states and the blue points indicate unstable steady states. The periodic orbits are not visualized in this plot. Hopf bifurcation points identified on this curve are indicated with pink stars. Either one of the detected bifurcation points could be numerically continued to draw a curve of bifurcation points as visualized in the bottom-left plot. This gives us the initial guesses needed for solving the parameter estimation problem. The bottom-right plot shows the initial guesses for the applied controls, marked by blue circles, and their corresponding measurement values, marked by black crosses.

Step 1: Solve the steady state condition to find an initial steady state

In order to compute a curve of steady states, we need to first find an initial steady state solution. We can do this by using Newton's method to solve the following steady state condition for $y \in \mathbb{R}^{n_y}$:

$$\psi(y, \underline{q_1}, \underline{q_2}, \bar{p}) = 0.$$

As an initial guess for the Newton solver, we use the state $y(t_{\text{end}}) \in \mathbb{R}^{n_y}$ obtained by solving the following initial value problem from a random initial state $\underline{y} \in \mathbb{R}^{n_y}$ until time $t_{\text{end}} \in \mathbb{R}_+$:

$$\begin{aligned} \dot{y}(t) &= \psi\left(y(t), \underline{q_1}, \underline{q_2}, \bar{p}\right) \\ y(0) &= \underline{y}. \end{aligned}$$

In this case, depending upon the values of the parameters and controls, the solution $y(t_{\text{end}})$ may approach a stable periodic orbit. We have illustrated this case in the top-left plot of Figure 4.4 where the dotted line marks t_{end}. At this point, a point on the periodic orbit is taken as the initial guess for the Newton solver to obtain the unstable steady state at the center of the orbit. We have marked this unstable steady state $\underline{\underline{y}} \in \mathbb{R}^{n_y}$ with a blue star in the top-left plot of Figure 4.4. We also show that this is indeed a steady state by solving the initial value problem with the unstable steady state as the initial state.

Step 2: Vary one control in a given range to compute a curve of steady states

Once we have the steady state solution $\underline{\underline{y}} \in \mathbb{R}^{n_y}$, we can use a numerical continuation method to compute a curve of steady states by varying the values of a control parameter. Let us, once again, call this control the *primary control* and without loss of generality, we can assume that it is $q_1 \in \mathbb{R}$. The other control, which will be kept fixed to the value $\underline{q_1} \in \mathbb{R}$ in this step, will then be called the *secondary control*.

To compute the steady-state curve starting from the known solution $\underline{\underline{y}}$, we use the deflated continuation method presented in Section 2.3. This method allows us to potentially find branches of steady states that may be disconnected from the initial point $\underline{\underline{y}}$.

This step is visualized in the top-right plot of Figure 4.4 where the black points are the stable steady states and the blue points are the unstable steady states. The blue star on this plot marks the initial steady state $\underline{\underline{y}}$.

Step 3: Detect bifurcation points on the curve and select one for continuation

From the steady-state curve computed in Step 2, we can now identify all the Hopf bifurcation points as described in Section 2.3.3. If we detect multiple bifurcation points on this curve, we select one of them for continuation.

Suppose we select the bifurcation point with the steady state $\underline{\underline{y}} \in \mathbb{R}^{n_y}$, and primary control $\underline{\underline{q_1}} \in \mathbb{R}$ from the steady-state curve. This point serves as a crude approximation of a Hopf bifurcation point. Now, in order to obtain a better approximation of the bifurcation point, we solve the following constrained nonlinear optimization problem:

$$\begin{aligned}\left(\underline{\underline{y}}, \underline{\underline{q_1}}, \underline{\underline{v}}, \underline{\underline{w}}, \underline{\underline{\mu}}\right) = \quad \min_{y,q_1,\mu,v,w} \quad & \begin{pmatrix} \dfrac{\partial \psi(y, q_1, \underline{\underline{q_2}}, \bar{p})}{\partial y} v + \mu w \\ \dfrac{\partial \psi(y, q_1, \underline{\underline{q_2}}, \bar{p})}{\partial y} w - \mu v \\ v^T w \end{pmatrix} \\ \text{s.t.} \quad & \psi(y, q_1, \underline{\underline{q_2}}, \bar{p}) = 0, \\ & v^T v + w^T w - 1 = 0.\end{aligned}$$

Here, we take $\underline{\underline{y}}$ as the initial guess for the steady state and $\underline{\underline{q_1}}$ as the initial guess for the primary control q_1. Furthermore, we obtain initial guesses for the additional variables $v \in \mathbb{R}^{n_y}$ and $w \in \mathbb{R}^{n_y}$ from the real and imaginary parts of the eigenvector of the Jacobian matrix at the approximate bifurcation point, with the imaginary part of the corresponding eigenvalue taken as the initial guess for $\mu \in \mathbb{R}$.

The bifurcation points detected in this step are marked as pink stars in the top-right and bottom-left plots of Figure 4.4.

Step 4: Continue the curve of bifurcation points along measured controls

We have identified one Hopf bifurcation point with steady state values $\underline{\underline{y}} \in \mathbb{R}^{n_y}$, controls $\left(\underline{\underline{q_1}}, \underline{\underline{q_2}}\right) \in \mathbb{R}^2$, and additional variables $\left(\underline{\underline{v}}, \underline{\underline{w}}, \underline{\underline{\mu}}\right) \in \mathbb{R}^{2n_y+1}$ in Step 3. We can now use this point as the initial known solution in a numerical continuation method to compute a curve of Hopf bifurcation points. This means that we need to solve the following system of nonlinear equations for varying values of the primary

control $q_1 \in \mathbb{R}$ to find steady states $y \in \mathbb{R}^{n_y}$, controls $(q_1, q_2) \in \mathbb{R}^2$ and additional variables $(v, w, \mu) \in \mathbb{R}^{2n_y+1}$:

$$\begin{aligned} \frac{\partial \psi(y, q, \bar{p})}{\partial y} v + \mu w &= 0, \\ \frac{\partial \psi(y, q, \bar{p})}{\partial y} w - \mu v &= 0, \\ v^T w &= 0, \\ \psi(y, q, \bar{p}) &= 0, \\ v^T v + w^T w - 1 &= 0. \end{aligned}$$

Note that the secondary control q_2, which was kept fixed to the value $\underline{q_2}$ so far, is now free to be estimated.

For this step, we may use either the deflated continuation method (see Section 2.3) or the pseudo-arclength continuation method (see Section 2.2) to compute the curve of bifurcation points. Moreover, since we would like to identify bifurcation points that lie close to the measurement data, we use the heuristic described in sections 2.2 and 2.3 to adapt the step size for this purpose.

This step is visualized in the bottom-left plot of Figure 4.4 where the black points on the curve are Hopf bifurcation points. The pink stars in this plot correspond to the bifurcation points detected in Step 3.

Step 5: Find, for each measurement data, a bifurcation point to use as initial guess

Once we have computed a curve of bifurcation points that passes through (or lies close to) the measured values of a control, we can select, for each pair of measured controls $(\tilde{q}_{1_i}, \tilde{q}_{2_i})$ for $i = 1, \ldots, N$, a corresponding matching solution from the bifurcation curve. In this way, we obtain a subset $\left(\bar{x}_i^{\text{loc}}\right)_{i=1,\ldots,N}$ of solutions from the curve of bifurcation points such that:

$$\bar{x}_i^{\text{loc}} = (\bar{y}_i, \bar{q}_{1_i}, \bar{q}_{2_i}, \bar{\mu}_i, \bar{v}_i, \bar{w}_i) \in \mathbb{R}^{3n_y+2} \quad \text{for} \quad i = 1, \ldots, N.$$

We select this subset such that $\bar{q}_{1_i} \approx \tilde{q}_{1_j}$ for some $j \in \{1, \ldots, N\}$. If there are measurements that do not exactly match any solutions on the curve of bifurcation points, we use a greedy approach to select a bifurcation point on this curve that lies closest to the measurement.

Finally, we can define $\bar{x}^{\text{glb}} = \bar{p}$ to get, together with the solutions $\bar{x}_i^{\text{loc}}$ for $i = 1, \ldots, N$, the initial guesses for all the optimization variables of the multi-experiment parameter estimation problem 4.3.3.

This step is visualized in the bottom-right plot of Figure 4.4 where the blue circles are the Hopf bifurcation points that form the initial guesses for the multi-experiment parameter estimation problem and the black crosses are the measurement data.

4.4.3 Challenges and limitations

In this chapter, we have formulated parameter estimation problems using measurements of saddle-node and Hopf bifurcation points, and presented numerical solution strategies to solve them. In particular, we have discussed in detail how to generate suitable initial guesses for the optimization variables to improve the convergence of the generalized Gauss-Newton solver. Let us now consider some challenges and limitations inherent in our approach.

One of the main challenges in using measurements of bifurcation points for parameter estimation lies in the design of mathematical models. Ensuring that the model structure allows for the desired bifurcation behavior can be a non-trivial task, especially for complex systems with many interacting components. Moreover, even with a suitable model structure, it can be an additional challenge to locate the region in parameter space where the mathematical model can exhibit the desired bifurcations. This may require extensive parameter exploration with multi-start optimization strategies to identify parameter sets that produce the expected bifurcation behavior.

Nevertheless, despite these challenges, our approach provides a powerful new tool for parameter estimation that can be used as an alternative to (or in combination with) the traditional approach using time-series data.

5 A-posteriori Sensitivity Analysis

Measurements of real-world processes are almost always subject to some degree of error. When these noisy measurements are used to estimate the parameters of a mathematical model, an important question arises: how reliable are the parameter estimates? In other words, how sensitive are the estimated parameters to uncertainties in the measurements? This chapter aims to find an answer to this question with the help of *a posteriori* sensitivity analysis.

We begin our discussion by defining the concept of sensitivity or identifiability. Next, we examine the underlying statistics of the least-squares formulation of our parameter estimation problem. This, in turn, provides the basis for finding a numerical approximation of the covariance matrix at the estimated parameters and thereby, defining confidence regions for the parameter estimates. Finally, we discuss how the multi-experiment structure of the parameter estimation problem can be exploited to compute the covariance matrix more efficiently. The material presented in this chapter follows the presentation in [Som17] based on the work by Bock et al. [BKK07] and Schlöder [Sch87].

5.1 Introduction

The concept of identifiability is fundamental to parameter estimation because it determines the quality and reliability of the estimated parameters. In particular, we distinguish between two different types of identifiability: structural identifiability and practical identifiability. A detailed discussion of these concepts can be found in [CD80; Ash+09; Chi+16]. Here, we only briefly summarize the key points.

A parameter is said to be *structurally identifiable* if it can be uniquely determined from the theoretically available (complete or partial) measurement data, assuming the measurements are error-free. There are various techniques for detecting structural non-identifiability in dynamical systems including a Taylor series approach [Poh78], similarity transformation approach [VGR89] and differential algebra ap-

proach [Aud+01]. However, in some simple cases, structural non-identifiability can even be detected by direct inspection of the mathematical model. For example, consider the following equations:

$$\dot{y}(t) = (p_1 + p_2)y(t) \quad \text{or} \quad \dot{y}(t) = p_1 p_2 y(t).$$

In both cases, it is clearly impossible to uniquely determine both the parameters $p_1 \in \mathbb{R}$ and $p_2 \in \mathbb{R}$, even with an arbitrary number of measurements of the state $y(t) \in \mathbb{R}$ for time $t \in \mathbb{R}$.

This type of identifiability analysis is typically done as part of *a priori* sensitivity analysis because it depends solely on the model structure rather than the actual measurements.

On the other hand, *practical identifiability* assesses whether the amount and quality of the actual measurement data are sufficient to reliably estimate the parameters, assuming the parameters are, in theory, structurally identifiable. This requires quantifying how measurement errors propagate into uncertainty in the estimated parameters. Since this analysis depends on both the model and the actual data, it is typically done as part of *a posteriori* sensitivity analysis. A widely used numerical approach for practical identifiability analysis involves computing the sensitivity matrix of the observables with respect to the estimated parameters. This sensitivity matrix allows us to then compute the covariance matrix and quantify the confidence region around the estimated parameters.

In this chapter, we focus on the practical identifiability analysis for the parameter estimation problems described in Chapter 4, and demonstrate how the covariance matrix and confidence regions can be computed for the estimated parameters. The next chapter will explore how the quality of the estimated parameters can then be improved by optimizing the experimental design.

5.2 Statistical motivation

The parameter estimation problems that we consider in this thesis are nonlinear least-squares problems that depend on measurements of pairs of applied external controls at bifurcation points (see Section 4.3 for more details). In this section, we will try to understand the statistical motivation behind their weighted least-squares formulation.

Suppose we run M experiments to obtain a set of $N \leq M$ noisy measurements of the applied external controls at bifurcation points. We denote these measurements by $(\tilde{q}_{1_i}, \tilde{q}_{2_i}) \in \mathbb{R}^2$ for $i = 1, \ldots, N$. Let the corresponding "true" values of these measurements be given by $(\bar{q}_{1_i}, \bar{q}_{2_i}) \in \mathbb{R}^2$ for $i = 1, \ldots, N$.

Assume that each experiment is done independently, so that the measurement errors are independent. Assume also that each measurement $(\tilde{q}_{1_i}, \tilde{q}_{2_i})$, for $i = 1, \ldots, N$ is subject to additive, normally distributed error with zero mean and variances given by $\sigma_{1_i}^2$ and $\sigma_{2_i}^2$ respectively[1]. Then, the measurements can be expressed as:

$$\begin{aligned} \tilde{q}_{1_i} &= \bar{q}_{1_i} + \epsilon_{1_i}, \qquad \epsilon_{1_i} \sim \mathcal{N}(0, \sigma_{1_i}^2) \\ \tilde{q}_{2_i} &= \bar{q}_{2_i} + \epsilon_{2_i}, \qquad \epsilon_{2_i} \sim \mathcal{N}(0, \sigma_{2_i}^2) \end{aligned} \qquad \forall i = 1, \ldots, N.$$

The goal of our parameter estimation problem is to use the measured controls to find the values of the unknown parameters of the mathematical model such that the theoretically predicted bifurcation points lie close to the measured bifurcation points.

In order to theoretically predict a bifurcation point, we need to know the values of the steady states, applied controls, and the eigenvectors and eigenvalues of the Jacobian matrix at the bifurcation point (cf. Section 4.3). This means that the optimization variables of the parameter estimation problem are given by:

$$\theta = \begin{bmatrix} y_1 & q_1 & z_1 & \cdots & y_N & q_N & z_N & p \end{bmatrix}^T \in \mathbb{R}^n$$

where $y_i \in \mathbb{R}^{n_y}$ are the steady state, $q_i \in \mathbb{R}^2$ the applied controls, and $z_i \in \mathbb{R}^{n_z}$ the additional variables that characterize the bifurcation point for the i-th experiment. The parameters $p \in \mathbb{R}^{n_p}$ are common to all the experiments.

We can try to find the best solution $\theta^* \in \mathbb{R}^n$ of the parameter estimation problem by maximizing the conditional probability of observing the measurements given the model. This is known as the *maximum likelihood approach*.

[1] A detailed discussion on why these assumptions are reasonable can be found in the dissertation by Sommer [Som17].

Since we assumed that our measurement errors are independent and additive, normally distributed, we can define the likelihood function as follows:

$$\theta^* = \underset{\theta \in \mathbb{R}^n}{\arg\max} \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}\sigma_{1_i}} \exp\left(-\frac{1}{2}\left(\frac{\tilde{q}_{1_i} - q_{1_i}}{\sigma_{1_i}}\right)^2\right) \cdot \frac{1}{\sqrt{2\pi}\sigma_{2_i}} \exp\left(-\frac{1}{2}\left(\frac{\tilde{q}_{2_i} - q_{2_i}}{\sigma_{2_i}}\right)^2\right)$$
$$\text{s.t.} \quad \left.\begin{aligned} \psi(y_i, q_i, p) &= 0 \\ \zeta(y_i, q_i, z_i, p) &= 0 \end{aligned}\right\} i = 1, \ldots, N$$

where $\psi \in \mathcal{C}^2$ is the right-hand side function of the dynamical system and $\zeta \in \mathcal{C}^2$ characterizes the bifurcation point (cf. Section 4.3).

Log-transforming the probabilities, which still preserves the location of the optimum, and flipping the sign of the objective function to turn the maximization problem into a minimization problem, we get the following optimization problem for the negative log-likelihood function:

$$\theta^* = \underset{\theta \in \mathbb{R}^n}{\arg\min} \frac{1}{2} \sum_{i=1}^{N} \log\left(2\pi\sigma_{1_i}^2\right) + \left(\frac{\tilde{q}_{1_i} - q_{1_i}}{\sigma_{1_i}}\right)^2 + \log\left(2\pi\sigma_{2_i}^2\right) + \left(\frac{\tilde{q}_{2_i} - q_{2_i}}{\sigma_{2_i}}\right)^2$$
$$\text{s.t.} \quad \left.\begin{aligned} \psi(y_i, q_i, p) &= 0 \\ \zeta(y_i, q_i, z_i, p) &= 0 \end{aligned}\right\} i = 1, \ldots, N.$$

When the variances $\sigma_{1_i}^2$ and $\sigma_{2_i}^2$ of the measurement errors are known for all $i = 1, \ldots, N$, then the corresponding terms $\log\left(2\pi\sigma_{1_i}^2\right)$ and $\log\left(2\pi\sigma_{2_i}^2\right)$ in the negative log-likelihood function become constants that can be ignored in the optimization. This results in the following optimization problem:

$$\theta^* = \underset{\theta \in \mathbb{R}^n}{\arg\min} \frac{1}{2} \sum_{i=1}^{N} \left(\frac{\tilde{q}_{1_i} - q_{1_i}}{\sigma_{1_i}}\right)^2 + \left(\frac{\tilde{q}_{2_i} - q_{2_i}}{\sigma_{2_i}}\right)^2$$
$$\text{s.t.} \quad \left.\begin{aligned} \psi(y_i, q_i, p) &= 0 \\ \zeta(y_i, q_i, z_i, p) &= 0 \end{aligned}\right\} i = 1, \ldots, N.$$

If this weighted least-squares problem looks familiar, that is because it is exactly our parameter estimation problem from Section 4.3. Therefore, this means that the solution θ^* of the parameter estimation problem is also a maximum likelihood estimate.

5.3 Computation of the covariance matrix

When solving an inverse problem such as in parameter estimation, errors and uncertainties in the measurements inevitably propagate into the estimated parameters. This uncertainty in the estimated parameters can be quantified by computing the covariance matrix of the estimated parameters.

Before we derive the covariance matrix, consider a modified version of the parameter estimation problem from Section 4.3 that explicitly takes into account the measurement noise.

Problem 5.3.1 : Perturbed parameter estimation problem
Find the values of the model parameters $p \in \mathbb{R}^{n_p}$ and for $i = 1, \dots, N$, also the steady states $y_i \in \mathbb{R}^{n_y}$, controls $q_i \in \mathbb{R}^2$ and additional variables $z_i \in \mathbb{R}^{n_z}$ such that they solve the following optimization problem:

$$\theta^* = \underset{\theta \in \mathbb{R}^n}{\arg\min} \quad \frac{1}{2} \left\| f\left(\theta; \epsilon\right) \right\|^2 := \frac{1}{2} \sum_{i=1}^{N} \left(\frac{\bar{q}_{1_i} + \epsilon_{1_i} - q_{1_i}}{\sigma_{1_i}} \right)^2 + \left(\frac{\bar{q}_{2_i} + \epsilon_{2_i} - q_{2_i}}{\sigma_{2_i}} \right)^2$$

$$\text{s.t.} \qquad g\left(\theta\right) := \begin{cases} \psi(y_i, q_i, p) = 0 \\ \zeta(y_i, q_i, z_i, p) = 0 \end{cases} \qquad i = 1, \dots, N,$$

where $f : \mathbb{R}^n \to \mathbb{R}^{m^{\text{obj}}}$ and $g : \mathbb{R}^n \to \mathbb{R}^{m^{\text{cons}}}$ are twice continuously differentiable functions, ϵ denotes the normally-distributed measurement errors:

$$\epsilon := \begin{bmatrix} \epsilon_{1_1} & \epsilon_{2_1} & \cdots & \epsilon_{1_N} & \epsilon_{2_N} \end{bmatrix}^T \text{ with } \begin{matrix} \epsilon_{1_i} \sim \mathcal{N}(0, \sigma_{1_i}^2), \\ \epsilon_{2_i} \sim \mathcal{N}(0, \sigma_{2_i}^2), \end{matrix} \quad \forall i = 1, \dots, N,$$

and $\theta \in \mathbb{R}^n$ is the vector of optimization variables defined as:

$$\theta = \begin{bmatrix} y_1 & q_1 & z_1 & \cdots & y_N & q_N & z_N & p \end{bmatrix}^T.$$

We will now use this perturbed parameter estimation problem to compute the covariance matrix of the estimated parameters. We will start by first formulating the general form of the covariance matrix without exploiting its multi-experiment structure. Then, we will discuss how the block structure of the Jacobian matrices in the perturbed parameter estimation problem can be exploited to compute the covariance matrix more efficiently.

