Computer-aided proofs and algorithms in analysis

Dissertation for the degree of Philosophiae Doctor (PhD)

Ferenc A. Bartha

Department of Mathematics
University of Bergen
Norway

2013

Acknowledgements

First of all I would like to thank my main supervisor Warwick Tucker for always being encouraging and positive with respect to my work and ideas. He gave me the freedom to work on what I wanted, and at the same time guided me in the right directions. I have met not just an excellent scientist but a friend in him. I would like to thank my supervisor Hans Munthe-Kaas who supported me here in Bergen, and at last but not at least, by inviting me to the MaGIC workshops, he got me on cross-country skis for the first time in my life.

I would like to thank the other (former and current) members of the CAPA group in Uppsala and Bergen who have always been helpful to me. I would also like to thank my other co-authors Tibor Krisztin and Ábel Garab for the excellent work and cooperation, Piotr Zgliczyński for his help not just in my projects but in other matters.

During my doctoral period I had the opportunity to visit the Department of Mathematics at the University of Uppsala numerous times, I thank the former and current heads of the department in Bergen for making these leaves possible.

I would like to thank Erik, Alexander, Morten, Huiyan, Hilde Kristine, Henning, Carina, Christian and Dagfinn; and all the students, employees and staff of the department for making my stay here pleasant.

Finally, I thank my family, my girlfriend, and my friends both in Bergen and at home, for their love, support, and for being there when I needed.

This dissertation is submitted as a partial fulfillment of the requirements for the Degree of Philosphy (PhD) at the Faculty of Mathematics and Natural Sciences, University of Bergen, Norway. In the preparation of this thesis I used the LaTeX template by Birkeland and Nepstad.

Financial support of my research has been granted by The Bergen Research Foundation (Bergens forskningsstiftelse). Project title: “Computer-Aided Proofs in Mathematical Analysis.” Project number: 801458; and by The Swedish Research Council (Vetenskapsrådet) award 2008-7510 for CAPA - Computer-aided proofs in analysis.

List of papers

A. Ferenc A. Bartha and Hans Z. Munthe-Kaas,
Computing of B-series by Automatic Differentiation

accepted for publication in Discrete and Continuous Dynamical Systems A – Special issue for the 65'th birthday of Arieh Iserles © : 2013 Published by AIMS

B. Ferenc A. Bartha, Ábel Garab and Tibor Krisztin,
Local stability implies global stability for the 2-dimensional Ricker map

second revision (minor) submitted to Journal of Difference Equations and Applications

C. Ferenc A. Bartha and Ábel Garab,
Necessary and sufficient condition for the global stability of a delayed discrete-time single neuron model

before submission

D. Ferenc A. Bartha and Warwick Tucker,
Fixed point of a destabilized Kuramoto-Sivashinsky equation

manuscript

List of papers that are not included in this thesis

E. Ferenc Bogár, Ferenc Bartha, Ferenc A. Bartha and Norman H. March,
Pauli potential from Heilmann-Lieb electron density obtained by summing hydrogenic closed-shell densities over the entire bound-state spectrum

published in Phys. Rev. A 83, 014502 (2011)

Abstract

The computational power has increased dramatically since the appearance of the first computers, making them a vital tool in the analysis of dynamical systems. We present further applications of those two basic ideas, namely interval arithmetic and automatic differentiation, that address the question of the reliability of the results and the difficulty of calculating derivatives.

In general, the result of a numerical calculation will be influenced by errors, since the set of the numbers represented by the machine is finite. This will inevitably lead to round-off and truncation errors. This should not be considered as a problem, but rather as the true nature of numerics. The notorious examples like evaluating $333.75y^6 + x^2(11x^2y^2 - y^6 - 121y^4 - 2) + 5.5y^8 + x/(2y)$ at $(x,y) = (77617, 33096)$ or plotting the polynomial $t^6 - 6t^5 + 15t^4 - 20t^3 + 15t^2 - 6t + 1$ in a small neighborhood of 1, still result in unexpected outcomes, if one is unaware of the potential risks of the floating point computations. We mention the failure of a Patriot missile on February 25, 1991 or the explosion of the unmanned space rocket Ariane 5 on June 4, 1996 as practical examples of these potential risks becoming real.

Therefore in mathematical proofs, where the beauty of the argument is its unquestionable truth itself, the usage of computers must be handled with extreme care. One technique, that is used to overcome these problems and make our computations rigorous, is called interval arithmetic.

To calculate derivatives of a given function is often considered to be a hard problem, since in general with increasing the order or the dimension, the complexity of the formula of the derivative grows exponentially. The observation, that we do not need these formulae in general, but only certain values of the derivatives, is crucial to understand why automatic differentiation is so useful.

The structure of the thesis is as follows. In Part I we give an introduction to the methods used in our papers. In Chapter 1 we get acquainted with the basic techniques, interval arithmetic, interval analysis, floating point computations and automatic differentiation. Chapter 2 gives an overview of the interaction between dynamical systems and different representations of the data. In Chapter 3 we take on the basic concept of automatic differentiation seen before, and present a method by Griewank *et al.* [17] to compute higher order derivatives of multivariate functions that will be used in Paper A. We go through the theory of graph representations in Chapter 4 by following the steps of Hohmann and Dellnitz [12] and Galias [15]. This theory may be used in qualitative analysis of maps. We give two applications in Paper B and Paper C. In addition, we give the proof of correctness of the algorithm for enclosing non-wandering points in Pa-

per B. In Chapter 5 we introduce the reader to the method of self-consistent bounds by Zgliczyński and Mischaikow [44] and Zgliczyński [40, 42, 43] that may be used to analyze a certain class of dissipative partial differential equations. An application of this concept to a destabilized Kuramoto-Sivashinsky equation is given in Paper D. Chapter 6 gives a short overview of the results of the included papers.

Part II is the main scientific contribution of this thesis, consisting of the formerly mentioned four papers.

Notations

$\mathbb{N}$	-	the set of natural numbers $1, 2, \ldots$
$\mathbb{N}_0$	-	the set of nonnegative integers $0, 1, 2, \ldots$
$\mathbb{Z}$	-	the set of integers
$\mathbb{R}$	-	the set of real numbers
$\mathbb{R}^+$	-	the set of positive real numbers
$\mathbb{R}_0^+$	-	the set of nonnegative real numbers
$\mathbb{C}$	-	the set of complex numbers
$\|\cdot\|$	-	the 2-norm on the corresponding space
$B(x;r)$	-	the open set $\{y : \|x-y\| < r\}$

Contents

Part I

Introduction

Chapter 1

Preliminaries

In this chapter we give a short introduction to the basic tools we will use. In order to achieve rigorous results, we will base our computations on intervals. That results in the so-called interval arithmetic that we discuss in Section 1.1. We take on this concept in Section 1.2 and work with interval valued versions of the standard functions, this is referred to as interval analysis. The endpoints of the intervals discussed here are real numbers for simplicity. In an implementation we must use floating point numbers, the round-off errors are controlled through using directed rounding modes of the computer. We comment on these questions in Section 1.3. In the end, we introduce the concept of automatic differentiation that we use to obtain higher order derivatives. This is discussed in Section 1.4.

1.1 Interval Arithmetic

Interval Arithmetic (IA) is the first step towards rigorous computations; we give a basic introduction here, the reader is referred to Moore [27], Alefeld [1] and Tucker [35, 36] for further details.

Definition 1.1. The closed and bounded intervals of the real line are denoted by

$$\mathbb{IR} = \{\boldsymbol{x} = [\underline{x}, \bar{x}] : -\infty < \underline{x} \leq \bar{x} < \infty\} \cup \{\emptyset\}.$$

$\underline{x}$ ($\bar{x}$) is the *lower* (*upper*) endpoint of the interval $\boldsymbol{x}$. If $\underline{x} = \bar{x}$, then we call it a *thin* interval. The natural embedding of $\mathbb{R}$ into $\mathbb{IR}$ are the thin intervals and is given by

$$\iota : \mathbb{R} \to \mathbb{IR},\ r \mapsto \boldsymbol{r} = [r, r].$$

We define the result of an arithmetic operation $\odot$ on two intervals $\boldsymbol{a}$ and $\boldsymbol{b}$ as the smallest interval containing all the numbers of the form $a \odot b$, where $a \in \boldsymbol{a}$ and $b \in \boldsymbol{b}$. It is easy to see that the set $\{a \odot b : a \in \boldsymbol{a}, b \in \boldsymbol{b}\}$ is always an interval for $+, -, \times$ and if $0 \notin \boldsymbol{b}$, then it is an interval for $\div$ as well. We can express the result interval from the

endpoints of the operands as

$$\begin{aligned}
\boldsymbol{a}+\boldsymbol{b} &= [\underline{a}+\underline{b}, \bar{a}+\bar{b}],\\
\boldsymbol{a}-\boldsymbol{b} &= [\underline{a}-\bar{b}, \bar{a}-\underline{b}],\\
\boldsymbol{a}\times\boldsymbol{b} &= [\min(\underline{a}\underline{b}, \underline{a}\bar{b}, \bar{a}\underline{b}, \bar{a}\bar{b}), \max(\underline{a}\underline{b}, \underline{a}\bar{b}, \bar{a}\underline{b}, \bar{a}\bar{b})],\\
\boldsymbol{a}\div\boldsymbol{b} &= \boldsymbol{a}\times[1/\bar{b}, 1/\underline{b}].
\end{aligned} \tag{1.1}$$