5.3.1 General form of the covariance matrix

The perturbed parameter estimation problem is a multi-experiment nonlinear least-squares problem that we treat similar to the parameter estimation problems presented in Section 4.3. This means that we can use the generalized Gauss-Newton method, which solves in each iteration a linearized version of the nonlinear least-squares problem, to compute the solutions. Under certain regularity assumptions, this allows us to compute a first-order approximation of the solution of the perturbed parameter estimation problem as follows:

Theorem 5.3.2 : First-order approximation of the solution
Let $F(\bar{\theta};0) \in \mathbb{R}^{m^{\text{obj}} \times n}$ and $G(\bar{\theta}) \in \mathbb{R}^{m^{\text{cons}} \times n}$ be the Jacobian matrices of the objective function and equality constraints of the perturbed parameter estimation problem evaluated at the "true" values of the optimization variables and satisfying the following two conditions:

$$\text{[CQ]} \quad \text{rank } G(\bar{\theta}) = m^{\text{cons}},$$
$$\text{[PD]} \quad \text{rank} \begin{bmatrix} F(\bar{\theta};0) \\ G(\bar{\theta}) \end{bmatrix} = n.$$

Then, there exist a neighborhood V of $\epsilon = 0$, a neighborhood U of $\bar{\theta}$ and a unique mapping $\theta(\epsilon) : V \longrightarrow U$ such that $\theta(\epsilon)$ minimizes the perturbed parameter estimation problem, and we have:

$$\theta(\epsilon) = \theta(0) - J(\bar{\theta};0)^{+} \begin{bmatrix} \Sigma^{-1}\epsilon \\ 0 \end{bmatrix} + \mathcal{O}(\|\epsilon\|^2).$$

Here, $\Sigma^2 := \mathbb{E}(\epsilon\epsilon^T)$ denotes the covariance matrix of the measurement errors, $\theta(0) = \bar{\theta}$ denotes the "true" values of the optimization variables and

$$J(\bar{\theta};0) = \begin{bmatrix} F(\bar{\theta};0)^T & G(\bar{\theta})^T \end{bmatrix}^T \in \mathbb{R}^{(m^{\text{obj}}+m^{\text{cons}})\times n},$$
$$J(\bar{\theta};0)^{+} = \begin{pmatrix} \mathbb{I} & 0 \end{pmatrix} \begin{pmatrix} F(\theta^*)^T F(\theta^*) & G(\theta^*)^T \\ G(\theta^*) & 0 \end{pmatrix}^{-1} \begin{pmatrix} F(\theta^*)^T & 0 \\ 0 & \mathbb{I} \end{pmatrix} \in \mathbb{R}^{n\times(m^{\text{obj}}+m^{\text{cons}})}.$$

Proof. See Bock et al. [BKK07]. □

We can now use this first-order approximation of the solution derived in Theorem 5.3.2 to quantify the sensitivity of the solution $\theta(\epsilon)$ with respect to the measurement errors ϵ.

Consider the estimated solution $\theta(\epsilon) \in \mathbb{R}^n$ as a perturbation from the "true" values $\bar{\theta} \in \mathbb{R}^n$. This perturbation $\delta\theta(\epsilon) \in \mathbb{R}^n$ is then given by:

$$\delta\theta(\epsilon) := \theta(\epsilon) - \theta(0) = -J\left(\bar{\theta}; 0\right)^+ \begin{bmatrix} \Sigma^{-1}\epsilon \\ 0 \end{bmatrix}.$$

Since the measurement errors are random variables, we can then compute the expected values and covariance matrix of the perturbation $\delta\theta(\epsilon)$ as follows:

$$\begin{aligned}
\mathbb{E}\left(\delta\theta\left(\epsilon\right)\right) &= 0, \\
\mathbb{E}\left(\delta\theta\left(\epsilon\right)\delta\theta\left(\epsilon\right)^T\right) &= \mathbb{E}\left(J\left(\bar{\theta}; 0\right)^+ \begin{bmatrix} \Sigma^{-1}\epsilon \\ 0 \end{bmatrix} \begin{bmatrix} \Sigma^{-1}\epsilon \\ 0 \end{bmatrix}^T \left(J\left(\bar{\theta}; 0\right)^+\right)^T\right) \\
&= J\left(\bar{\theta}; 0\right)^+ \begin{bmatrix} \Sigma^{-1}\mathbb{E}\left(\epsilon\epsilon^T\right)\Sigma^{-T} & 0 \\ 0 & 0 \end{bmatrix} \left(J\left(\bar{\theta}; 0\right)^+\right)^T \\
&= J\left(\bar{\theta}; 0\right)^+ \begin{bmatrix} \mathbb{I}_{m^{\text{obj}}} & 0 \\ 0 & 0 \end{bmatrix} \left(J\left(\bar{\theta}; 0\right)^+\right)^T.
\end{aligned}$$

Substituting the expression for the generalized inverse, we can formulate the covariance matrix of the estimated parameters $\theta^* \in \mathbb{R}^n$ as follows:

$$C(\theta^*) := \begin{pmatrix} \mathbb{I} & 0 \end{pmatrix} \begin{pmatrix} F(\theta^*)^T F(\theta^*) & G(\theta^*)^T \\ G(\theta^*) & 0 \end{pmatrix}^{-1} \begin{pmatrix} F(\theta^*)^T F(\theta^*) & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} F(\theta^*)^T F(\theta^*) & G(\theta^*)^T \\ G(\theta^*) & 0 \end{pmatrix}^{-T} \begin{pmatrix} \mathbb{I} \\ 0 \end{pmatrix}.$$

This formulation of the covariance matrix can also be further simplified as demonstrated by the following lemma from Bock et al. [BKK07].

Lemma 5.3.3 : Covariance matrix as a solution of a linear system
Let us define a matrix $M \in \mathbb{R}^{(n+m^{\text{cons}})\times(n+m^{\text{cons}})}$ such that:

$$M^{-1} = \begin{pmatrix} F(\theta^*)^T F(\theta^*) & G(\theta^*)^T \\ G(\theta^*) & 0 \end{pmatrix}^{-1} = \begin{pmatrix} X & Y^T \\ Y & Z \end{pmatrix}$$

where $X \in \mathbb{R}^{n\times n}$, $Y \in \mathbb{R}^{m^{\text{cons}}\times n}$ and $Z \in \mathbb{R}^{m^{\text{cons}}\times m^{\text{cons}}}$.

Then the covariance matrix $C(\theta^*) \in \mathbb{R}^{n\times n}$ of the estimated parameters is equal to the matrix X and satisfies the following linear system:

$$\begin{aligned}
F(\theta^*)^T F(\theta^*) C(\theta^*) + G(\theta^*)^T Y &= \mathbb{I}, \\
G(\theta^*) C(\theta^*) &= 0.
\end{aligned}$$

Proof. See Bock et al. [BKK07]. □

Therefore, the covariance matrix of the estimated parameters can be written as:

$$C(\theta^*) := \begin{pmatrix} \mathbb{I} & 0 \end{pmatrix} \begin{pmatrix} F(\theta^*)^T F(\theta^*) & G(\theta^*)^T \\ G(\theta^*) & 0 \end{pmatrix}^{-1} \begin{pmatrix} \mathbb{I} \\ 0 \end{pmatrix}.$$

5.3.2 Covariance matrix with structure-exploitation

We already saw in Section 3.3 how the multi-experiment structure of a nonlinear least-squares problem can be exploited to efficiently compute the steps in the generalized Gauss-Newton method. In this section, we will briefly discuss how this idea can be further extended to efficiently compute the covariance matrix.

Looking at the linear system to compute the perturbation of the solution $\delta\theta(\epsilon)$, we notice that it strongly resembles the linear system from Section 3.3 to compute the steps Δx at some iterate x of the generalized Gauss-Newton method:

$$\delta\theta = -J\left(\theta^*\right)^+ \begin{bmatrix} \Sigma^{-1}\epsilon \\ 0 \end{bmatrix} \quad \text{vs.} \quad \Delta x = -J^+(x)\phi(x).$$

Recall that the function $\phi : \mathbb{R}^n \to \mathbb{R}^{m^{\text{obj}}+m^{\text{cons}}}$ evaluated at the iterate x defines a vector of objective functions and equality constraints evaluated at x.

This suggests that we can transform the Jacobian matrix $J\left(\theta^*\right)$ in the same way as we did in Section 3.3 to exploit the block structure of the covariance matrix. In particular, we can transform the Jacobian matrix $J\left(\theta^*\right)$ into a block upper triangular form $\tilde{J}(\theta^*)$ of the following form (cf. Section 3.3):

$$\tilde{J}(\theta^*) = \begin{bmatrix} R_{111} & R_{112} & & & & & & \tilde{S}_{11} \\ 0 & R_{122} & & & & & & \tilde{S}_{12} \\ & & R_{211} & R_{212} & & & & \tilde{S}_{21} \\ & & 0 & R_{222} & & & & \tilde{S}_{22} \\ & & & & \ddots & & & \vdots \\ & & & & & R_{N11} & R_{N12} & \tilde{S}_{N1} \\ & & & & & 0 & R_{N22} & \tilde{S}_{N2} \\ & & & & & & & R_\gamma \\ & & & & & & & 0 \end{bmatrix}.$$

Lemma 5.3.4 : Covariance matrix with structure-exploitation
The covariance matrix $C \in \mathbb{R}^{n \times n}$ of the estimated parameters $\theta^* \in \mathbb{R}^n$ can be written in the following block form:

$$C = \begin{bmatrix} C_{11}^{\text{loc}} & C_{12}^{\text{loc}} & \cdots & C_{1N}^{\text{loc}} & C_{1}^{\text{glb}} \\ C_{21}^{\text{loc}} & C_{22}^{\text{loc}} & \cdots & C_{2N}^{\text{loc}} & C_{2}^{\text{glb}} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ C_{N1}^{\text{loc}} & C_{N2}^{\text{loc}} & \cdots & C_{NN}^{\text{loc}} & C_{N}^{\text{glb}} \\ \left(C_{1}^{\text{glb}}\right)^T & \left(C_{2}^{\text{glb}}\right)^T & \cdots & \left(C_{N}^{\text{glb}}\right)^T & C^{\text{glb}} \end{bmatrix}$$

where the sub-matrices are given by:

$$\begin{aligned} C^{\text{glb}} &= P_\gamma R_\gamma^{-1} R_\gamma^{-T} P_\gamma^T, \\ C_i^{\text{glb}} &= -P_i^{\text{loc}} A_i C^{\text{glb}}, \\ C_{ii}^{\text{loc}} &= P_i^{\text{loc}} \left(\begin{bmatrix} E_i F_i E_i^T & E_i F_i \\ F_i E_i^T & F_i \end{bmatrix} + A_i C^{\text{glb}} A_i^T \right) \left(P_i^{\text{loc}}\right)^T, \\ C_{ij, i \neq j}^{\text{loc}} &= P_i^{\text{loc}} A_i C^{\text{glb}} A_j^T \left(P_j^{\text{loc}}\right)^T. \end{aligned}$$

The column permutation matrices P_γ and P_i^{loc} for $i = 1, \ldots, N$ are defined in Section 3.3, and the matrices A_i, E_i and F_i for $i = 1, \ldots, N$ are defined as follows:

$$A_i = \begin{bmatrix} R_{i11} & R_{i12} \\ 0 & R_{i22} \end{bmatrix}^{-1} \begin{bmatrix} \tilde{S}_{i1} \\ \tilde{S}_{i2} \end{bmatrix}, \quad E_i = -R_{i11}^{-1} R_{i12}, \quad F_i = R_{i22}^{-1} R_{i22}^{-T}.$$

Proof. See Schlöder [Sch87] or Nattermann [Nat14]. □

Exploiting the block structure of the covariance matrix to define sub-matrices as we did in Lemma 5.3.4 allows us to selectively compute only the necessary variances required for computing the confidence regions. Furthermore, it also enables us to parallelize the computation of the covariance matrix, which can significantly reduce the computational time.

5.4 Confidence intervals

Once we have the covariance matrix of the estimated parameters, we can define confidence regions where the “true” values of the parameters are expected to lie with

a certain probability. A *confidence region* is defined as an ellipsoid in the parameter space and can be written as:

$$E_N(\alpha) := \left\{\theta \in \mathbb{R}^n \mid g(\theta) = 0, \|f(\theta)\|^2 - \|f(\theta^*)\|^2 \leq \chi_n^2(1-\alpha)\right\}$$

where $\chi_n^2(1-\alpha)$ is the quantile of the chi-squared distribution with n degrees of freedom at the level $1-\alpha$.

However, in practice, it can be difficult to compute this nonlinear confidence region. Therefore, we linearize the constraints and objective function around the estimated parameters θ^* to obtain a *linearized confidence region* as follows:

$$\begin{aligned} E_L(\alpha) := \{\theta \in \mathbb{R}^n \mid & g(\theta^*) + G(\theta^*)(\theta - \theta^*) = 0, \\ & \|f(\theta^*) + F(\theta^*)(\theta - \theta^*)\|^2 - \|f(\theta^*)\| \leq \chi_n^2(1-\alpha)\}. \end{aligned}$$

Since the solution θ^* is assumed to be optimal, the optimality conditions are satisfied at this point. This insight allows us to compute the following representation of the linearized confidence regions [BKK07]:

Lemma 5.4.1 : Linearized confidence region
The linearized confidence region $E_L(\alpha)$ for the estimated parameters $\theta^* \in \mathbb{R}^n$ can be written as:

$$E_L(\alpha) = \left\{\theta^* + \delta\theta \in \mathbb{R}^n \mid \delta\theta = -J^+(\theta^*) \begin{bmatrix} \eta \\ 0 \end{bmatrix},\ \|\eta\|_2^2 \leq \chi_n^2(1-\alpha)\right\}.$$

Proof. See Bock et al. [BKK07]. □

We can then use these linearized confidence regions to compute the confidence intervals of the estimated parameters [BKK07]:

Lemma 5.4.2 : Confidence intervals
The confidence intervals for the estimated parameters $\theta^* \in \mathbb{R}^n$ can be computed as:

$$E_L(\alpha) \subset \bigtimes_{i=1}^{n} \left[\theta_i^* - \sqrt{C_{ii}\chi_n^2(1-\alpha)}, \theta_i^* + \sqrt{C_{ii}\chi_n^2(1-\alpha)}\right]$$

where C_{ii} is the i-th diagonal element of the covariance matrix C.

Proof. See Bock et al. [BKK07]. □

6 Optimal Experimental Design

In Chapter 5, we discussed how measurement errors can affect the quality of the estimated parameters and presented methods to quantify the resulting uncertainty. In this chapter, we turn our attention to strategies for designing new experiments that can potentially improve the quality of the estimated parameters.

The literature offers a wide range of problem formulations and numerical approaches for designing optimal experiments in the context of parameter estimation with measurements of system states at multiple time points [Kör02; FM08; BB08; Jan15; Let+16; BRI19]. However, in this chapter, we would like to design experiments that measure a different kind of data — the values of applied external controls at bifurcation points. We will do this by adapting the approach presented in [Kör02].

We will start our discussion of optimal experimental design (OED) by formulating the optimization problem and discussing various design choices for the objective function. Next, we will present numerical techniques for solving the OED problem with a particular focus on the details of derivative computation. Finally, we will discuss how optimal experimental design and parameter estimation can be integrated in practice to complement one another.

6.1 Introduction

In this thesis, we have considered parameter estimation problems that use measurements of applied external controls at bifurcation points to find the values of the unknown parameters of the corresponding mathematical model. We formulated this parameter estimation problem as an equality-constrained nonlinear least-squares problem in Section 4.3.

Let us now consider a modified version of this parameter estimation problem where we attach sampling weights to each measurement. This means that, for every pair of measured controls $(\tilde{q}_{1_i}, \tilde{q}_{2_i})$ with $i = 1, \ldots, N$, we assign a binary weight $w_i \in \{0, 1\}$,

where $w_i = 1$ if the controls are measured and used for parameter estimation, and $w_i = 0$ otherwise[1].

Problem 6.1.1 : Weighted parameter estimation problem
Find the values of the model parameters $p \in \mathbb{R}^{n_p}$ and for $i = 1, \dots, N$, also the steady states $y_i \in \mathbb{R}^{n_y}$, controls $q_i \in \mathbb{R}^2$ and additional variables $z_i \in \mathbb{R}^{n_z}$ such that they solve the following optimization problem:

$$\begin{aligned} \min_{\theta \in \mathbb{R}^n} \quad & \|f(\theta, w)\|_2^2 := \sum_{i=1}^{N} w_i \left(\frac{\tilde{q}_{1_i} - q_{1_i}}{\sigma_{1_i}} \right)^2 + w_i \left(\frac{\tilde{q}_{2_i} - q_{2_i}}{\sigma_{2_i}} \right)^2 \\ \text{s.t.} \quad & g(\theta) := \begin{cases} \psi(y_i, q_i, p) = 0 \\ \zeta(y_i, q_i, z_i, p) = 0 \end{cases} \qquad i = 1, \dots, N, \end{aligned}$$

where $f : \mathbb{R}^n \to \mathbb{R}^{m^{\text{obj}}}$ and $g : \mathbb{R}^n \to \mathbb{R}^{m^{\text{cons}}}$ are twice continuously differentiable functions and $\theta \in \mathbb{R}^n$ is the vector of optimization variables defined as:

$$\theta = \begin{bmatrix} y_1 & q_1 & z_1 & \cdots & y_N & q_N & z_N & p \end{bmatrix}^T \in \mathbb{R}^n.$$

The measured values of the applied external controls at bifurcation points are given by $\tilde{q}_{1_i}$ and $\tilde{q}_{2_i}$ with variances $\sigma_{1_i}^2$ and $\sigma_{2_i}^2$, respectively for $i = 1, \dots, N$.

When solving the parameter estimation problem to find the unknown parameters, we use all the available measurements to solve the nonlinear least-squares problem — meaning all sampling weights are set to 1.

In order to quantify the uncertainty in the solution $\theta^* \in \mathbb{R}^n$ of the parameter estimation problem, we can compute its covariance matrix $C(\theta^*)$ as described in Section 5.3. The covariance matrix is then given by:

$$C(\theta^*) = \begin{bmatrix} \mathbb{I} & 0 \end{bmatrix} \begin{bmatrix} F(\theta^*)^T F(\theta^*) & G(\theta^*)^T \\ G(\theta^*) & 0 \end{bmatrix}^{-1} \begin{bmatrix} \mathbb{I} \\ 0 \end{bmatrix} \in \mathbb{R}^{n \times n},$$

where the matrices $F(\theta^*)$ and $G(\theta^*)$ denote the first-order derivatives of the objective function and the equality constraints of the weighted parameter estimation problem.

[1] In case the reader is wondering how we might have a measurement $(\tilde{q}_{1_i}, \tilde{q}_{2_i})$ when its corresponding weight $w_i = 0$, we will soon see that the exact value of these measurements do not play a role in our OED problem. This means that we could simply assign dummy values to these measurements.

Due to the multi-experiment structure of the parameter estimation problem, the Jacobian matrices $F \equiv F(\theta^*)$ and $G \equiv G(\theta^*)$ are block-structured matrices of the following form:

$$F = \begin{bmatrix} F_{11}^{\text{loc}} & 0 & \cdots & 0 & F_1^{\text{glb}} \\ 0 & F_{22}^{\text{loc}} & \cdots & 0 & F_2^{\text{glb}} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & F_{NN}^{\text{loc}} & F_N^{\text{glb}} \end{bmatrix} \in \mathbb{R}^{m^{\text{obj}} \times n},$$

$$G = \begin{bmatrix} G_{11}^{\text{loc}} & 0 & \cdots & 0 & G_1^{\text{glb}} \\ 0 & G_{22}^{\text{loc}} & \cdots & 0 & G_2^{\text{glb}} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & G_{NN}^{\text{loc}} & G_N^{\text{glb}} \end{bmatrix} \in \mathbb{R}^{m^{\text{cons}} \times n},$$

where the individual blocks of the Jacobian matrices are then given by:

$$\forall i = 1, \ldots, N : \begin{cases} F_{ii}^{\text{loc}} := \begin{bmatrix} 0 & \cdots & 0 & \frac{\sqrt{w_{1_i}}}{\sigma_{1_i}} & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & \frac{\sqrt{w_{2_i}}}{\sigma_{2_i}} & 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{m_i^{\text{obj}} \times n_i^{\text{loc}}} \\ G_{ii}^{\text{loc}} := \begin{bmatrix} \frac{\partial g}{\partial y_i} & \frac{\partial g}{\partial q_i} & \frac{\partial g}{\partial z_i} \end{bmatrix} \in \mathbb{R}^{m_i^{\text{cons}} \times n_i^{\text{loc}}} \\ F_i^{\text{glb}} := 0 \in \mathbb{R}^{m_i^{\text{obj}} \times n^{\text{glb}}} \\ G_i^{\text{glb}} := \frac{\partial g}{\partial p} \in \mathbb{R}^{m_i^{\text{cons}} \times n^{\text{glb}}} \end{cases}.$$

It is worth noting here that the exact values of the measurements $(\tilde{q}_{1_i}, \tilde{q}_{2_i})$ for $i = 1, \ldots, N$ do not appear in the Jacobian matrices. Therefore, the covariance matrix does not depend on the exact value of the measured controls. This insight allows us to formulate the OED problem as a function of the covariance matrix.