Definition 1.2. We define the *mignitude* and *magnitude* of an interval $\boldsymbol{a}\in\mathbb{IR}$ as follows

$$\begin{aligned}
\operatorname{mig}(\boldsymbol{a}) &= \min\{|a| : a\in\boldsymbol{a}\},\\
\operatorname{mag}(\boldsymbol{a}) &= \max\{|a| : a\in\boldsymbol{a}\}.
\end{aligned}$$

The *absolute value* is defined by

$$\operatorname{abs}(\boldsymbol{a}) = [\operatorname{mig}(\boldsymbol{a}), \operatorname{mag}(\boldsymbol{a})].$$

We introduce the following *metric* on $\mathbb{IR}$

$$d(\boldsymbol{a},\boldsymbol{b}) = \max\{|\underline{a}-\underline{b}|, |\bar{a}-\bar{b}|\}.$$

The intervals are sets as well, so we have set theoretical operations and relations

$$\begin{aligned}
&a\in\boldsymbol{b} \quad \text{if } \underline{b}\le a\le\bar{b}\text{; for the thin interval } \boldsymbol{a}\text{, we define } \boldsymbol{a}\in\boldsymbol{b}\Leftrightarrow a\in\boldsymbol{b},\\
&\boldsymbol{a}\subseteq\boldsymbol{b} \quad \text{if } \underline{b}\le\underline{a} \text{ and } \bar{a}\le\bar{b},\\
&\boldsymbol{a}\subset\boldsymbol{b} \quad \text{if } \underline{b}<\underline{a} \text{ and } \bar{a}<\bar{b},\\
&\boldsymbol{a}\cap\boldsymbol{b} = \begin{cases} \emptyset & \text{if } \bar{a}<\underline{b} \text{ or } \bar{b}<\underline{a},\\ [\max(\underline{a},\underline{b}), \min(\bar{a},\bar{b})] & \text{otherwise.}\end{cases}
\end{aligned}$$

The union of two intervals is not necessarily an interval. Since we want $\mathbb{IR}$ to be closed for the operations, we will use the interval hull instead of the union

$$\boldsymbol{a}\sqcup\boldsymbol{b} = [\min(\underline{a},\underline{b}), \max(\bar{a},\bar{b})].$$

The partial ordering $\le$ of the intervals is an extension of the standard ordering of $\mathbb{R}$:

$$\begin{aligned}
\boldsymbol{a}\le\boldsymbol{b} &\text{ if } \bar{a}\le\underline{b},\\
\boldsymbol{a}<\boldsymbol{b} &\text{ if } \bar{a}<\underline{b}.
\end{aligned}$$

It is often useful to represent an interval not by the endpoints, but by the midpoint and the radius

$$\begin{aligned}
\operatorname{mid}(\boldsymbol{a}) &= \frac{\underline{a}+\bar{a}}{2},\\
\operatorname{rad}(\boldsymbol{a}) &= \frac{\bar{a}-\underline{a}}{2},\\
\operatorname{symrad}(\boldsymbol{a}) &= [-\operatorname{rad}(\boldsymbol{a}), \operatorname{rad}(\boldsymbol{a})],
\end{aligned}$$

thus

$$\boldsymbol{a} = \operatorname{mid}(\boldsymbol{a}) + \operatorname{symrad}(\boldsymbol{a}) = \operatorname{mid}(\boldsymbol{a}) + \operatorname{rad}(\boldsymbol{a})\times[-1,1].$$

1.2 Interval Analysis

We can do arithmetic with intervals, consequently we have rational functions of intervals. We want to extend this and handle the standard functions of intervals as well. It turns out that two properties – range inclusion and inclusion isotonicity – will characterize the *good* interval functions. More details may be found, again, in Moore [27], Alefeld [1] and Tucker [35, 36].

Definition 1.3. We say that the function $F\colon \mathcal{D}_F \subseteq \mathbb{IR} \to \mathbb{IR}$ is an *interval extension* of the real function $f\colon \mathbb{R} \to \mathbb{R}$ if it satisfies for all $\boldsymbol{x} \in \mathcal{D}_F$ that

$$\{f(x) : x \in \boldsymbol{x}\} \subseteq F(\boldsymbol{x}) \text{ (range inclusion)},$$
$$\boldsymbol{y} \subseteq \boldsymbol{z} \subseteq \boldsymbol{x} \Rightarrow F(\boldsymbol{y}) \subseteq F(\boldsymbol{z}) \text{ (inclusion isotonicity)}.$$

Remark 1.4. If F is an interval extension of f, then so is $\boldsymbol{x} \mapsto F(\boldsymbol{x}) + \boldsymbol{x} - \boldsymbol{x}$.