6.2 Problem formulation

The primary goal of optimal experimental design (OED) in our context is to reduce the uncertainty in the estimated parameters by identifying new measurement points that best contribute to this reduction. To achieve this, we propose a set of candidate experiments and associate each with a binary sampling weight indicating whether it should be selected. The OED problem then selects a subset of these candidates under certain constraints by assigning sampling weights to each measurement such that the uncertainty in the parameter estimates is minimized.

Recall that in the parameter estimation problem, each measurement $i = 1, \ldots, N$ was obtained by fixing one control $\tilde{q}_{2_i}$ and varying the other control $\tilde{q}_{1_i}$ until a bifurcation point — identified by a qualitative change in the system's dynamics — was observed. We can extend this idea to experimental design by proposing $\bar{N}$ new values $\tilde{q}_{2_{N+1}} \leq \cdots \leq \tilde{q}_{2_{N+\bar{N}}}$ for the fixed control such that we select those that best improve the parameter estimates. This means that, for each of these fixed control values, the experimenter could vary the other free control to observe bifurcation points. The question is then: which of the candidate measurements offer the most valuable information for minimizing the uncertainty in the parameter estimates?

We can formulate this OED problem as a mixed-integer optimization problem where the objective is to minimize a scalar function $\Phi : \mathbb{R}^{n\times n} \to \mathbb{R}$ of the covariance matrix $C \in \mathbb{R}^{n\times n}$, which is a function of the sampling weights w_i for $i = 1, \ldots, N + \bar{N}$.

Problem 6.2.1 : Mixed-Integer Optimal Experimental Design
Find the values of the steady states $y_i \in \mathbb{R}^{n_y}$, applied controls $q_i \in \mathbb{R}^2$, additional variables $z_i \in \mathbb{R}^{n_z}$ and binary sampling weights $w_i \in \{0, 1\}$ for $i = 1, \ldots, N + \bar{N}$ such that they solve the following mixed-integer optimization problem:

$$\begin{aligned} \min_{y,q,z,w} \quad & \Phi\left(C\right) \\ \text{s.t.} \quad & w_i = 1, \quad i = 1, \ldots, N \\ & w_i \in \{0, 1\}, \quad i = N + 1, \ldots, N + \bar{N} \\ & \sum_{i=N+1}^{N+\bar{N}} w_i \leq M \leq \bar{N}. \end{aligned}$$

Here the covariance matrix C is given by:

$$C = \begin{bmatrix} \mathbb{I} & 0 \end{bmatrix} \begin{bmatrix} F^T F & G^T \\ G & 0 \end{bmatrix}^{-1} \begin{bmatrix} \mathbb{I} \\ 0 \end{bmatrix},$$

where F and G are the Jacobian matrices of the objective function and equality constraints of the following weighted parameter estimation problem evaluated at its solution:

$$\begin{aligned} \min_{y,q,z,p} \quad & \sum_{i=1}^{N+\bar{N}} w_i \left(\frac{\tilde{q}_{1_i} - q_{1_i}}{\sigma_{1_i}}\right)^2 + w_i \left(\frac{\tilde{q}_{2_i} - q_{2_i}}{\sigma_{2_i}}\right)^2 \\ \text{s.t.} \quad & \begin{cases} \psi(y_i, q_i, p) = 0 \\ \zeta(y_i, q_i, z_i, p) = 0 \end{cases} \qquad i = 1, \ldots, N + \bar{N}. \end{aligned}$$

In the OED problem, the values of the model parameters are kept fixed to the solution of the parameter estimation problem. This means that, given the measurement candidates for one of the applied controls, we can then find the corresponding candidates for the other control together with the steady states and additional variables at the candidate bifurcation points. We do this by computing the two-parameter bifurcation diagram as we did in Section 4.4.2. This allows us to now fix most of the optimization variables in the Problem 6.2.1 as follows:

$$
\begin{aligned}
&\text{parameters:} && \hat{p} \in \mathbb{R}^{n_p},\\
&\text{applied controls:} && \hat{q}_1 := \left\{\hat{q}_{1_i} \in \mathbb{R} : i = 1, \dots, N + \bar{N}\right\},\\
& && \hat{q}_2 := \left\{\hat{q}_{2_i} \in \mathbb{R} : i = 1, \dots, N + \bar{N}\right\},\\
&\text{steady states:} && \hat{y} := \left\{\hat{y}_i \in \mathbb{R}^{n_y} : i = 1, \dots, N + \bar{N}\right\},\\
&\text{additional variables:} && \hat{z} := \left\{\hat{z}_i \in \mathbb{R}^{n_z} : i = 1, \dots, N + \bar{N}\right\}.
\end{aligned}
$$

Therefore, we can simplify the mixed-integer OED Problem 6.2.1 to the following integer programming problem:

Problem 6.2.2 : Integer Optimum Experimental Design
Find the values of the binary sampling weights $w \in \{0,1\}^{N+\bar{N}}$ that solve the following integer programming problem:

$$
\begin{aligned}
\min_{w} \quad & \Phi\left(C\left(w, \hat{y}, \hat{q}_1, \hat{q}_2, \hat{z}, \hat{p}\right)\right)\\
\text{s.t.} \quad & w_i = 1, \quad i = 1, \dots, N\\
& w_i \in \{0,1\}, \quad i = N+1, \dots, N+\bar{N}\\
& \sum_{i=N+1}^{N+\bar{N}} w_i \leq M.
\end{aligned}
$$

6.3 Design criteria

Since the goal of our OED strategy is to reduce some function of the covariance matrix, we will first look at different choices for the objective function $\Phi(C) : \mathbb{R}^{n\times n} \to \mathbb{R}$ as presented in [Kör02].

- **A-criterion**: The objective function is defined as the average of the trace of the covariance matrix:
$$\Phi_A(C) = \frac{1}{n}\operatorname{trace}(C).$$

Since the diagonal elements of the covariance matrix contain the variances of the estimated parameters, the main idea of this criterion is to minimize the average variance of the parameter estimates.

- **D-criterion**: The objective function in this case is given by the determinant of the covariance matrix projected onto a k-dimensional subspace of the space of design variables. The projection is given by a full-rank matrix $K \in \mathbb{R}^{n \times k}$ so that the criterion takes the following form:

$$\Phi_D(C) = \left(\det(K^T C K)\right)^{\frac{1}{n}}.$$

  This objective function minimizes the volume of the confidence ellipsoid.
- **E-criterion**: This criterion computes the largest eigenvalue of the covariance matrix:

$$\Phi_E(C) = \max\{\lambda : \lambda \text{ is an eigenvalue of } C\}.$$

  The main idea of this criterion is to minimize the maximum variance of the parameter estimates.
- **M-criterion**: This criterion uses the largest confidence interval of the parameter estimates as the objective function:

$$\Phi_M(C) = \max\left\{\sqrt{C_{ii}},\ i = 1, \ldots, n\right\}.$$

A geometrical visualization of the design criteria for a two-dimensional confidence ellipsoid is shown in the figure 6.1.

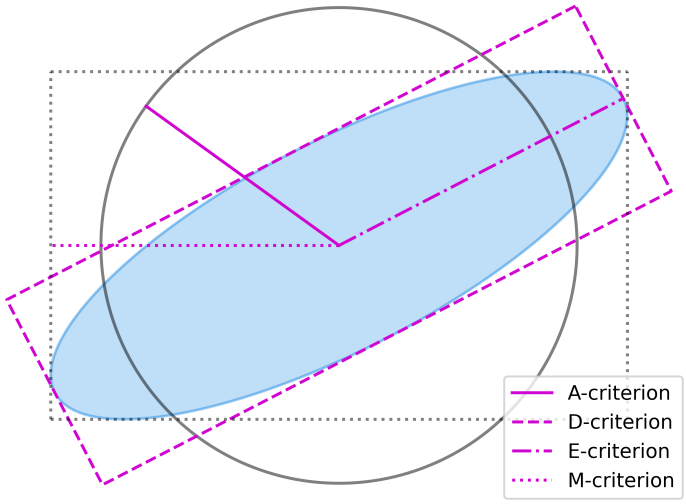


Fig. 6.1.: Confidence ellipsoid (in blue) with geometrical interpretation of A-, D-, E- and M-criteria (taken in modified form from [Wal11]).

It is important to note here that the magnitude of the parameters can have significant influence on the results of the OED. Therefore, it is generally recommended to scale the parameters to have the same order of magnitude before optimizing the covariance matrix.

6.4 Numerical solution methods

There are various numerical techniques for solving integer programming problems [Gre71]. For example, if the number of possible discrete values that the optimization variables may take on is small, we could solve the problem by enumeration. Another possible approach is the branch and bound procedure which branches the optimization problem into smaller sub-problems and uses bounds to eliminate sub-problems that cannot contain the solution.

The solution approach that we consider in this thesis is continuous optimization by integer relaxation as in [Kör02]. This means that we relax the integer requirement on the optimization variables, so that they can now take on continuous values. Then, we obtain a constrained nonlinear optimization problem that can be solved using the SQP method presented in 3.2.1.

In order to use the SQP method to solve the OED problem, we will first translate the integer programming problem into a standard nonlinear optimization problem by integer relaxation. Then, we will discuss how the derivatives of the objective function and constraints with respect to the sampling weights can be computed.

6.4.1 Integer relaxation

We can re-write the integer programming problem defined in Problem 6.2.2 as a nonlinear optimization problem by relaxing the sampling weights. This means that we relax the condition that the weights $w_i \in \{0, 1\}$ take on discrete values to get the continuous weights $w_i \in [0, 1]$ for all $i = 1, \ldots, N + \bar{N}$. Then, the relaxed OED problem can then be written as follows:

Problem 6.4.1 : Optimum Experimental Design with Relaxed Constraints
Find the values of the sampling weights $w \in [0,1]^{N+\bar{N}}$ that solve the following optimization problem:

$$\begin{aligned} \min_{w} \quad & \Phi\left(C\left(w, \hat{y}, \hat{q}_1, \hat{q}_2, \hat{z}, \hat{p}\right)\right) \\ \text{s.t.} \quad & w_i = 1, \quad i = 1, \ldots, N \\ & w_i \in [0,1], \quad i = N+1, \ldots, N+\bar{N} \\ & \sum_{i=N+1}^{N+\bar{N}} w_i \leq M. \end{aligned}$$

In order to then get the discrete solutions from the continuous solutions of this optimization problem, we can use greedy rounding as a heuristic to translate the relaxed weights to binary sampling weights. This means that we set the $\bar{N}$ measurement points with the largest weights to 1 and the rest to 0.

6.4.2 Derivative computation

In order to solve the relaxed OED problem using the SQP method, we need to compute the first-order derivatives of its objective function and constraints with respect to the sampling weights. However, since the objective function and constraints of the OED problem, in turn, depend on the sensitivity of the solutions of the underlying parameter estimation problem, the computation of the derivatives can be challenging and requires closer attention.

The directional derivative of the objective function $\Phi(C)$ with respect to the continuous weights w in the direction Δw can be computed using the chain rule as follows:

$$\frac{d\Phi\left(C\left(w\right)\right)}{dw}\Delta w = \frac{d\Phi\left(C\right)}{dC}\frac{dC}{dJ}\frac{dJ}{dw}\Delta w, \tag{6.1}$$

where the Jacobian matrix J is given by $J = \begin{bmatrix} F \\ G \end{bmatrix}$.

The individual derivative terms in equation (6.1) can be written as follows:

$$\Delta\Phi = \frac{d\Phi}{dC}\Delta C, \quad \Delta C = \frac{dC}{dJ}\Delta J, \quad \Delta J = \frac{dJ}{dw}\Delta w.$$

A detailed analysis of the derivative computation can be found in the dissertation by Körkel [Kör02]. Here, we will only state the final form of the derivatives.

Derivatives of objective function w.r.t. covariance matrix

Given a symmetric and positive-semi-definite matrix $C \in \mathbb{R}^{n \times n}$, the derivatives of the objective function $\Phi(C)$ with respect to C are given by:

- **Derivative of A-criterion**:

$$\frac{d\Phi_A(C)}{dC}\Delta C = \frac{1}{n}\operatorname{trace}(\Delta C),$$

- **Derivative of D-criterion**:

$$\frac{d\Phi_D(C)}{dC}\Delta C = \frac{1}{k}\left(\det\left(K^T C K\right)\right)^{\frac{1}{k}} \sum_{i,j=1}^{k}\left(\left(K^T C K\right)^{-1}\right)_{ij}\left(K^T \Delta C K\right)_{ij},$$

- **Derivative of E-criterion**:

$$\frac{d\Phi_E(C)}{dC}\Delta C = v^T \Delta C v,$$

  where $v \in \mathbb{R}^n$ is the unit eigenvector for the largest simple eigenvalue of C.

- **Derivative of M-criterion**: In this case, the min-max problem can be reformulated by defining an additional variable $v \in \mathbb{R}$ which satisfies the inequality constraints:

$$v \geq \sqrt{C_{ii}}, \quad i = 1, \ldots, N + \bar{N}.$$

  Then, the derivative of each inequality constraint with respect to the covariance matrix is given by:

$$\frac{d\sqrt{C_{ii}}}{dC}\Delta C = \frac{1}{2\sqrt{C_{ii}}}\Delta C, \quad i = 1, \ldots, N + \bar{N}.$$

Since the directional derivatives of the objective function with respect to the covariance matrix depend on the matrix ΔC, we will now discuss how ΔC can be computed.

Derivatives of covariance matrix w.r.t. Jacobian matrices

Consider the covariance matrix $C \in \mathbb{R}^{n \times n}$ given by:

$$C = \begin{bmatrix}\mathbb{I} & 0\end{bmatrix}\begin{bmatrix}F^T F & G^T \\ G & 0\end{bmatrix}^{-1}\begin{bmatrix}\mathbb{I} \\ 0\end{bmatrix}. \tag{6.2}$$

The directional derivative of this covariance matrix C with respect to the Jacobian matrix $J \in \mathbb{R}^{(m^{\text{obj}}+m^{\text{cons}})\times n}$ is then given by:

$$\Delta C = \frac{dC}{dJ}\Delta J = -\begin{bmatrix}\mathbb{I} & 0\end{bmatrix} Z^{-1}\Delta Z Z^{-1}\begin{bmatrix}\mathbb{I}\\ 0\end{bmatrix},$$

where the matrix $\Delta Z \in \mathbb{R}^{(m^{\text{cons}}+n)\times(m^{\text{cons}}+n)}$ is defined as:

$$\Delta Z := \frac{dZ}{dJ}\Delta J = \begin{bmatrix}\Delta F^T F + F^T \Delta F & \Delta G^T\\ \Delta G & 0\end{bmatrix}.$$

It remains now to compute the derivatives ΔF and ΔG of the Jacobian matrices F and G with respect to the design variables.

Derivatives of objective Jacobian F w.r.t. sampling weights

The Jacobian matrix F of the objective function in the parameter estimation problem is given by:

$$F = \overbrace{\begin{bmatrix} F_1^{\text{loc}} & 0 & 0 & \cdots & 0 & F_1^{\text{glb}}\\ 0 & F_2^{\text{loc}} & 0 & \cdots & 0 & F_2^{\text{glb}}\\ \vdots & & \ddots & & \vdots & \vdots\\ 0 & \cdots & \cdots & \cdots & F_{N+\bar{N}}^{\text{loc}} & F_{N+\bar{N}}^{\text{glb}}\end{bmatrix}}^{n \text{ columns}} \Bigg\updownarrow {\scriptstyle m^{\text{obj}} \text{ rows}}$$

where the matrices F_i^{loc} and F_i^{glb} are given by:

$$\forall i = 1,\ldots,N+\bar{N}: \begin{cases} F_i^{\text{loc}} := \begin{bmatrix} 0 & \cdots & 0 & \frac{\sqrt{w_i}}{\sigma_{1_i}} & 0 & 0 & \cdots & 0\\ 0 & \cdots & 0 & 0 & \frac{\sqrt{w_i}}{\sigma_{2_i}} & 0 & \cdots & 0\end{bmatrix} \in \mathbb{R}^{m_i^{\text{obj}}\times n_i^{\text{loc}}}\\ F_i^{\text{glb}} := 0 \in \mathbb{R}^{m_i^{\text{obj}}\times n^{\text{glb}}}\end{cases}.$$

Then its directional derivatives with respect to the sampling weights w are:

$$\frac{dF}{dw}\Delta w = \begin{bmatrix} \frac{dF_1^{\text{loc}}}{dw_1}\Delta w_1 & 0 & 0 & \cdots & 0 & \frac{dF_1^{\text{glb}}}{dw_1}\Delta w_1\\ 0 & \frac{dF_2^{\text{loc}}}{dw_2}\Delta w_2 & 0 & \cdots & 0 & \frac{dF_2^{\text{glb}}}{dw_2}\Delta w_2\\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots\\ 0 & 0 & 0 & \cdots & \frac{dF_{N+\bar{N}}^{\text{loc}}}{dw_{N+\bar{N}}}\Delta w_{N+\bar{N}} & \frac{dF_{N+\bar{N}}^{\text{glb}}}{dw_{N+\bar{N}}}\Delta w_{N+\bar{N}}\end{bmatrix}$$

where the individual derivatives with respect to the weights are given by:

$$\forall i = 1, \ldots, N + \bar{N} : \begin{cases} \dfrac{dF_i^{\text{loc}}}{dw_i} \Delta w_i = \begin{bmatrix} 0 & \cdots & 0 & \frac{1}{2\sqrt{w_i}\sigma_{1_i}} \Delta w_i & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & \frac{1}{2\sqrt{w_i}\sigma_{2_i}} \Delta w_i & 0 & \cdots & 0 \end{bmatrix} \\ \dfrac{dF_i^{\text{glb}}}{dw_i} \Delta w_i = 0 \end{cases}.$$

Derivatives of constraint Jacobian G w.r.t. sampling weights

The Jacobian matrix G of the equality constraints in the parameter estimation problem is given by:

$$G = \overbrace{\begin{bmatrix} G_1^{\text{loc}} & 0 & 0 & \cdots & 0 & G_1^{\text{glb}} \\ 0 & G_2^{\text{loc}} & 0 & \cdots & 0 & G_2^{\text{glb}} \\ \vdots & & \ddots & & \vdots & \vdots \\ 0 & \cdots & \cdots & \cdots & G_{N+\bar{N}}^{\text{loc}} & G_{N+\bar{N}}^{\text{glb}} \end{bmatrix}}^{n \text{ columns}} \Bigg\updownarrow {\scriptstyle m^{\text{cons}} \text{ rows}}$$

where the matrices G_i^{loc} and G_i^{glb} are given by:

$$\forall i = 1, \ldots, N + \bar{N} : \begin{cases} G_i^{\text{loc}} := \begin{bmatrix} \frac{\partial \psi_i}{\partial y_i} & \frac{\partial \psi_i}{\partial q_i} & 0 \\ \frac{\partial \zeta_i}{\partial y_i} & \frac{\partial \zeta_i}{\partial q_i} & \frac{\partial \zeta_i}{\partial z_i} \end{bmatrix} \in \mathbb{R}^{m_i^{\text{cons}} \times n_i^{\text{loc}}} \\ G_i^{\text{glb}} := \begin{bmatrix} \frac{\partial \psi_i}{\partial p} \\ \frac{\partial \zeta_i}{\partial p} \end{bmatrix} \in \mathbb{R}^{m_i^{\text{cons}} \times n^{\text{glb}}} \end{cases}.$$

Then its directional derivatives with respect to the sampling weights w are:

$$\frac{dG}{dw} \Delta w = 0.$$

6.4.3 Sequential procedure of OED

In practice, the process of parameter estimation and experimental design is an iterative one. First, experiments are conducted in order to obtain measurement data. Then, the model is calibrated by estimating its parameters using the given measurements. The parameter estimates are then used to guide the design of new

experiments to obtain more measurements such that the quality of the parameter estimates can be improved. Using the new measurement data, the model can then be re-calibrated. This sequential procedure can be repeated until reliable parameter estimates are obtained. This process is illustrated in the flow chart shown in Figure 6.2.

During this iterative process, the underlying mathematical model may change to account for new insights from the experiments. Old measurements are usually not discarded in the OED process. Instead, they are included with fixed sampling weights along with the candidate measurements.

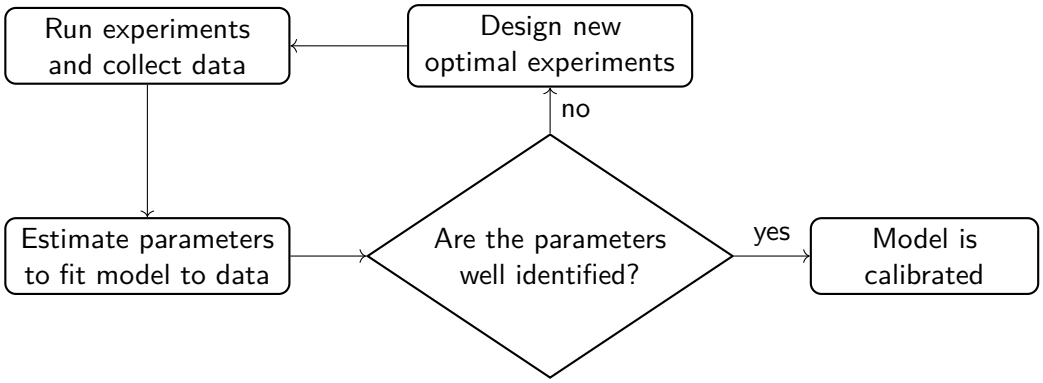


Fig. 6.2.: Flow chart of the sequential procedure of OED. Figure is adapted from [Jan15].

Part III

Numerical Implementation and Results

7 Applications

To the best of our knowledge, there has been limited research exploring the use of measured bifurcation points for estimating model parameters. Therefore, in this chapter, we aim to motivate our parameter estimation approach by highlighting a diverse range of applications where bifurcation points have been experimentally observed. For each application, we will review relevant experimental studies from the literature where external controls were varied to measure saddle-node and Hopf bifurcation points. Note that this list is by no means exhaustive.

7.1 Semiconductor lasers

Semiconductor lasers are widely used in optical communication systems, barcode scanners, laser printers, and many other applications because they are small, highly efficient, and inexpensive to produce. However, these lasers are also highly sensitive to external perturbations such as optical injection and external optical feedback. These perturbations can lead to complex dynamical behaviors in the laser output, including bistability, periodic oscillations, and chaos [Wie+05].

Single-mode semiconductor lasers receiving optical injection from a stable laser source are conceptually simple systems that still exhibit complex dynamics. For this reason, they have been widely studied both theoretically and experimentally [Wig03; Wie+05; Kra06; AHA07]. The dynamics of these lasers have been described by a system of three ordinary differential equations, which show good agreement with experimental observations [Wie+03]. Saddle-node and Hopf bifurcations can be observed in these lasers by varying input controls such as injection field strength K and detuning ω of the distributed feedback laser. Figure 7.1 compares the experimental bifurcation diagram produced using these input controls with the corresponding theoretical bifurcation diagram. This laser system is a good example of experimental measurements that satisfy the requirements of our parameter estimation framework. In Section 8.2.1, we will demonstrate this in a case study using synthetic measurements of saddle-node bifurcation points and estimate the parameters of the mathematical model.

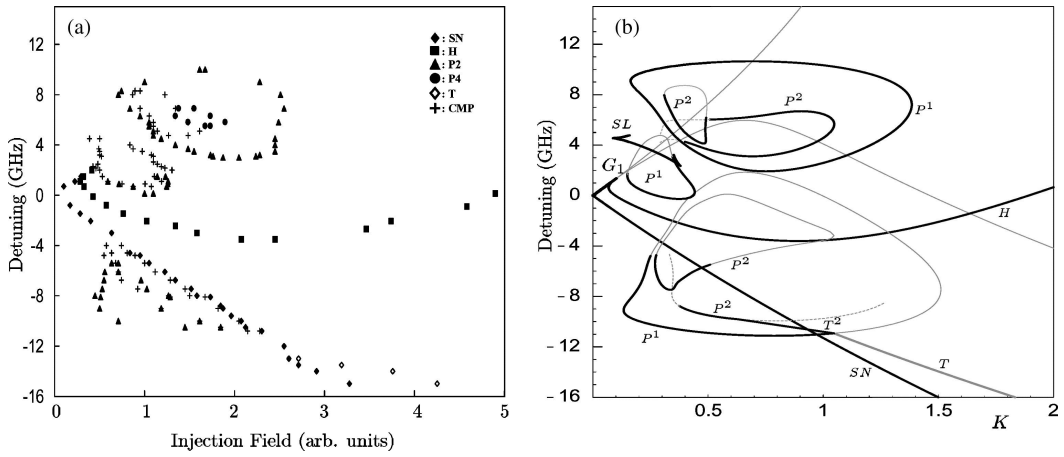


Fig. 7.1.: Bifurcation diagram of a distributed feedback semiconductor laser with optical injection. (a) Experimental bifurcation diagram. (b) Theoretical bifurcation diagram. Figures were taken with permission from [Wie+03].

7.2 Biochemistry

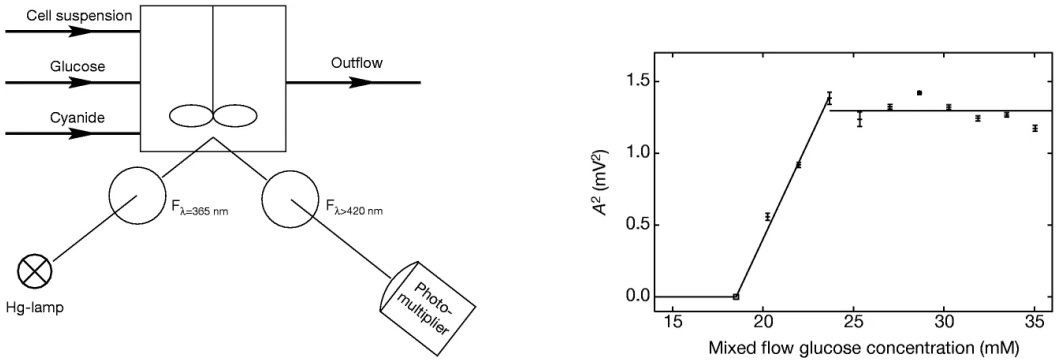
Laboratory studies of various biochemical processes have revealed experimentally-controlled bistable and oscillatory behavior. Some of these processes have also been investigated theoretically by developing mathematical models and studying their bifurcations. In this section, we present a selection of these biochemical systems.

Glycolysis

Glycolysis is a vital metabolic pathway found in most carbon-based life forms. It is responsible for metabolizing glucose, fructose, and other carbohydrates to produce pyruvate and NADH (nicotinamide adenine dinucleotide) molecules [MB03; Cha21]. In the presence of oxygen, pyruvate and NADH are transported into the mitochondria, where ATP (adenosine triphosphate) is generated. In contrast, under anaerobic conditions, NADH is reoxidized by reducing pyruvate into lactate, which also generates ATP, but with a higher cost. This dual functionality of the glycolytic pathway highlights the key role it plays in energy metabolism, particularly in critical illnesses such as hypoxemia, infection, or inflammation.

Oscillatory behavior in the glycolytic pathway, especially in living yeast cells, has been well-documented for decades [DSH99; Gus+14]. For example, Dano et al. [DSH99] demonstrated in an experiment that NADH fluorescence transitions between a stable steady state and sustained oscillations in a suspension of yeast cells as the in-flow rates of glucose and cyanide are varied (see Figure 7.2(a) for

the experimental setup). We can see this experimentally observed Hopf bifurcation point, as a function of the glucose in-flow rate, in Figure 7.2(b). Similar results were also observed by Richard et al. [Ric+94] when varying the in-flow rate of cyanide. In order to study this dynamical system theoretically, mathematical models describing the glycolytic pathway in yeast cells were developed by Hynne et al. [HDG01] and Gustavsson et al. [Gus+14]. We propose that, with the experimentally measured Hopf bifurcation points, the parameters of these mathematical models can be estimated using our parameter estimation framework.



(a) Experimental setup **(b)** Bifurcation diagram

Fig. 7.2.: Experimental study of sustained oscillations in yeast cells for varying glucose and cyanide inputs. (a) Experimental setup for observing NADH fluorescence. (b) Experimental bifurcation diagram showing the square of the amplitude of the oscillations as a function of the glucose in-flow rate. A Hopf bifurcation is observed at the mixed flow glucose concentration of 18.5mM. Figures were taken with permission from [DSH99].

Peroxidase-oxidase reaction

Peroxidases are enzymes that catalyze the oxidation of a wide range of organic and inorganic substrates by breaking down hydrogen-peroxide (H_2O_2). These enzymes are found in various organisms, where they play an important role in removing the toxic H_2O_2 molecules that are produced as a byproduct during respiration. However, some peroxidases can also use molecular oxygen (O_2) as a substrate and catalyze the oxidation of hydrogen donors such as NADH. This process is known as the peroxidase-oxidase reaction [Sen05].

A widely used choice of the peroxidase enzyme is the horseradish peroxidase (HRP) extracted from the roots of horseradishes. Experimental studies of this reaction under varying external controls, such as oxygen concentration and temperature, and modifiers, such as DCP (2,4-dichlorophenol) and MB (methylene blue), have shown both sustained oscillations and steady states in the NADH fluorescence [AHF90].

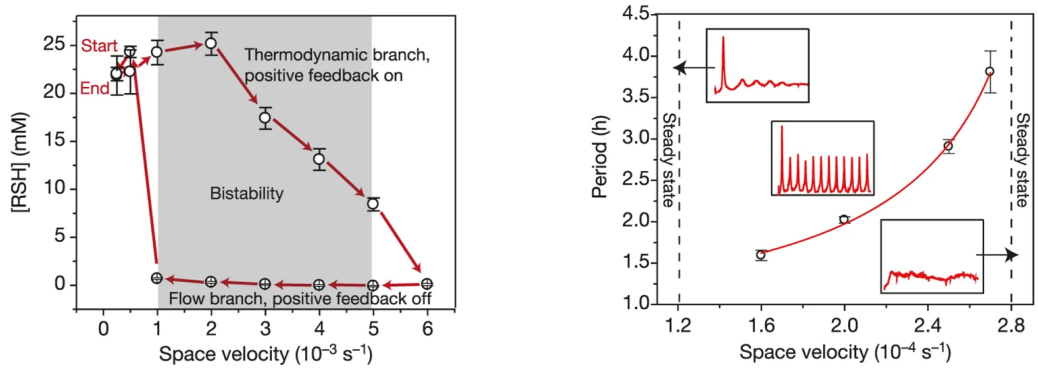
The transition between sustained oscillation and stable equilibrium in this dynamical system has been attributed to a supercritical Hopf bifurcation.

Many detailed mathematical models of the peroxidase-oxidase reaction based on the reaction kinetics have been proposed to capture the observed dynamics, notably [SLA88; AL90; BSO01]. Using Hopf bifurcation points observed in experimental studies, we suggest that the unknown parameters of these mathematical models can be estimated with our parameter estimation framework. In fact, we will demonstrate this in a case study in Section 8.3.1, where we generate artificial measurements of Hopf bifurcation points and estimate the parameters of the mathematical model developed by Steinmetz et al. [SLA88] using these measurements.

Synthetic biology

One of the main goals of synthetic biology is to understand the principles of self-organization found in many organic chemical reactions from an engineering perspective. Consequently, some studies have developed networks of chemical reactions that can be investigated and controlled both theoretically and experimentally to explore their emergent dynamics [Sem+15; Sem+16].

For example, Semenov et al. [Sem+16] investigated autocatalytic behavior, bistability, and periodic oscillations in the concentrations of organic thiols and amides by constructing a modular chemical reaction network. Sets of chemical reactions were designed to trigger exponential growth in organic thiols (RSH), amplified by an autocatalytic reaction involving cystamine (CSSC) and L-alanine ethyl thioester (AlaSEt). However, the growth of the thiols could also potentially be suppressed by acrylamide and maleimide. To study this system experimentally, all reactants and products were mixed in a continuously stirred tank reactor (CSTR) while allowing the in-flow and out-flow of chemical species. External controls of this system included the space velocity (ratio of flow rate to reactor volume), pH of the mixture, temperature, and concentrations of the chemical species. Figure 7.3 shows results from experiments where the space velocity was varied to observe hysteresis and sustained oscillations in RSH. In the same paper, Semenov et al. also presented a simple kinetic model to theoretically analyze the observed dynamics. In Section 8.2.2, we will study this system in a case study and demonstrate how our parameter estimation framework can be used to fit this model using artificially-generated measurements of saddle-node bifurcation points.



(a) Hysteresis curve **(b)** Oscillations

Fig. 7.3.: Experimental results from a CSTR fed with AlaSEt, CSSC and maleimide under different flow rates [Sem+16]. The open circles in both figures represent the experimental data and error bars correspond to their standard deviation. (a) Hysteresis curve of monitored [RSH] steady states for different space velocities. (b) Stability and period of the oscillations in [RSH] for varying space velocities. The solid line shows a hyperbolic fit to the data. Figures were taken with permission from [Sem+16].

Gene regulatory networks

The interactions between transcription factors and gene expression levels of mRNA inside cells are mapped by a gene regulatory network. These networks often serve as a "blueprint" of the structural and functional molecular interactions inside the cell and can guide experiments [EDH14].

One of the best-studied examples of a gene regulatory network is the lac operon in Escherichia coli (E. coli) bacteria. The lac operon is a set of genes required for transporting lactose into cells and breaking it down inside the cell. While E. coli bacteria generally prefer glucose as their carbon source, in the absence of glucose, they use lactose as an alternative source of carbon. As a result, the lac operon is repressed in the presence of glucose but activated in the absence of glucose and presence of lactose [JM61; Ozb+04]. Experimental studies varying extracellular concentrations of glucose and lactose in single cells have shown bistability and hysteresis in the gene expression levels of the lac operon [Ozb+04]. Numerous mathematical models have also been proposed to theoretically study this observed dynamical behavior [TB05; NP08; Lei21; Mac+15; SMZ07]. Based on these studies, we suggest that unknown parameters of the mathematical models of the lac operon can be estimated with our parameter estimation framework using measurements of externally-controlled glucose and lactose concentrations at bifurcation points.

7.3 Ecology

One of the main goals of ecology is to understand the temporal dynamics of ecosystems and the interactions between different species. Various field and laboratory studies across different communities have shown complex dynamics in their populations, including bistability, periodic oscillations, and chaotic behavior [Jos+73; Fus+00; Yos+03; Bec+05; Fus+05]. In fact, some experimental studies with ciliate-bacteria [Jos+73], rotifer-algae [Fus+00], and flour beetles [Cos+97; Den+97] have identified possible bifurcations in their population dynamics. As a result, mathematical models based on predator-prey interactions have been developed to understand the mechanisms behind these observations [Yos+03; Fus+00; Fus+05].

For instance, Fussmann et al. [Fus+00] studied the interactions of planktonic rotifers feeding on unicellular green algae in a chemostat and monitored their population dynamics under varying experimental conditions. The resource concentration in the inflow medium and the dilution rate of the chemostat were used as external controls that could be varied. Experimental studies of this dynamical system revealed extinction, coexistence, and periodic oscillations in the populations of the species under different conditions (see Figure 7.4). Consequently, Fussmann et al. also formulated a system of four ordinary differential equations to mathematically describe the observed population dynamics. In Section 8.3.2, we will demonstrate, with the help of synthetic measurement data, how the external controls of this dynamical at experimentally observed Hopf bifurcation points can be used to estimate unknown parameters of the mathematical model.

7.4 Neuroscience

Various neurological conditions, such as Parkinson's disease, epilepsy, gait abnormalities, and altered circadian rhythms, are characterized by periodic oscillations [Mil+89]. Mathematically, these oscillations can be described using Hopf bifurcations arising as a result of changes in some parameters in the neuronal circuits.

Mathematical models of neurons are typically based on the physics of action potential generation and propagation across cell membranes. Some of the most well-known neuron models include the Hodgkin-Huxley model [HH52], Hindmarsh-Rose model [HR84], Morris-Lecar model [ML81], and Fitzhugh-Nagumo model [Fit61; NAY62]. These models of neurons can be implemented in silicon-based electronic circuits to mimic the electrical activity of biological neurons [Pat+99; Bin+06; LPS10;

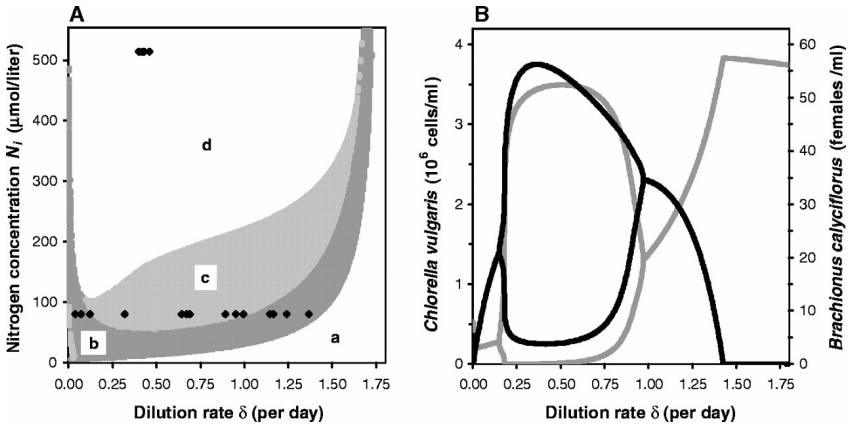


Fig. 7.4.: Predicted and experimental model outcomes of a predator-prey model described by Fussmann et al. [Fus+00]. (A) Two-parameter bifurcation diagram with a classification of regions based on their model outcomes. Region a: extinction of predator or extinction of both predator and prey. Region b: coexistence at an equilibrium. Region c: coexistence on a stable limit cycle. Region d: extinction due to extreme cycles. The black points represent experimentally measured bifurcation points. (B) One-parameter bifurcation diagram showing the transition from stable steady states to stable limit cycles and back through Hopf bifurcations. The curves of the limit cycle represent the maxima and minima of the predator (plotted in black) and prey (plotted in gray) concentrations. Figures were taken with permission from [Fus+00].

Rut+20]. This allows bifurcations in the mathematical models to be experimentally investigated in their hardware counterparts by varying certain controls, such as the externally applied current, in the silicon neuron [Pat+99; Bin+06; LPS10]. Therefore, we propose that the bifurcation points observed in these experiments can be used to estimate the parameters of their mathematical models with the help of our parameter estimation framework.

7.5 Chemical engineering

Bistability, hysteresis, and periodic oscillations have been observed in a wide range of chemical engineering processes, such as bioethanol production [Eln+06], catalytic oxidation of carbon monoxide [GL87; SB87], ethane oxidation on a Pt/Al_2O_3 pellet [HL87], and budding yeast [ZBH01]. In these cited references, experimental systems were set up with controllable parameters, such as feed concentration, temperature, and dilution rate, to observe saddle-node or Hopf bifurcations, and mathematical models describing the chemical reactions were proposed.

For example, consider the case of bioethanol production. Elnashaie et al. [Eln+06] designed an experimental setup to efficiently ferment sugars produced by hydrolysis, thereby producing more ethanol. The feed sugar concentration and the dilution rate of the fermenter were used as external controls. By varying these controls, the authors observed a transition between stable equilibrium and sustained oscillations through a Hopf bifurcation in their experiments. This suggests that our parameter estimation framework is well-suited for estimating the parameters of such models of chemical engineering processes.

8 Case Studies

In Chapter 7, we presented several fields of application where bifurcations have been observed experimentally. In this chapter, we investigate four of these applications as detailed case studies. These include two systems exhibiting saddle-node bifurcations and two systems exhibiting Hopf bifurcations.

We begin by outlining the general approach that we follow in all our case studies. This includes an explanation of how we generate artificial measurement data, derive initial guesses for the optimization variables, and solve the multi-experiment nonlinear least-squares problem. Then, for each case study, we describe the experimental system with its external controls, set up the parameter estimation problem, and discuss the results obtained.

8.1 General approach

Artificial measurement data

Since we do not have access to real experimental data for our case studies, we generate artificial, noisy measurements of the external controls at bifurcation points to demonstrate our parameter estimation procedure. We do this by initializing the model parameters to values reported in the literature and computing a curve of saddle-node or Hopf bifurcation points using a bifurcation software, such as MATCONT[1] [Gov+19]. From the computed curve, we then select a small subset of bifurcation points (around 15-20) and add Gaussian noise with zero mean and a 5% standard deviation to each point to emulate measurement noise.

Initial guesses for optimization variables

For each case study, we focus on estimating two key parameters while assuming the remaining parameters are known. To solve this parameter estimation problem,

[1] Although our software also supports the computation of bifurcation points, we use a different software to generate the measurement data to avoid potential bias in our procedure.

we first need suitable initial guesses for all optimization variables. We do this by initializing the model parameters using Latin hypercube sampling [MBC79] and then computing the initial guesses for the remaining optimization variables following the procedure described in Section 4.4.2.

This procedure begins by computing a steady-state solution of the system through simulation and optimization. The steady state is then numerically continued using the deflated continuation method (see Section 2.3) by varying the primary control to obtain a curve of steady states. If bifurcation points are detected on this curve, they are selected and numerically continued using either the deflated continuation method or the pseudo-arclength method (see Section 2.2) by additionally varying the secondary control. This results in a curve of bifurcation points that trace the measurement data. For each artificially measured bifurcation point, we select the closest bifurcation point on the predicted curve as the corresponding initial guess for the parameter estimation problem.

Parameter estimation procedure

Once the initial guesses are computed, we solve the parameter estimation problem using the generalized Gauss-Newton method with structure exploitation, as described in Chapter 3. The optimization procedure is terminated when the L2-norm of the search direction falls below a specified threshold (e.g., 10^{-4}). To evaluate the quality of the estimated parameters, we also compute the covariance matrix and 95% confidence intervals of the parameter estimates using the sensitivity analysis approach detailed in Chapter 5.

8.2 Saddle-node bifurcations

In this section, we investigate two case studies of bistable systems exhibiting saddle-node bifurcations: a semiconductor laser system [Wie+03] and an autocatalytic reaction network from synthetic biology [Sem+16].

8.2.1 Semiconductor lasers

Simpson et al. [Sim03] conducted controlled experiments on a single-mode semiconductor laser by injecting highly monochromatic light from a stable master laser to observe its dynamical behavior. By varying the applied injection strength and

the offset (detuning) frequency between the master and semiconductor lasers, they observed qualitative changes, such as hysteresis and oscillations, in the optical spectra of the laser output. This approach allowed them to experimentally trace a curve of saddle-node and Hopf bifurcation points in the two-dimensional parameter space defined by the injection strength K and detuning frequency ω (see Section 7.1 for more details).

To theoretically study the observed dynamics of the semiconductor laser system, Wieczorek et al. [WKL99; Wie+03] proposed the following mathematical model:

$$\dot{E} = K + \left(\frac{1}{2}(1 + i\alpha)\, n - i\omega\right) E, \tag{8.1a}$$

$$\dot{n} = -2\Gamma n - (1 + 2Bn)\left(|E|^2 - 1\right), \tag{8.1b}$$

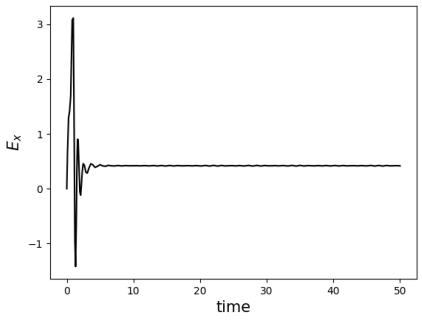
where $E = E_x + iE_y$ represents the complex electric field amplitude, and n represents the population inversion. The external controls of this model are the injection strength K and the detuning frequency ω (marked in pink). The remaining parameters α, Γ, and B represent the rescaled damping rate of the intrinsic resonance, the rescaled cavity lifetime of photons, and the line width enhancement factor, respectively. Wieczorek et al. [Wie+03] showed that this model can exhibit saddle-node and Hopf bifurcations, consistent with the experimental findings of Simpson et al. [Sim03]. We now illustrate how our parameter estimation framework can be applied to estimate the parameters of this model.

Suppose we would like to estimate the values of two unknown model parameters α and B (marked in blue in equations (8.1)). We attempt to solve this parameter estimation problem from 25 random initial guesses obtained by Latin hypercube sampling in the following intervals:

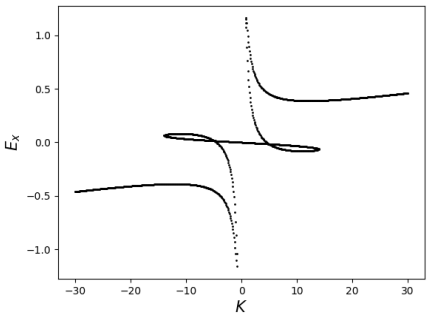
$$\alpha \in [1, 10], \quad B \in [0.01, 0.1],$$

where the true values of the model parameters lie within these intervals.

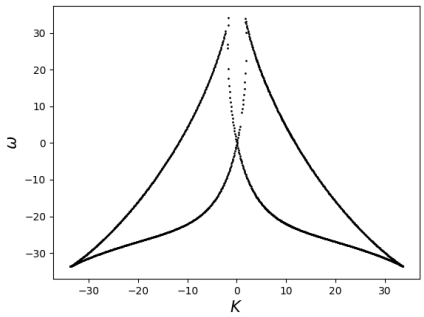
To estimate the parameters using our parameter estimation framework, we follow the general approach described in Section 8.1. We first create a set of artificial, noisy measurements for the external controls K and ω at saddle-node bifurcation points. Then, we generate initial guesses for all the optimization variables and solve the parameter estimation problem to obtain the solutions with their confidence intervals. To illustrate these steps, we show plots from the initial guess generation and the results of the parameter estimation for a representative example in Figure 8.1.



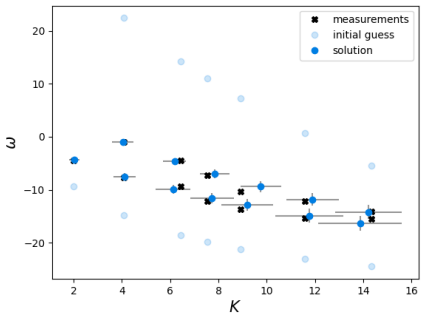
(a) Time-integration of the ODEs



(b) One-parameter bifurcation diagram



(c) Two-parameter bifurcation diagram



(d) Parameter estimation results

Fig. 8.1.: Example of the initial guess generation procedure and parameter estimation results for the semiconductor laser model. For the initial guess generation procedure (see Figures (a) - (c)), the parameters were initialized to $\alpha = 6.2535$ and $B = 0.0619$, and the controls to one of the data points $K = 6.4432$ and $\omega = -4.5445$. Figure (d) shows the results of the parameter estimation with the measurement data marked with black crosses, the initial guesses for the controls marked by light blue circles and the solutions marked by dark blue circles. The bars on the solutions show the confidence intervals of the estimated controls. Moreover, the parameters converged to $\alpha = 2.6199 \pm 0.1971$ and $B = 0.0292 \pm 0.0099$, where the true values are $\alpha = 2.6$ and $B = 0.0295$.

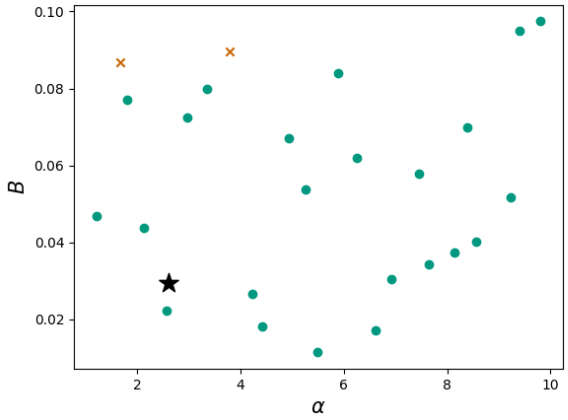


Fig. 8.2.: Convergence region of the parameter estimation method for the semiconductor laser model. The black star indicates the true solution, the green points indicate the initial guesses that converged to an optimal solution and the red crosses indicate initial guesses that failed to find a solution.

Furthermore, the results obtained by trying to solve the parameter estimation problem from 25 different initial guesses for the model parameters are shown in Figure 8.2. This figure visualizes the convergence region for our parameter estimation method applied to the semiconductor laser model. The green points in this plot indicate the initial guesses that successfully converged to an optimal solution, while the red crosses denote those that did not.

8.2.2 Autocatalytic reaction network

In an experimental and theoretical study, Semenov et al. [Sem+16] designed a modular chemical reaction network that exhibits both bistability and periodic oscillations in the concentrations of organic thiols and amides. Their experiments used a continuously stirred tank reactor (CSTR) to investigate an autocatalytic reaction network consisting of organic thiols (RSH), cystamines (CSSC), L-alanine ethyl thioesters (AlaSEt) and maleimide. In this network, interactions between CSSC and AlaSEt generate small amounts of cysteamine (CSH), an organic thiol capable of self-amplification. However, the presence of maleimide, which reacts quickly with thiolates, delays the autocatalytic growth until the maleimide is depleted from the reactor. By continuously monitoring the total concentration of thiols (RSH), the experiments revealed that the reaction network can display bistable behavior [Sem+16]. The external controls — such as the space velocity (i.e., the ratio of the flow rate to the reactor volume) and the inflow concentrations of CSSC, AlaSEt, and maleimide — were used to modulate the system's dynamics (see Section 7.2 for more details).

To theoretically study the observed behaviors, Semenov et al. also proposed the following system of three ordinary differential equations to describe the autocatalytic network:

$$\dot{A} = -k_1 SA - k_2 IA - k_3 A - k_0 A + k_4 S, \tag{8.2a}$$

$$\dot{I} = -k_0 I_0 - k_0 I - k_2 IA, \tag{8.2b}$$

$$\dot{S} = k_0 S_0 - k_0 S - k_4 S - k_1 SA. \tag{8.2c}$$

Here, A represents the concentration of organic thiols RSH, I the concentration of maleimides, and S the concentration of the thioester AlaSEt. The concentration of CSSC is assumed to be constant and therefore, does not directly show up in the model. The inflow concentrations of maleimides and AlaSEt are given by I_0 and S_0 respectively (marked in pink), and k_0 denotes the space velocity. The parameters k_1, k_2, k_3 and k_4 are the reaction rate constants. This model has been shown to produce

bistability and oscillations in the concentration of thiols through saddle-node and Hopf bifurcations, consistent with experimental observations [Sem+16]. We will now demonstrate how to calibrate this mathematical model using our parameter estimation framework.

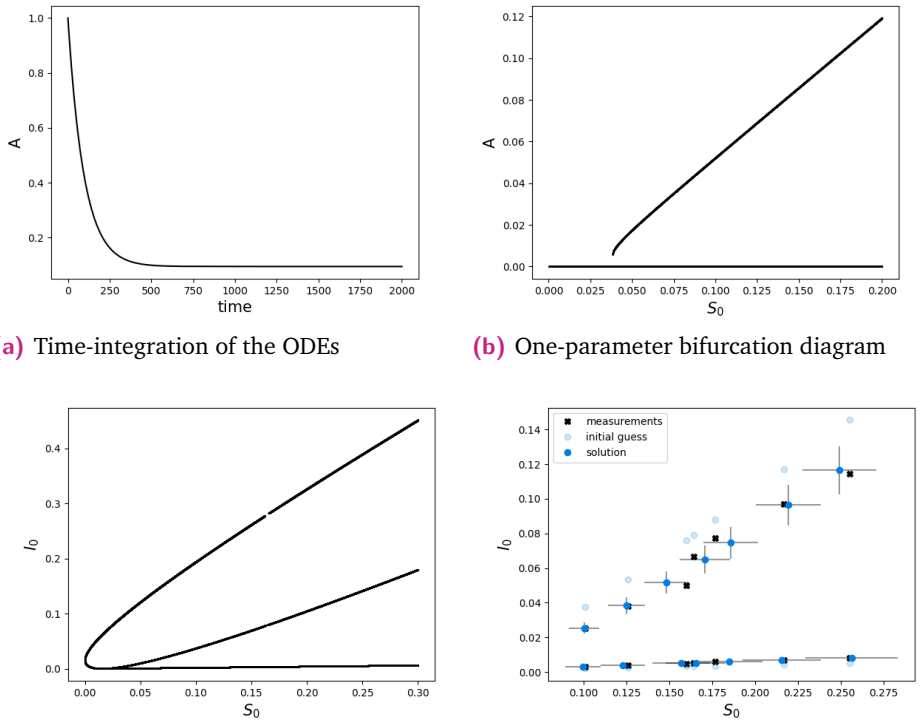
Let us choose the inflow concentrations of AlaSEt S_0 and maleimides I_0 as our external controls (marked in pink in equations (8.2)). Suppose we would like to estimate the values of two unknown model parameters k_0 and k_1 (marked in blue in equations (8.2)) using our parameter estimation framework. We attempt to solve this parameter estimation problem from 25 random initial guesses obtained by Latin hypercube sampling in the following intervals:

$$k_0 \in [0.001, 0.01] \quad \text{and} \quad k_1 \in [0.1, 1],$$

where the true values of the model parameters lie within these intervals.

To estimate the parameters using our parameter estimation framework, we follow the general approach described in Section 8.1. This means that we first create a set of artificial, noisy measurements for the external controls S_0 and I_0 at saddle-node bifurcation points. Then, we generate initial guesses for all the optimization variables and finally, solve the parameter estimation problem to obtain the solutions with their confidence intervals. To illustrate these steps, we show the plots from the initial guess generation and the results of the parameter estimation procedure for a representative example in Figure 8.3.

Moreover, the results obtained by trying to solve the parameter estimation problem from 25 different initial guesses for the model parameters are shown in Figure 8.4. This figure visualizes the convergence region for our parameter estimation method applied to the autocatalytic reaction network model. The green points in this plot indicate the initial guesses that successfully converged to an optimal solution, while the red crosses denote those that did not.



(a) Time-integration of the ODEs

(b) One-parameter bifurcation diagram

(c) Two-parameter bifurcation diagram

(d) Parameter estimation results

Fig. 8.3.: Example of the initial guess generation procedure and parameter estimation results for the autocatalytic reaction model. For the initial guess generation procedure (see Figures (a) - (c)), the parameters were initialized to $k_0 = 0.0070$ and $k_1 = 0.6822$ and the controls to one of the data points $S_0 = 0.1639$ and $I_0 = 0.0053$. Figure (d) shows the results of the parameter estimation with the measurement data marked with black crosses, the initial guesses for the controls marked by light blue circles and the solutions marked by dark blue circles. The bars on the solutions show the confidence intervals of the estimated controls. Moreover, the parameters converged to $k_0 = 0.0029 \pm 0.0003$ and $k_1 = 0.2541 \pm 0.04$ where the true values are $k_0 = 0.003$ and $k_1 = 0.25$ respectively.

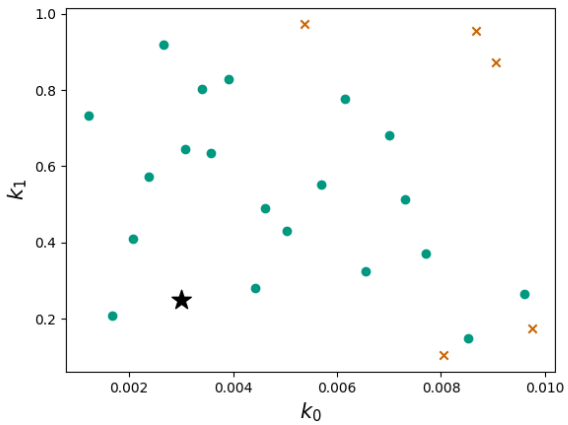


Fig. 8.4.: Convergence region of the parameter estimation method for the autocatalytic reaction model. The black star indicates the true solution, the green points indicate the initial guesses that converged to an optimal solution and the red crosses indicate initial guesses that failed to find a solution.

8.3 Hopf bifurcations

In this section, we explore two case studies of oscillatory systems exhibiting Hopf bifurcations: a peroxidase-oxidase reaction system [DOP79] and a predator-prey system [Fus+00].

8.3.1 Peroxidase-oxidase reaction

A peroxidase-oxidase reaction is a biochemical process in which a peroxidase enzyme uses molecular oxygen (O_2) as a substrate to catalyze the oxidation of hydrogen donors such as NADH. To study the dynamics of this reaction, Degn, Olsen and Perram [OD77; DOP79] conducted experiments in a semi-open, well-stirred biochemical reaction chamber. By controlling the inflow of oxygen and NADH into the chamber, they observed a range of complex dynamical behaviors, such as bistability, periodic oscillations and even chaos, in the oxygen concentration inside the reaction chamber (see Section 7.2 for more details).

To explain these complex dynamics, the authors also proposed a mathematical model based on mass-action kinetics, describing the interactions between four species: oxygen, NADH and two intermediates. The model is given by the following system of ordinary differential equations:

$$\dot{A} = -k_1 ABX - k_3 ABY + k_7 - k_{-7} A \tag{8.3a}$$

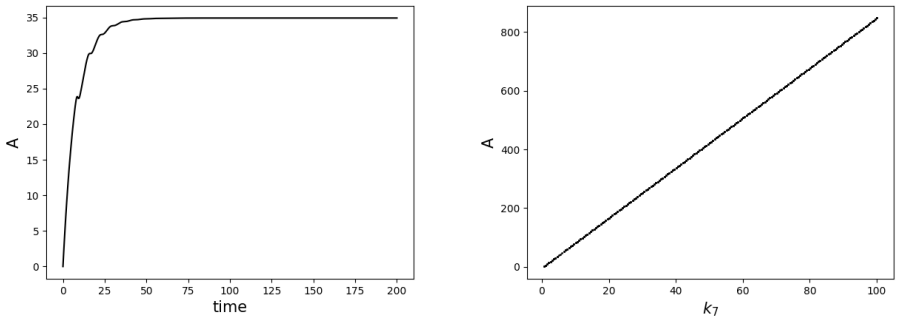
$$\dot{B} = -k_1 ABX - k_3 ABY + k_8 \tag{8.3b}$$

$$\dot{X} = k_1 ABX - 2k_2 X^2 + 2k_3 ABY - k_4 X + k_6 \tag{8.3c}$$

$$\dot{Y} = -k_3 ABY + 2k_2 X^2 - k_5 Y. \tag{8.3d}$$

Here, A and B denote the concentrations of O2 and NADH, respectively, and X and Y are the concentrations of the two intermediates. The inflow rates of oxygen and NADH are represented by the controls k_7 and k_8 respectively (marked in pink). The remaining parameters k_1, k_2, k_3, k_4, k_5, k_6 and k_{-7} are reaction rate constants. Numerical simulations have shown that this model can reproduce the qualitative dynamics observed experimentally, including the emergence of periodic oscillations through Hopf bifurcations [DOP79; SLA88].

In the following, we demonstrate how our parameter estimation framework can be applied to estimate parameters of this model using measurements of the external controls at Hopf bifurcation points.



(a) Time-integration of the ODEs

(b) One-parameter bifurcation diagram

(c) Two-parameter bifurcation diagram

(d) Parameter estimation results

Fig. 8.5.: Example of the initial guess generation procedure and parameter estimation results for the peroxidase-oxidase model. For the initial guess generation procedure (see Figures (a) - (c)), the parameters were initialized to $k_1 = 0.3133$ and $k_5 = 1.8440$ and the controls to one of the data points $k_7 = 4.5907$ and $k_8 = 0.4869$. Figure (d) shows the results of the parameter estimation with the measurement data marked with black crosses, the initial guesses for the controls marked by light blue circles and the solutions marked by dark blue circles. The bars on the solutions show the confidence intervals of the estimated controls. Moreover, the parameters converged to $k_1 = 0.1568 \pm 0.1195$ and $k_5 = 1.0762 \pm 0.5596$ where the true values are $k_1 = 0.1631021$ and $k_5 = 1.104$ respectively.

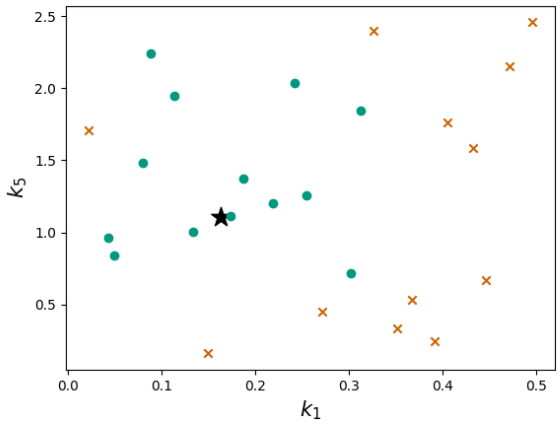


Fig. 8.6.: Convergence region of the parameter estimation method for the peroxidase-oxidase model. The black star indicates the true solution, the green points indicate the initial guesses that converged to an optimal solution and the red crosses indicate initial guesses that failed to find a solution.

Suppose we would like to estimate the values of two unknown model parameters k_1 and k_5 (marked in blue in equations (8.3)). We attempt to solve this parameter estimation problem from 25 random initial guesses obtained by Latin hypercube sampling in the following intervals:

$$k_1 \in [0.01, 0.5] \quad \text{and} \quad k_5 \in [0.1, 2.5],$$

where the true values of the model parameters lie within these intervals.

To estimate the parameters using our parameter estimation framework, we follow the general approach described in Section 8.1. This means that we first create a set of artificial, noisy measurements for the external controls k_7 and k_8 at Hopf bifurcation points. Then, we generate initial guesses for all the optimization variables and finally, solve the parameter estimation problem to obtain the solutions with their confidence intervals. To illustrate these steps, we show the plots from the initial guess generation and the results of the parameter estimation procedure for a representative example in Figure 8.5.

Additionally, the results obtained by trying to solve the parameter estimation problem from 25 different initial guesses for the model parameters are shown in Figure 8.6. This figure visualizes the convergence region for our parameter estimation method applied to the peroxidase-oxidase reaction model. The green points in this plot indicate the initial guesses that successfully converged to an optimal solution, while the red crosses denote those that did not. For this model, one of the reasons many of our attempts at estimating the parameters failed is because the initial guess generation procedure could not find any Hopf bifurcation points along the curve of steady states within the given range $k_7 \in [0, 100]$.

8.3.2 Predator-prey system

Fussmann et al. [Fus+00] experimentally investigated the dynamics of a predator-prey system in a nitrogen-filled chemostat, using cultures of planktonic rotifers (Brachionus calyciflorus) as predators and unicellular green algae (Chlorella vulgaris) as prey. The rotifers feed on the algae, while the algae rely on nitrogen in the environment to grow. In their experiments, the authors varied two external controls: the nitrogen concentration in the inflow medium and the dilution rate (the fraction of the culture volume that is continuously removed). By tuning these controls, they observed qualitative changes in the system's behavior, including stable coexistence,

sustained oscillations in population densities, and species extinction (see Section 7.3 for more details).

To explain the observed dynamics in the predator-prey system, Fussmann et al. [Fus+00] proposed the following mathematical model:

$$\dot{N} = \delta(N_i - N) - \frac{b_C N}{K_C + N} C \tag{8.4a}$$

$$\dot{C} = \frac{b_C N}{K_C + N} C - \frac{b_B C}{K_B + C} \frac{B}{\epsilon} - \delta C \tag{8.4b}$$

$$\dot{R} = \frac{b_B C}{K_B + C} R - (\delta + m + \lambda) R \tag{8.4c}$$

$$\dot{B} = \frac{b_B C}{K_B + C} R - (\delta + m) B. \tag{8.4d}$$

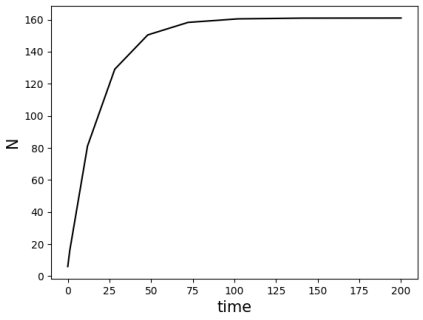
In this model, N represents the nitrogen concentration, C the prey (Chlorella vulgaris) population, R the reproducing predator (Brachionus calyciflorus) population, and B the total predator population. The external controls are the nitrogen concentration N_i and the dilution rate δ (marked in pink). The model parameters b_C and b_B denote the maximum growth rates, while K_C and K_B are the half-saturation constants. Additionally, the parameter m denotes the predator's mortality rate, λ its reproductive decay rate, and ϵ its assimilation efficiency. Numerical simulations have shown that this model can exhibit Hopf bifurcations consistent with the experimental findings [Fus+00]. We will now demonstrate how we can use our parameter estimation framework to estimate parameters of this mathematical model.

Suppose we would like to estimate the values of two unknown model parameters b_C and b_B (marked in blue in equations (8.4)). We attempt to solve this parameter estimation problem from 25 random initial guesses obtained by Latin hypercube sampling in the following intervals:

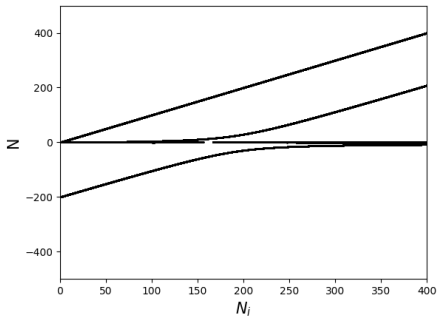
$$b_C \in [0, 6] \quad \text{and} \quad b_B \in [0, 6],$$

where the true values of the model parameters lie within these intervals.

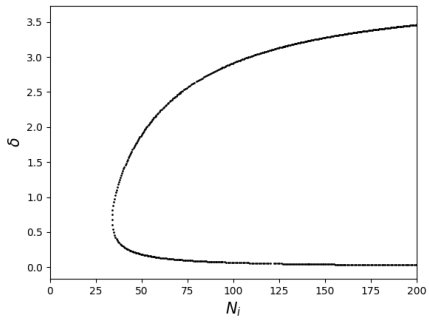
To estimate the parameters using our parameter estimation framework, we follow the general approach described in Section 8.1. This means that we first create a set of artificial, noisy measurements for the external controls N_i and δ at Hopf bifurcation points. Then, we generate initial guesses for all the optimization variables and finally, solve the parameter estimation problem to obtain the solutions with their confidence intervals. To illustrate these steps, we show the plots from the initial guess generation and the results of the parameter estimation procedure for a representative example in Figure 8.7.



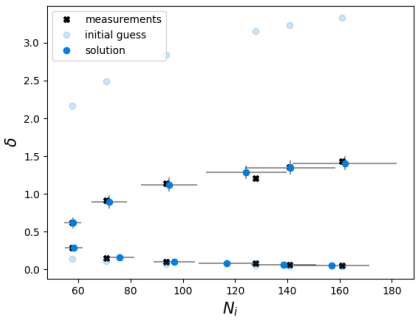
(a) Time-integration of the ODEs



(b) One-parameter bifurcation diagram



(c) Two-parameter bifurcation diagram



(d) Parameter estimation results

Fig. 8.7.: Example of the initial guess generation procedure and parameter estimation results for the predator-prey model. For the initial guess generation procedure (see Figures (a) - (c)), the parameters were initialized to $b_C = 5.6361$ and $b_B = 4.4483$ and the controls to one of the data points $N_i = 161.0696$ and $\delta = 0.0560$. Figure (d) shows the results of the parameter estimation with the measurement data marked with black crosses, the initial guesses for the controls marked by light blue circles and the solutions marked by dark blue circles. The bars on the solutions show the confidence intervals of the estimated controls. Moreover, the parameters converged to $b_C = 3.1717 \pm 0.3434$ and $b_B = 2.2494 \pm 0.1130$ where the true values are $b_C = 3.3$ and $b_B = 2.25$ respectively.

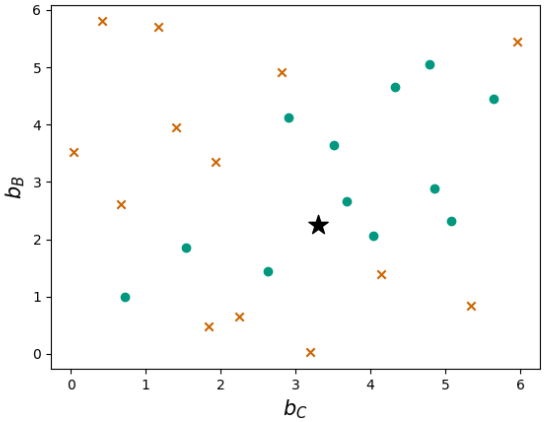


Fig. 8.8.: Convergence region of the parameter estimation method for the predator-prey model. The black star indicates the true solution, the green points indicate the initial guesses that converged to an optimal solution and the red crosses indicate initial guesses that failed to find a solution.

The results obtained by trying to solve the parameter estimation problem from 25 different initial guesses for the model parameters are shown in Figure 8.8. This figure visualizes the convergence region for our parameter estimation method applied to the predator-prey model. The green points in this plot indicate the initial guesses that successfully converged to an optimal solution, while the red crosses denote those that did not. The failed parameter estimation attempts were mostly because no Hopf bifurcation points could be found in the given parameter range $N_i \in [0, 1000]$. In a few cases, the optimization algorithm also failed to converge because the initial guesses were not good enough.

9 Implementation

One of the key contributions of this thesis is the development of an open-source software package called `bifit` that implements numerical methods for parameter estimation using bifurcation points. Our software is specifically designed to estimate unknown parameters of mathematical models using measurements of applied external controls at bifurcation points. In this chapter, we provide an overview of the software design, highlight its main features, and explain how to use it in practice.

9.1 Overview

As part of this thesis, we have developed `bifit`, a modular, user-friendly Python package that brings together the parameter estimation framework from Chapter 4 and the *a posteriori* sensitivity analysis from Chapter 5. Using our software, users can easily and reliably estimate unknown parameters of ODE-based models using noisy measurements of two external controls at saddle-node or Hopf bifurcation points. In particular, `bifit` automates every step of the workflow:

- Automatic initialization of all optimization variables
- Efficient structure-exploiting solvers for the parameter estimation problem
- Computation of confidence intervals for the estimated parameters.

Furthermore, to allow researchers to easily use, adapt and extend our software, we have made the complete source code of `bifit` openly available on GitHub[1], accompanied by extensive documentation and a suite of examples. By implementing most numerical routines ourselves, we have also ensured that `bifit` has few external dependencies (see Table 9.1 for the full list and their use-cases). To help navigate the codebase, Figure 9.1 shows the directory structure, and Table 9.2 provides a concise overview of the main modules and respective functionalities.

[1] `https://github.com/h-varma/bifit`

Module	Version	Our uses	Reference
`scipy`	1.14.1	optimization, integration, linear algebra	[Vir+20]
`numpy`	2.1.0	linear algebra, array manipulation	[Har+20]
`autograd`	1.7.0	automatic differentiation	[MDA15]
`osqp`	0.6.7	sparse optimization solver	[Ste+20]
`matplotlib`	3.9.2	plotting	[Hun07]
`pandas`	2.2.2	data manipulation	[McK10]

Tab. 9.1.: External dependencies of the `bifit` package.

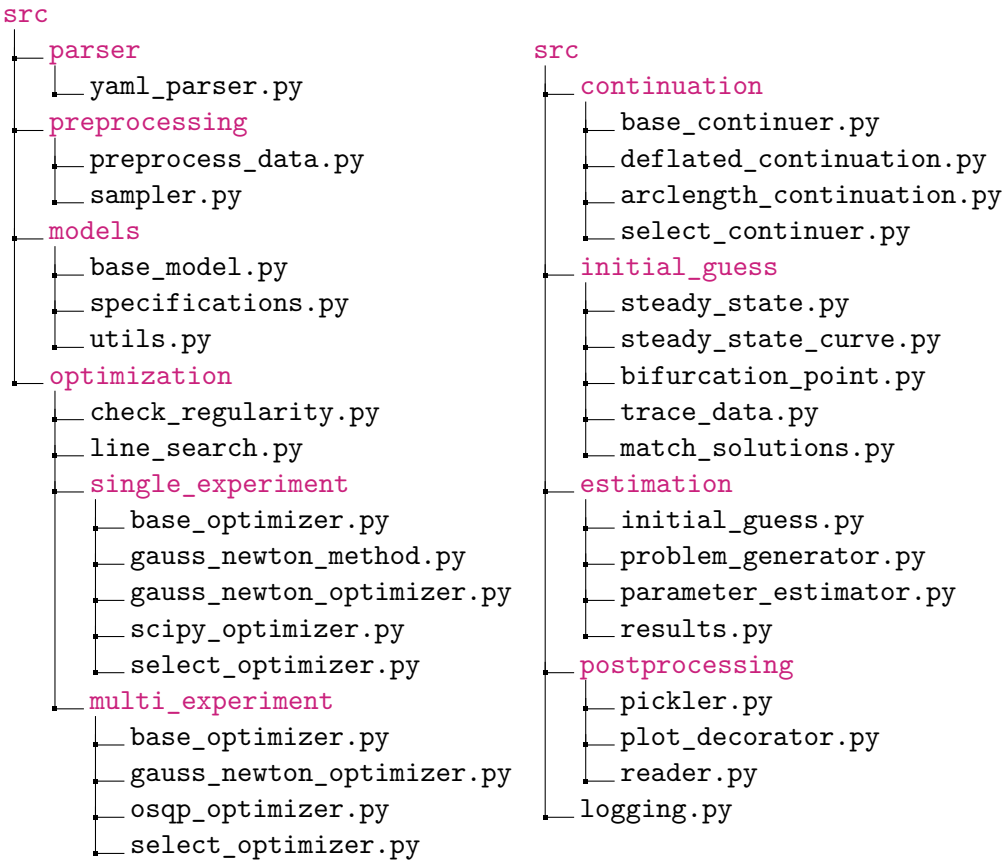


Fig. 9.1.: The directory structure of the internal source-code in `bifit`.

Module	Description
`parser`	parses YAML files containing model specifications
`preprocessing`	loads and preprocesses measurement data, and samples parameter guesses for initialization
`models`	defines an abstract base model class and a base model specifications class
`optimization`	contains our implementation of optimization routines, and interfaces to `scipy` and `osqp` routines
`continuation`	contains our implementation of numerical continuation methods for bifurcation analysis
`initial_guess`	contains the steps of the initial guess generation strategy for parameter estimation
`estimation`	computes initial guesses, formulates and solves parameter estimation problem, and gets results
`postprocessing`	postprocesses the parameter estimation results, including plotting, saving and reading results

Tab. 9.2.: Overview of the main modules in the `bifit` source code.

9.2 Core workflow and key features

Our software streamlines the entire parameter estimation pipeline into a single, automated workflow (see Figure 9.2). This means that given the model definition, measurement data and user-defined settings, `bifit` can execute the following steps:

(i) Generate initial guesses for the optimization variables

- Compute a steady state solution of the model.
- Perform one-parameter continuation to construct a bifurcation diagram.
- Allow the user to select one of the detected bifurcation Points
- Perform two-parameter continuation to get a curve of bifurcation points.
- Identify points on this curve that best align with the measurements.

For more details on this step, see Section 4.4.2.

(ii) Solve the parameter estimation problem

- Solve the multi-experiment nonlinear least-squares problem using a generalized Gauss-Newton solver.

- Exploit the block-structure of the Jacobian matrices to efficiently solve the QP subproblems.

For more details on the parameter estimation problem, see Section 4.3.

(iii) Compute the confidence intervals of the estimated parameters

- Once the parameter estimation solver finds a solution, compute the covariance matrix of the estimated parameters.
- Using the covariance matrix, compute an approximation of the confidence intervals of the estimated parameters.

For more details on the *a posteriori* sensitivity analysis, see Chapter 5.

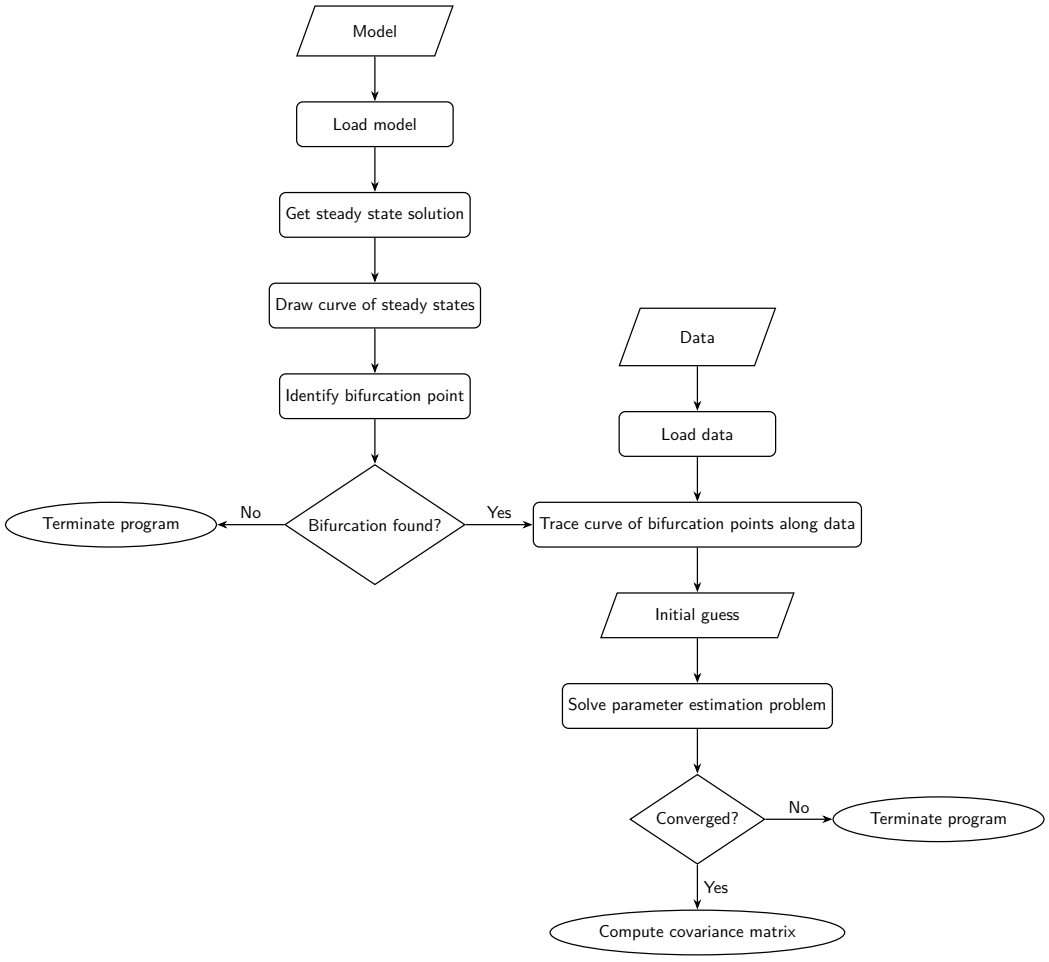


Fig. 9.2.: A flow chart of the core workflow as implemented in `bifit`.

Behind the scenes of this workflow, we implement two different numerical continuation methods to enable the user to map the complex bifurcation landscapes reliably. Our optimization module also offers a wide array of nonlinear optimization solvers

both for standard nonlinear optimization problems and multi-experiment nonlinear least-squares problems. This includes our own implementation of the generalized Gauss-Newton method, as well as interfaces to the `scipy` and `osqp` libraries.

In the following sections, we unpack the details of the continuation and optimization modules of `bifit`. For additional implementation notes, please refer to the online documentation and the source code of our software.

9.2.1 Continuation module

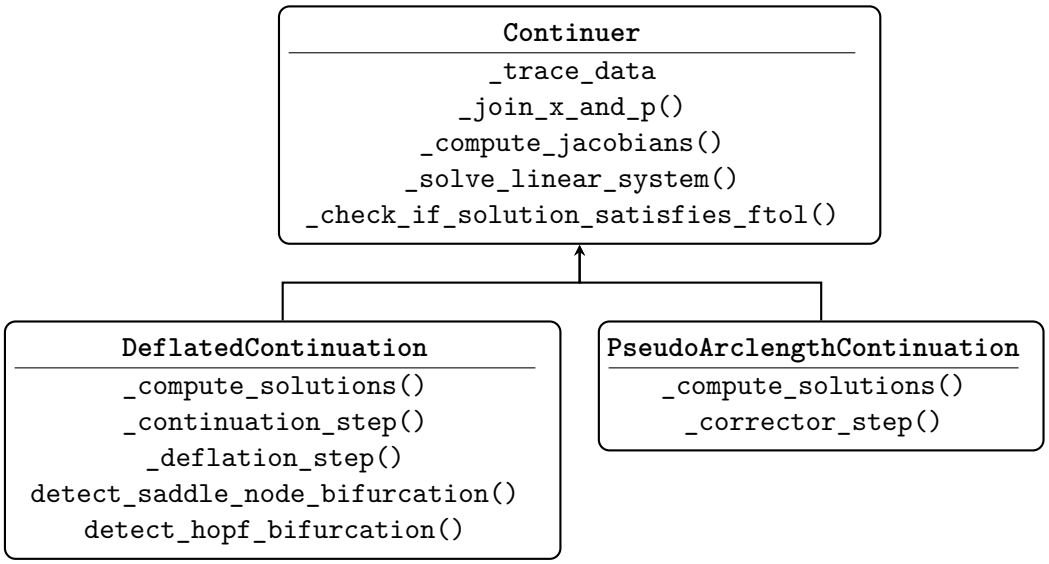


Fig. 9.3.: Visualization of the class structure and inheritance relations of the numerical continuation routines. Only the public and protected methods of the classes are represented in this diagram.

In the process of generating initial guesses for the optimization variables of the parameter estimation problem, `bifit` needs to compute one-parameter and two-parameter bifurcation diagrams using numerical continuation methods. As introduced in Chapter 2, we support two different numerical continuation algorithms for this purpose — the pseudo-arclength method (see Algorithm 2.1) and the deflated continuation method (see Algorithm 2.2) — implemented as subclasses of a common base `Continuer` class (see Figure 9.3):

- **One-parameter continuation**: Due to its robustness against the choice of the initial steady state, we use the deflated continuation method to trace the curve of steady states and automatically detect all bifurcation points on this steady state curve.

- **Two-parameter continuation**: Once a relevant bifurcation point is selected, the user may choose either continuation method to map out the two-parameter curve of bifurcation points that pass through (or close to) the measured bifurcation points.

9.2.2 Optimization module

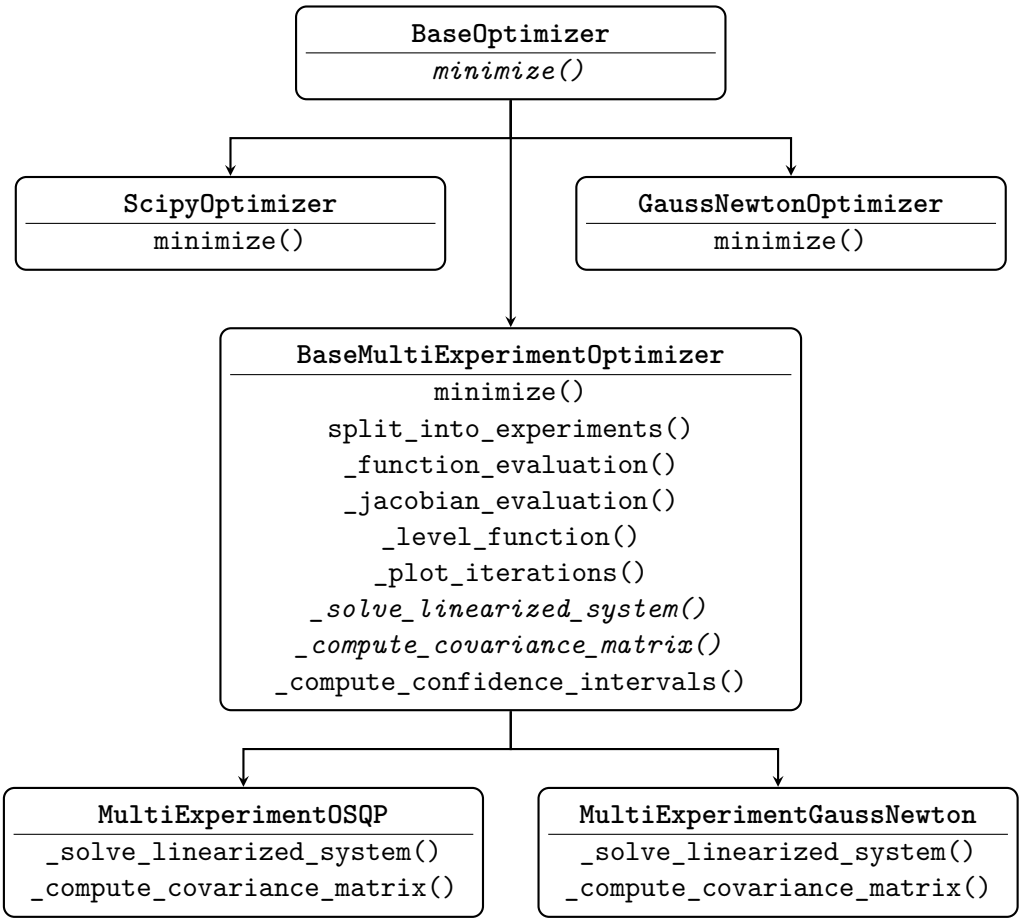


Fig. 9.4.: Visualization of the class structure and inheritance relations of the optimization routines. Only the public and protected methods of the classes are represented in this diagram. The method names in italics are abstract function declarations that need to be defined in each child class.

Throughout both the initial guess generation and the final parameter-estimation phases, `bifit` tackles a variety of nonlinear optimization problems: computing the steady state solution, refining the crude approximation of the bifurcation point and solving the multi-experiment nonlinear least-squares parameter estimation problem. To handle this diverse array of optimization problems, we provide two different

optimization submodules, each inheriting from a common `BaseOptimizer` base class (see Figure 9.4):

- **Single-experiment optimization**: This submodule targets standard nonlinear optimization problems and nonlinear least-squares problems without a multi-experiment structure. It includes our own implementation of the generalized Gauss-Newton method and an interface to the `scipy.optimize.minimize` and `scipy.optimize.least_squares` functions of the `scipy` library [Vir+20].
- **Multi-experiment optimization**: This submodule is designed to solve nonlinear least-squares problems with multi-experiment structure. It includes our own implementation of the generalized Gauss-Newton method with two different options for solving the QP subproblems in each iterate: the efficient structure-exploitation scheme described in Section 3.3 and an interface to a generic sparse QP solver from the `osqp` library [Ste+20].

9.3 Getting started with `bifit`

To apply our parameter estimation framework in practice, we need to first install the `bifit` package following the instructions in the `README.md` file of the GitHub repository[2]. Once the package is installed, for each model, we prepare a new working directory containing three files:

- `data.dat`
- `model_equations.py`
- `meta_parameters.yaml`

In the same directory, we place a `main.py` script that runs the complete workflow, including loading the data, generating initial guesses, solving the parameter estimation problem, and computing the confidence intervals.

To illustrate how to set up the working directory, we will consider the autocatalytic reaction network model by Semenov et al. [Sem+16] as an example (see also Section 8.2.2). Recall that this model captures the interactions between organic thiols

[2] https://github.com/h-varma/bifit

(RSH), maleimides and thioesters (AlaSEt) in a CSTR described by the following equations:

$$\begin{aligned}\dot{A} &= -k_1SA - k_2IA - k_3A - k_0A + k_4S,\\ \dot{I} &= -k_0I_0 - k_0I - k_2IA,\\ \dot{S} &= k_0S_0 - k_0S - k_4S - k_1SA\end{aligned}$$

where A represents the concentrations of RSH, I the concentrations of maleimides and S the concentrations of AlaSEt. We used the inflow concentrations of maleimides I_0 and AlaSEt S_0 (marked in pink) measured at saddle-node bifurcation points as our artificially-generated measurement data to estimate the parameters k_0 and k_1 (marked in blue).

Data file (`data.dat`)

The experimentally measured values of the external controls at bifurcation points need to be stored in a `.dat` file. The first line of this file lists the names of the two controls separated by a space, and the subsequent lines store their numerical values.

For the autocatalytic reaction model, the data file `data.dat` looks like:

```
S0 I0
0.1000 0.0031,0.0246
0.1250 0.0039,0.0376
0.1500 0.0047,0.0517
0.1750 0.0055,0.0667
0.2000 0.0063,0.0824
0.2250 0.0071,0.0987
0.2500 0.0079,0.1154
```

In this case, since we observed two saddle-node bifurcation points for each value of the control S_0, we have separated the corresponding values of I_0 by a comma.

Meta-parameters file (`meta_parameters.yaml`)

In this YAML file, we specify:

- name of the model (e.g., `autocatalytic`)
- settings for the initial value problem (`t_end`)
- type of measured bifurcation point (`saddle-node` or `hopf`)
- method for two-parameter continuation (`pseudo-arclength` or `deflated`)
- model compartments with their names and initial conditions
- which model compartment to use for plotting
- model parameters and external controls.

The model parameters and the external controls are all listed as parameters in the `meta_parameters.yaml` file. For each of them, we need to specify a name, a type, and a default value. The available types for the parameters are:

- `fixed`: parameter is known and fixed to the default value
- global: parameter is unknown and needs to be estimated
- `control_1`: parameter is an external control treated as our primary control
- `control_2`: parameter is an external control treated as our secondary control.

Since the primary control is used as the bifurcation parameter for computing the bifurcation diagrams, it is important to also specify the range of values for this control and the initial step size for numerical continuation. Code 9.1 and Code 9.2 show an example of this file for the autocatalytic reaction model.

Model definition (`model_equations.py`)

In order to define the model equations in the `bifit` package, we need to create a child class called `Model` of the abstract `BaseModel` class from the `models` module. The `Model` class then needs to contain the right-hand side function of the ODE system in the `rhs_` method, and its Jacobian matrix in the `jacobian_` method. When initializing the `Model` class, we also load the model specifications and meta parameters from the `meta_parameters.yaml` file.

For the autocatalytic reaction model, the model equations are defined in the `model_equations.py` file as shown in Code 9.3. In particular, for every model, the contents of the `__init__.py` remain the same and the user only needs to change the contents of the `rhs_` and `jacobian_` methods.

```yaml
# model parameters and controls

parameter_1:
    name: 'k0'
    type: 'global'
    default_value: 0.003

parameter_2:
    name: 'k1'
    type: 'global'
    default_value: 0.25

parameter_3:
    name: 'k2'
    type: 'fixed'
    default_value: 300

parameter_4:
    name: 'k3'
    type: 'fixed'
    default_value: 0.0035

parameter_5:
    name: 'k4'
    type: 'fixed'
    default_value: 0.00007

parameter_6:
    name: 'I0'
    type: 'control_2'
    default_value: 0.004

parameter_7:
    name: 'S0'
    type: 'control_1'
    default_value: 0.05
    min_value: 0.001
    max_value: 0.5
    step_size: 0.001
```

Code 9.1: Example of `meta_parameters.yaml` for the autocatalytic network model.

```
model_name: 'autocatalytic'

t_end: 1000

bifurcation:
    type: 'saddle-node'

two_parameter_continuation_method: 'deflated'

# model compartments - order should match that of model equations

compartment_1:
    name: 'A'
    value: 0

compartment_2:
    name: 'I'
    value: 0

compartment_3:
    name: 'S'
    value: 0

to_plot: 'A'
```

Code 9.2: Example of `meta_parameters.yaml` for the autocatalytic network model (cont'd.).

```python
import os
import autograd.numpy as np
from bifit.parser.yaml_parser import YamlParser
from bifit.models.utils import nparray_to_dict
from bifit.models.base_model import BaseModel

class Model(BaseModel):

    def __init__(self):
        super().__init__()
        file_path = os.path.dirname(__file__)
        parser = YamlParser(file_path=file_path)
        self.specifications = parser.get_problem_specifications()

    def rhs_(self, x: np.ndarray) -> np.ndarray:
        c, p, _ = nparray_to_dict(x=x, model=self)
        A, I, S = c["A"], c["I"], c["S"]
        k1, k2, k3, k4 = p["k1"], p["k2"], p["k3"], p["k4"]
        k0, I0, S0 = p["k0"], p["I0"], p["S0"]

        model_equations = {
            "A": k1*S*A - k2*I*A - k3*A - k0*A + k4*S,
            "I": k0*I0 - k0*I - k2*I*A,
            "S": k0*S0 - k0*S - k4*S - k1*S*A,
        }

        M_list = [model_equations[key] for key in self.compartments]
        return np.array(M_list)

    def jacobian_(self, x: np.ndarray) -> np.ndarray:
        c, p, _ = nparray_to_dict(x=x, model=self)
        A, I, S = c["A"], c["I"], c["S"]
        k1, k2, k3, k4 = p["k1"], p["k2"], p["k3"], p["k4"]
        k0, I0, S0 = p["k0"], p["I0"], p["S0"]

        model_jacobian = {
            "A": np.array([k1*S - k2*I - k3 - k0,
                           -k2*A,
                           k1*A + k4]),
            "I": np.array([-k2*I,
                           -k0 - k2*A,
                           0]),
            "S": np.array([-k1*S,
                           0,
                           -k0 - k4 - k1*A]),
        }

        J_list = [model_jacobian[key] for key in self.compartments]
        return np.row_stack(J_list)
```

Code 9.3: Example of `model_equations.py` for the autocatalytic network model.

Execution script (`main.py`)

Finally, the `main.py` script implements the complete run of the parameter estimation workflow: loading the model, loading and preprocessing the data, generating initial guesses, solving the parameter estimation problem and computing the confidence intervals. Code 9.4 shows an example of this file for the autocatalytic reaction model.

```
import sys
import os

file_path = os.path.dirname(__file__)
sys.path.append(os.path.abspath(os.path.join(file_path, "../..")))

from model_equations import Model
from bifit.preprocessing.preprocess_data import DataPreprocessor
from bifit.estimation.initial_guess import InitialGuessGenerator
from bifit.estimation.parameter_estimator import ParameterEstimator

# Load the model and parameters
model = Model()
model.set_parameters()

# Preprocess the data and compute measurement errors
preprocessor = DataPreprocessor()
preprocessor.load_the_data(file_path=file_path, error_scale=0.05)
model.data = preprocessor.data
model.data_weights = preprocessor.weights

# Generate initial guesses for parameter estimation
initializer = InitialGuessGenerator()
initializer.generate_initial_guess(model=model)

# Solve parameter estimation problem
fit = ParameterEstimator(
    x0=initializer.initial_guesses,
    model=model,
    method="gauss-newton",
    plot_iters=True,
    compute_ci=True,
)
```

Code 9.4: Example of the `main.py` file to run the parameter estimation problem.

10 Conclusions and Outlook

In this thesis, we have developed a new approach to parameter estimation for dynamical systems — one that leverages measurements of external controls at bifurcation points instead of relying on traditional time-series data. This approach was inspired by experimental systems in fields such as chemical engineering, neuroscience, ecology, and biochemistry, where applied external controls are used to control and monitor changes in the qualitative dynamics of the system. Focusing on qualitative dynamics rather than quantitative observations can be particularly useful for systems exhibiting bistability or oscillations. It is also well-suited to experimental systems where obtaining measurements at multiple time points can be challenging or even infeasible.

To estimate the unknown parameters of a mathematical model, we formulated a constrained nonlinear least-squares problem that aligns measurements of external controls at bifurcation points with their corresponding theoretical predictions from the mathematical model. The optimization problem is then solved using the generalized Gauss-Newton method with efficient structure-exploitation and the uncertainty of the parameter estimates is quantified through *a posteriori* sensitivity analysis. Since the optimization variables include not only the parameters to be estimated but also the values of the applied controls at bifurcation points and their corresponding steady states, we developed a robust numerical strategy to generate suitable initial guesses that satisfy these requirements. This process involves computing bifurcation diagrams using numerical continuation methods, such as deflated continuation and pseudo-arclength continuation. Notably, we used these continuation methods with an adaptive step size to ensure that the initial guesses theoretically predict bifurcation points close to the measured bifurcation points.

Furthermore, we implemented and published our parameter estimation framework as an open-source software, enabling researchers to easily calibrate their own mathematical models using bifurcation point measurements. Using our software, we demonstrated the effectiveness of our approach through four case studies spanning diverse fields of application. Moreover, we also highlighted a variety of experimental studies involving bifurcation points where our method is particularly well-suited.

Finally, we extended the standard optimal experimental design problem, which aims to minimize the uncertainty of parameter estimates, to our bifurcation-based parameter estimation framework. This adaptation may enable researchers to design additional experiments to further refine their parameter estimates.

Future directions and outlook

There are several compelling directions for future research on this topic. An interesting question to investigate might be to find out how parameter estimation using bifurcation points compares to traditional time-series data in terms of accuracy and reliability. A quantitative comparison between these approaches would offer valuable insights into the scenarios where bifurcation-based calibration is most advantageous.

Another promising direction is the extension of our framework to more complex dynamical systems, such as partial differential equations (PDEs) and delay differential equations (DDEs). Many real-world systems involve spatial or time-delayed interactions, and adapting our methods to these settings would significantly expand their applicability.

Currently, our software implementation does not include the optimal experimental design module or the capability to estimate parameters using measurements from more than two external controls. Adding these features into the software would expand the practical usability of our approach and make it more adaptable to diverse experimental setups.

To conclude, our work offers a fresh perspective on the types of data that can be used to calibrate mathematical models. With our open-source implementation of the numerical methods from this thesis, researchers from various fields can now easily apply, adapt, and extend our methods to their own problems, fostering further research in this area.

Bibliography

[AHF90] B. D. Aguda, L. L. Hofmann Frisch, and L. Folke Olsen. “Experimental evidence for the coexistence of oscillatory and steady states in the peroxidase-oxidase reaction”. In: *Journal of the American Chemical Society* 112.18 (1990), pp. 6652–6656 (cit. on pp. 3, 4, 107).

[AL90] B. D. Aguda and R. Larter. “Sustained oscillations and bistability in a detailed mechanism of the peroxidase-oxidase reaction”. In: *Journal of the American Chemical Society* 112.6 (1990), pp. 2167–2174 (cit. on p. 108).

[AG03] E. L. Allgower and K. Georg. *Introduction to Numerical Continuation Methods*. Society for Industrial and Applied Mathematics, 2003 (cit. on p. 21).

[Arn92] V. I. Arnold. *Ordinary Differential Equations*. 1st ed. Springer Berlin, Heidelberg, 1992 (cit. on p. 15).

[Ash+09] M. Ashyraliyev, Y. Fomekong-Nanfack, J. A. Kaandorp, and J. G. Blom. “Systems biology: parameter estimation for biochemical models”. In: *The FEBS Journal* 276.4 (2009), pp. 886–902 (cit. on p. 81).

[Aud+01] S. Audoly, G. Bellu, L. D’Angio, M.P. Saccomani, and C. Cobelli. “Global identifiability of nonlinear models of biological systems”. In: *IEEE Transactions on Biomedical Engineering* 48.1 (2001), pp. 55–65 (cit. on p. 82).

[BB08] J. R. Banga and E. Balsa-Canto. “Parameter estimation and optimal experimental design”. In: *Essays in biochemistry* 45 (2008), pp. 195–209 (cit. on p. 91).

[Bec+05] L. Becks, F. M. Hilker, H. Malchow, K. Jürgens, and H. Arndt. “Experimental demonstration of chaos in a microbial food web”. In: *Nature* 435 (2005), pp. 1226–1229 (cit. on p. 110).

[Bel59] B. P. Belousov. “A Periodic Reaction and Its Mechanism”. In: *Collection of Short Papers on Radiation Medicine for 1958* (1959) (cit. on p. 59).

[Bin+06] S. Binczak, S. Jacquir, J. M. Bilbault, V. B. Kazantsev, and V. I. Nekorkin. “Experimental study of electrical FitzHugh–Nagumo neurons with modified excitability”. In: *Neural Networks* 19.5 (2006), pp. 684–693 (cit. on pp. 3, 4, 110, 111).

[Boc87] H. G. Bock. “Randwertproblemmethoden zur Parameteridentifizierung in Systemen nichtlinearer Differentialgleichungen”. In: *Bonner Mathematische Schriften, Universität Bonn* 183 (1987) (cit. on pp. 4, 6, 35, 40–43).

[BKK07] H. G. Bock, E. Kostina, and O. Kostyukova. “Covariance Matrices for Parameter Estimates of Constrained Parameter Estimation Problems”. In: *SIAM Journal on Matrix Analysis and Applications* 29.2 (2007), pp. 626–642 (cit. on pp. 81, 86, 87, 90).

[BKS00] H. G. Bock, E. Kostina, and J. P. Schlöder. “On the Role of Natural Level Functions to Achieve Global Convergence for Damped Newton Methods”. In: *System Modelling and Optimization*. Ed. by M. J. D. Powell and S. Scholtes. Boston, MA: Springer US, 2000, pp. 51–74 (cit. on pp. 44, 45).

[BFR23] N. Boullé, P. E. Farrell, and M. E. Rognes. “Optimization of Hopf Bifurcation Points”. In: *SIAM Journal on Scientific Computing* 45.3 (2023), B390–B411 (cit. on p. 61).

[BRI19] N. Braniff, A. Richards, and B. Ingalls. “Optimal Experimental Design for a Bistable Gene Regulatory Network”. In: *IFAC-PapersOnLine* 52.26 (2019), pp. 255–261 (cit. on p. 91).

[BSO01] T. V. Bronnikova, W. M. Schaffer, and L. F. Olsen. “Nonlinear Dynamics of the Peroxidase-Oxidase Reaction: I. Bistability and Bursting Oscillations at Low Enzyme Concentrations”. In: *The Journal of Physical Chemistry B* 105.1 (2001), pp. 310–321 (cit. on p. 108).

[BG71] K. M. Brow and W. B. Gearhart. “Deflation techniques for the calculation of further solutions of a nonlinear system”. In: *Numerische Mathematik* 16 (1971), pp. 334–342 (cit. on p. 29).

[Can98] C. A. Canizares. “Calculating optimal system parameters to maximize the distance to saddle-node bifurcations”. In: *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications* 45.3 (1998), pp. 225–237 (cit. on p. 61).

[Car82] J. Carr. *Applications of Centre Manifold Theory*. 1st ed. Applied Mathematical Sciences. Springer New York, NY, 1982 (cit. on p. 16).

[CK05] G. Cedersund and C. Knudsen. “Improved parameter estimation for systems with an experimentally located Hopf bifurcation”. In: *Systems Biology* 152.3 (2005), pp. 161–168 (cit. on p. 61).

[Cha21] N. S. Chandel. “Glycolysis”. In: *Cold Spring Harbor Perspectives in Biology* 13.5 (2021), a040535 (cit. on p. 106).

[Chi+16] O. T. Chis, A. F. Villaverde, J. R. Banga, and E. Balsa-Canto. “On the relationship between sloppiness and identifiability”. In: *Mathematical Biosciences* 282 (2016), pp. 147–161 (cit. on p. 81).

[CD80] C. Cobelli and J. J. Distefano 3rd. “Parameter and structural identifiability concepts and ambiguities: a critical review and analysis”. In: *The American Journal of Physiology* 239.1 (1980), R7–R24 (cit. on p. 81).

[CT84] F. Conrad and V. Treguera-Seguda. “Parameter estimation in some diffusion and reaction models: An application of bifurcation theory”. In: *Chemical Engineering Science* 39.4 (1984), pp. 705–711 (cit. on p. 3).

[Cos+97] R. F. Costantino, R. A. Desharnais, J. M. Cushing, and B. Dennis. “Chaotic Dynamics in an Insect Population”. In: *Science* 275.5298 (1997), pp. 389–391 (cit. on pp. 3, 4, 110).

[DSH99] S. Danø, P. G. Sørensen, and F. Hynne. “Sustained oscillations in living cells”. In: *Nature* 402.6759 (1999), pp. 320–322 (cit. on pp. 3, 4, 106, 107).

[DOP79] H. Degn, L. F. Olsen, and J. W. Perram. “Bistability, oscillation, and chaos in an enzyme reaction”. In: *Annals of the New York Academy of Sciences* 316.1 (1979), pp. 623–637 (cit. on pp. 5, 120).

[Den+97] B. Dennis, R. A. Desharnais, J. M. Cushing, and R. F. Costantino. “Transitions in Population Dynamics: Equilibria to Periodic Cycles to Aperiodic Cycles”. In: *Journal of Animal Ecology* 66.5 (1997), pp. 704–729 (cit. on pp. 3, 4, 110).

[Eln+06] S. S. E. H. Elnashaie, Z. Chen, P. Garhyan, P. Prasad, and A. Mahecha-Botero. “Practical Implications of Bifurcation and Chaos in Chemical and Biological Reaction Engineering”. In: *International Journal of Chemical Reactor Engineering* 4.1 (2006) (cit. on pp. 3, 5, 111, 112).

[EDH14] F. Emmert-Streib, M. Dehmer, and B. Haibe-Kains. “Gene regulatory networks and their applications: understanding biological and medical problems in terms of networks”. In: *Frontiers in cell and developmental biology* 2 (2014), p. 38 (cit. on p. 109).

[FBB16] P. E. Farrell, C. H. L. Beentjes, and A. Birkisson. *The computation of disconnected bifurcation diagrams*. 2016. arXiv: `1603.00809 [math.NA]` (cit. on pp. 4, 5, 29–31, 33).

[Fit61] R. FitzHugh. “Impulses and Physiological States in Theoretical Models of Nerve Membrane”. In: *Biophysical Journal* 1.6 (1961), pp. 445–466 (cit. on p. 110).

[Fle00] R. Fletcher. *Practical Methods of Optimization*. John Wiley & Sons, Ltd, 2000 (cit. on p. 35).

[FM08] G. Franceschini and S. Macchietto. “Model-based design of experiments for parameter precision: State of the art”. In: *Chemical Engineering Science* 63.19 (2008), pp. 4846–4872 (cit. on p. 91).

[Fus+05] G. F. Fussmann, S. P. Ellner, N. G. Hairston Jr, et al. “Ecological and Evolutionary Dynamics of Experimental Plankton Communities”. In: *Advances in Ecological Research* 37 (2005), pp. 221–243 (cit. on p. 110).

[Fus+00] G. F. Fussmann, S. P. Ellner, K. W. Shertzer, and N. G. Hairston Jr. “Crossing the Hopf Bifurcation in a Live Predator-Prey System”. In: *Science* 290.5495 (2000), pp. 1358–1360 (cit. on pp. 3–5, 110, 111, 120, 122, 123).

[GB81] W. Geiseler and K. Bar-Eli. “Bistability of the oxidation of cerous ions by bromate in a stirred flow reactor”. In: *The Journal of Physical Chemistry* 85.7 (1981), pp. 908–914 (cit. on p. 3).

[GMW19] P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2019 (cit. on p. 35).

[Gov+19] W. Govaerts, Y. A. Kuznetsov, H. G. E. Meijer, et al. *MATCONT: Continuation toolbox for ODEs in Matlab*. Manual for MatCont 7.1. Aug. 2019 (cit. on pp. 26, 113).

[GL87] W. R. C. Graham and D. T. Lynch. “CO oxidation on Pt: Model discrimination using experimental bifurcation behavior”. In: *American Institute of Chemical Engineers Journals* 33.5 (1987), pp. 792–800 (cit. on pp. 3, 5, 111).

[Gre71] H. Greenberg. *Integer Programming*. 1st ed. Vol. 76. Academic Press, 1971 (cit. on p. 97).

[GR83] A. Griewank and G. Reddien. "The Calculation of Hopf Points by a Direct Method". In: *IMA Journal of Numerical Analysis* 3.3 (1983), pp. 295–303 (cit. on p. 67).

[GW08] A. Griewank and A. Walther. *Evaluating Derivatives*. 2nd. Society for Industrial and Applied Mathematics, 2008 (cit. on p. 56).

[Gro59] D. M. Grobman. "Homeomorphism of systems of differential equations". In: *Doklady Akademii Nauk SSSR* 128.5 (1959), pp. 880–881 (cit. on p. 15).

[GH13] J. Guckenheimer and P. Holmes. *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*. Applied Mathematical Sciences. Springer New York, NY, 2013 (cit. on pp. 11, 15).

[Gus+14] A. K. Gustavsson, D. D. van Niekerk, C. B. Adiels, et al. "Allosteric regulation of phosphofructokinase controls the emergence of glycolytic oscillations in isolated yeast cells". In: *The FEBS Journal* 281.12 (2014), pp. 2784–2793 (cit. on pp. 106, 107).

[HL87] M. P. Harold and D. Luss. "Use of bifurcation map for kinetic parameter estimation. 1. Ethane oxidation". In: *Industrial & Engineering Chemistry Research* 26.10 (1987), pp. 2092–2098 (cit. on pp. 3, 5, 60, 111).

[Har+20] C. R. Harris, K. J. Millman, S. J. van der Walt, et al. "Array programming with NumPy". In: *Nature* 585.7825 (Sept. 2020), pp. 357–362 (cit. on p. 128).

[Har60] P. Hartman. "A lemma in the theory of structural stability of differential equations". In: *Proceedings of the American Mathematical Society* 11 (1960), pp. 610–620 (cit. on p. 15).

[Har63] P. Hartman. "On the Local Linearization of Differential Equations". In: *Proceedings of the American Mathematical Society* 14.4 (1963), pp. 568–573 (cit. on p. 15).

[HR84] J. L. Hindmarsh and R. M. Rose. "A model of neuronal bursting using three coupled first order differential equations". In: *Proceedings of the Royal Society of London. Series B, Biological Sciences* 221.1222 (1984), pp. 87–102 (cit. on p. 110).

[HH52] A. L. Hodgkin and A. F. Huxley. "A quantitative description of membrane current and its application to conduction and excitation in nerve". In: *The Journal of physiology* 117.4 (1952), pp. 500–544 (cit. on p. 110).

[AHA07] N. M. Al-Hosiny, I. D. Henning, and M. J. Adams. "Tailoring enhanced chaos in optically injected semiconductor lasers". In: *Optics Communications* 269.1 (2007), pp. 166–173 (cit. on p. 105).

[Hun07] J. D. Hunter. "Matplotlib: A 2D graphics environment". In: *Computing in Science & Engineering* 9.3 (2007), pp. 90–95 (cit. on p. 128).

[HDG01] F. Hynne, S. Danø, and Sørensen P. G. “Full-scale model of glycolysis in Saccharomyces cerevisiae”. In: *Biophysical Chemistry* 94.1 (2001), pp. 121–163 (cit. on p. 107).

[JM61] F. Jacob and J. Monod. “Genetic regulatory mechanisms in the synthesis of proteins”. In: *Journal of Molecular Biology* 3.3 (1961), pp. 318–356 (cit. on p. 109).

[Jan15] D. Janka. “Sequential quadratic programming with indefinite Hessian approximations for nonlinear optimum experimental design for parameter estimation in differential–algebraic equations”. PhD thesis. Ruprecht-Karls-Universität Heidelberg, 2015 (cit. on pp. 42, 91, 102).

[Jos+73] J. L. Jost, J. F. Drake, A. G. Fredrickson, and H. M. Tsuchiya. “Interactions of Tetrahymena pyriformis, Escherichia coli, Azotobacter vinelandii, and glucose in a minimal medium”. In: *Journal of Bacteriology* 113.2 (1973), pp. 834–840 (cit. on pp. 3, 4, 110).

[Kel86] H. B. Keller. *Lectures on Numerical Methods in Bifurcation Problems*. 1986 (cit. on pp. 4, 5, 22).

[Kör02] S. Körkel. “Numerische Methoden für optimale Versuchsplanungsprobleme bei nichtlinearen DAE-Modellen”. PhD thesis. Ruprecht-Karls-Universität Heidelberg, 2002 (cit. on pp. 91, 95, 97, 98).

[Kra06] B. Krauskopf. *Semiconductor lasers as dynamical systems*. English. Working Paper. Sponsorship: With support from an Engineering and Physical Sciences Research Council (EPSRC) Advanced Research Fellowship grant. Mar. 2006 (cit. on p. 105).

[Kuz23] Y. A. Kuznetsov. *Elements of Applied Bifurcation Theory*. 4th. Vol. 112. Applied Mathematical Sciences. Springer Cham, 2023 (cit. on pp. 11, 13, 17, 18, 20, 22, 26).

[Lei21] J. Lei. “Mathematical Models for Gene Regulatory Network Dynamics”. In: *Systems Biology: Modeling, Analysis, and Simulation*. Cham: Springer International Publishing, 2021, pp. 145–198 (cit. on p. 109).

[Let+16] B. Letham, P. A. Letham, C. Rudin, and E. P. Browne. “Prediction uncertainty and optimal experimental design for learning dynamical systems”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 26 (2016) (cit. on p. 91).

[LPS10] D. Linaro, T. Poggi, and M. Storace. “Experimental bifurcation diagram of a circuit-implemented neuron model”. In: *Physics Letters A* 374.45 (2010), pp. 4589–4593 (cit. on pp. 3, 4, 110, 111).

[Lya92] A. M. Lyapunov. “The General Problem of the Stability of Motion”. In: *Kharkov Mathematical Society, Kharkov* (1892) (cit. on p. 14).

[Mac+15] M. C. Mackey, M. Santillán, M. Tyran-Kamińska, and E. S. Zeron. “The utility of simple mathematical models in understanding gene regulatory dynamics”. In: *In Silico Biology* 12.1-2 (2015), pp. 23–53 (cit. on p. 109).

[MDA15] D. Maclaurin, D. Duvenaud, and R. P. Adams. “Autograd: Effortless gradients in numpy”. In: *ICML 2015 AutoML Workshop*. Vol. 238. 2015, p. 5 (cit. on p. 128).

[MB03] P. A. Mayes and D. A. Bender. “Glycolysis & the oxidation of pyruvate”. In: *Harper’s Illustrated Biochemistry, 26th Edition*. Ed. by R. K. Murray, D. K. Granner, P. A. Mayes, and V. W. Rodwell. 2003, pp. 136–144 (cit. on p. 106).

[MBC79] M. D. McKay, R. J. Beckman, and W. J. Conover. “A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code”. In: *Technometrics* 21.2 (1979), pp. 239–245 (cit. on p. 114).

[McK10] W. McKinney. “Data Structures for Statistical Computing in Python”. In: *Proceedings of the 9th Python in Science Conference. SciPy 2010*. Python in Science Conference (Austin, Texas). Ed. by Stéfan van der Walt and Jarrod Millman. Proceedings of the Python in Science Conference. SciPy, 2010, pp. 56–61 (cit. on p. 128).

[Mil+89] J. G. Milton, A. Longtin, A. Beuter, M. C. Mackey, and L. Glass. “Complex dynamics and bifurcations in neurology”. In: *Journal of Theoretical Biology* 138.2 (1989), pp. 129–147 (cit. on p. 110).

[MM02] M. Mönnigmann and W. Marquardt. “Normal Vectors on Manifolds of Critical Points for Parametric Robustness of Equilibrium Solutions of ODE Systems”. In: *Journal of Nonlinear Science* 12 (2002), pp. 85–112 (cit. on p. 67).

[ML81] C. Morris and H. Lecar. “Voltage oscillations in the barnacle giant muscle fiber”. In: *Biophysical Journal* 35.1 (1981), pp. 193–213 (cit. on p. 110).

[NAY62] J. Nagumo, S. Arimoto, and S. Yoshizawa. “An Active Pulse Transmission Line Simulating Nerve Axon”. In: *Proceedings of the IRE* 50.10 (1962), pp. 2061–2070 (cit. on p. 110).

[NP08] A. Narang and S. S. Pilyugin. “Bistability of the lac Operon During Growth of Escherichia coli on Lactose and Lactose + Glucose”. In: *Bulletin of Mathematical Biology* 70 (2008), pp. 1032–1064 (cit. on p. 109).

[Nat14] M. Nattermann. “Numerical Methods of Optimum Experimental Design Based on a Second-Order Approximation of Confidence Regions”. PhD thesis. Philipps-Universität Marburg, 2014 (cit. on pp. 46, 89).

[NW06] J. Nocedal and S. J. Wright. *Numerical Optimization*. 2nd. Springer Series in Operations Research and Financial Engineering. Springer New York, NY, 2006 (cit. on pp. 35, 37–41, 43, 44).

[NFT71] R. M. Noyes, R. J. Field, and R. C. Thompson. “Mechanism of reaction of bromine(V) with weak one-electron reducing agents”. In: *Journal of the American Chemical Society* 93.26 (1971), pp. 7315–7316 (cit. on p. 59).

[OD77] L. F. Olsen and H. Degn. “Chaos in an enzyme reaction”. In: *Nature* 267 (1977), pp. 177–178 (cit. on p. 120).

[OYS14] I. Otero-Muras, P. Yordanov, and J. Stelling. “A method for inverse bifurcation of biochemical switches: inferring parameters from dose response curves”. In: *BMC Systems Biology* 8.114 (2014) (cit. on p. 61).

[Ozb+04] E. M. Ozbudak, M. Thattai, H. N. Lim, B. I. Shraiman, and A. van Oudenaarden. “Multistability in the lactose utilization network of Escherichia coli”. In: *Nature* 427 (2004), pp. 737–740 (cit. on pp. 3, 4, 109).

[Pat+99] G. Patel, G. Cymbalyuk, R. Calabrese, and S. DeWeerth. “Bifurcation Analysis of a Silicon Neuron”. In: *Advances in Neural Information Processing Systems*. Ed. by S. Solla, T. Leen, and K. Müller. Vol. 12. MIT Press, 1999 (cit. on pp. 3, 4, 110, 111).

[Poh78] H. Pohjanpalo. “System identifiability based on the power series expansion of the solution”. In: *Mathematical Biosciences* 41.1-2 (1978), pp. 21–33 (cit. on p. 81).

[Ric+94] P. Richard, J. A. Diderich, B. M. Bakker, et al. “Yeast cells with a specific cellular make-up and an environment that removes acetaldehyde are prone to sustained glycolytic oscillations”. In: *FEBS Letters* 341 (1994), pp. 223–226 (cit. on pp. 3, 4, 107).

[Rut+20] G. H. Rutherford, Z. D. Mobille, J. Brandt-Trainer, R. Follmann, and E. Rosa. “Analog implementation of a Hodgkin–Huxley model neuron”. In: *American Journal of Physics* 88 (2020), pp. 918–923 (cit. on p. 111).

[SMZ07] M. Santillán, M. C. Mackey, and E. S. Zeron. “Origin of bistability in the lac operon”. In: *Biophysical Journal* 92.11 (2007), pp. 3830–3842 (cit. on p. 109).

[SB83] J. Schlöder and H. G. Bock. “Identification of Rate Constants in Bistable Chemical Reactions”. In: *Numerical Treatment of Inverse Problems in Differential and Integral Equations: Proceedings of an International Workshop, Heidelberg, Fed. Rep. of Germany, August 30–September 3, 1982*. Ed. by P. Deuflhard and E. Hairer. Boston, MA: Birkhäuser Boston, 1983, pp. 27–47 (cit. on pp. 2–4, 6, 46, 59, 70).

[Sch87] J. P. Schlöder. “Numerische Methoden zur Behandlung hochdimensionaler Aufgaben der Parameteridentifizierung”. In: *Bonner Mathematische Schriften, Universität Bonn* 187 (1987) (cit. on pp. 2–4, 6, 35, 46, 59, 70, 81, 89).

[Sch19] I. Schrot. *Comparative Analysis of the Restrictive Monotonicity Test and the Backward Step Control for Newton’s Method*. Ruprecht-Karls-University Heidelberg, 2019 (cit. on pp. 44, 45).

[Sem+16] S. N. Semenov, L. J. Kraft, A. Ainla, et al. “Autocatalytic, bistable, oscillatory networks of biologically relevant organic reactions”. In: *Nature* 537 (2016), pp. 656–660 (cit. on pp. 3–5, 108, 109, 114, 117, 118, 133).

[Sem+15] S. N. Semenov, A. S. Y. Wong, R. M. van der Made, et al. “Rational design of functional and tunable oscillating enzymatic networks”. In: *Nature Chemistry* 7 (2015), pp. 160–165 (cit. on p. 108).

[Sen05] A. Sensse. “Convex and toric geometry to analyze complex dynamics in chemical reaction systems”. PhD thesis. Otto-von-Guericke-Universität, Magdeburg, 2005 (cit. on p. 107).

[SB87] B. H. Shanks and J. E. Bailey. “Experimental investigations using feedback-induced bifurcation: carbon-monoxide oxidation over supported silver”. In: *Chemical Engineering Communications* 61.1-6 (1987), pp. 127–149 (cit. on pp. 3, 5, 60, 111).

[Sim03] T. B. Simpson. “Mapping the nonlinear dynamics of a distributed feedback semiconductor laser subject to external optical injection”. In: *Optics Communications* 215.1-3 (2003), pp. 135–151 (cit. on pp. 114, 115).

[Som17] A. Sommer. “Numerical methods for parameter estimation in dynamical systems with noise with applications in systems biology”. PhD thesis. Ruprecht-Karls-Universität Heidelberg, 2017 (cit. on pp. 81, 83).

[SLA88] C. G Steinmetz, R. Larter, and B. D. Aguda. “Modelling a Biochemical Oscillator”. In: *Proceedings of the Indiana Academy of Science*. Vol. 98. 1988, pp. 157–168 (cit. on pp. 108, 120).

[Ste+20] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd. “OSQP: an operator splitting solver for quadratic programs”. In: *Mathematical Programming Computation* 12.4 (2020), pp. 637–672 (cit. on pp. 6, 128, 133).

[TB05] T. Tian and K. Burrage. “A Mathematical Model for Genetic Regulation of the Lactose Operon”. In: *Computational Science and Its Applications – ICCSA 2005*. Ed. by O. Gervasi, M. L. Gavrilova, V. Kumar, et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 1245–1253 (cit. on p. 109).

[VGR89] S. Vajda, K. R. Godfrey, and H. Rabitz. “Similarity transformation approach to identifiability analysis of nonlinear compartmental models”. In: *Mathematical Biosciences* 93.2 (1989), pp. 217–248 (cit. on p. 81).

[Van89] A. Vanderbauwhede. “Centre Manifolds, Normal Forms and Elementary Bifurcations”. In: *Dynamics Reported: A Series in Dynamical Systems and Their Applications*. Ed. by U. Kirchgraber and H. O. Walther. Wiesbaden: Vieweg+Teubner Verlag, 1989, pp. 89–169 (cit. on p. 16).

[Vir+20] P. Virtanen, R. Gommers, T. E. Oliphant, et al. “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python”. In: *Nature Methods* 17 (2020), pp. 261–272 (cit. on pp. 5, 128, 133).

[Wal11] S. F. Walter. “Structured higher-order algorithmic differentiation in the forward and reverse mode with application in optimum experimental design”. PhD thesis. Humboldt-Universität zu Berlin, 2011 (cit. on p. 96).

[WKL99] S. Wieczorek, B. Krauskopf, and D. Lenstra. “A unifying view of bifurcations in a semiconductor laser subject to optical injection”. In: *Optics Communications* 172.1-6 (1999), pp. 279–295 (cit. on pp. 5, 115).

[Wie+05] S. Wieczorek, B. Krauskopf, T. B. Simpson, and D. Lenstra. “The dynamical complexity of optically injected semiconductor lasers”. In: *Physics Reports* 416.1-2 (2005), pp. 1–128 (cit. on p. 105).

[Wie+03] S. Wieczorek, T. B. Simpson, B. Krauskopf, and D. Lenstra. “Bifurcation transitions in an optically injected diode laser: theory and experiment”. In: *Optics Communications* 215.1-3 (2003), pp. 125–134 (cit. on pp. 3, 5, 105, 106, 114, 115).

[Wig03] S. Wiggins. *Introduction to Applied Nonlinear Dynamical Systems and Chaos*. 2nd. Texts in Applied Mathematics. Springer New York, NY, 2003 (cit. on pp. 11, 105).

[Yos+03] T. Yoshida, L. E. Jones, S. P. Ellner, G. F. Fussmann, and N. G. Hairston Jr. “Rapid evolution drives ecological dynamics in a predator–prey system”. In: *Nature* 424 (2003), pp. 303–306 (cit. on p. 110).

[ZBH01] A. Q. M. Zamamiri, G. Birol, and M. A. Hjortsø. “Multiple stable states and hysteresis in continuous, oscillating cultures of budding yeast”. In: *Biotechnology and Bioengineering* 75.3 (2001), pp. 305–312 (cit. on pp. 3, 5, 111).

[Zha64] A. M. Zhabotinsky. “Periodical oxidation of malonic acid in solution (a study of the Belousov reaction kinetics)”. In: *Biofizika* 9 (1964), pp. 306–311 (cit. on p. 59).

List of Figures