Example 1.5. *Some interval extensions:*

$$\boldsymbol{x}^n = \begin{cases} [(\underline{x})^n, (\bar{x})^n] & \text{if } n \in \mathbb{Z}^+ \text{ is odd,} \\ [\operatorname{mig}(\boldsymbol{x})^n, \operatorname{mag}(\boldsymbol{x})^n] & \text{if } n \in \mathbb{Z}^+ \text{ is even,} \\ [1,1] & \text{if } n = 0, \\ [1/\bar{x}, 1/\underline{x}]^n & \text{if } n \in \mathbb{Z}^- \text{ and } 0 \notin \boldsymbol{x}, \end{cases}$$
$$e^{\boldsymbol{x}} = [e^{\underline{x}}, e^{\bar{x}}].$$

Having interval extensions for the standard functions – arithmetic operations, trigonometric functions, exponential, logarithmic functions, power function – we may obtain interval extensions for finite combination of these – the so-called *elementary functions* – simply by replacing every occurrence of the variable x with $\boldsymbol{x}$. The obtained interval valued function is called the *natural extension*, it is easy to show that it satisfies the two required properties. As multiple interval extensions exist for a given function, it is important to take care which one we choose. Obviously, we want the inclusion to be as tight as possible. As a rule of thumb, we should go with that natural extension of f (obtained from the formula for f) which has the minimal number of appearances of the variable x.

Example 1.6. $[-1,1]^2 = [0,1]$, *while* $[-1,1] \times [-1,1] = [-1,1]$.

The reason for this phenomenon is that Interval Analysis is unable to distinguish between the operands and in fact, it handles every appearance of the same variable as an independent one. This is called the *dependency problem*. If f is differentiable and we have the interval extension F' of f', then using the mean value theorem we obtain

$$f(\boldsymbol{x}) \subseteq f(\operatorname{mid}(\boldsymbol{x})) + F'(\boldsymbol{x}) \times \operatorname{rad}(\boldsymbol{x})\,[-1,1]. \tag{1.2}$$

We emphasize that $f(\operatorname{mid}(\boldsymbol{x}))$ is a point value. Formula (1.2) may result in a tighter enclosure than just using the natural extension $F(\boldsymbol{x})$.

Remark 1.7. Naturally, we may consider higher order Taylor expansions of the function f – usually centered at the midpoint of the interval $\boldsymbol{x}$ – and use Taylor's theorem to enclose the range of the function by calculating the Taylor coefficients and enclosing the remainder.

Remark 1.8. It is straightforward to generalize these ideas to $\mathbb{R}^n$ and thus obtain the set of n-dimensional *interval boxes* denoted by $\mathbb{IR}^n$. The same concept, as we have presented above, is used to obtain interval extensions of functions of the form $f\colon \mathbb{R}^n \to \mathbb{R}^m$.

There are several freely available interval software. We list some, without attempting to be comprehensive: the `C++` libraries `CAPD` [10], `Filib++` [22]; and `INTLAB` [32], the interval toolbox for `MATLAB`.

1.3 Interval Arithmetic and floating-point numbers

A modern computer uses floating-point numbers in general. When evaluating an operation, the result is rounded to a certain floating-point number represented in the computer, also called *representable number* or *machine number*. The set of these numbers is denoted by $\mathbb{F}$. The rounding is described by the *rounding mode*, we list the most common ones in Table 1.1.

Table 1.1: The standard rounding modes

Rounding mode	Notation
Round to Zero	$\bigcirc$
Round Up	$\triangle$
Round Down	$\triangledown$
Round to Nearest	$\square_n$

By default, the computations are carried out using $\square_n$. In order to calculate rigorously, we shall keep switching between Round Up and Round Down. We have shown several formulae in (1.1) that calculate the endpoints of the result of an operation. In an actual implementation, we shall evaluate the lower endpoint using $\triangledown$ and the upper endpoint using $\triangle$.

We use intervals with floating point endpoints in practice. The set of these intervals is denoted by $\mathbb{IF}$. As we have said before, the arithmetic is the same as for $\mathbb{IR}$, but using directed rounding modes:

$$\begin{aligned}
\boldsymbol{a}+\boldsymbol{b} &= [\triangledown(\underline{a}+\underline{b}), \triangle(\bar{a}+\bar{b})],\\
\boldsymbol{a}-\boldsymbol{b} &= [\triangledown(\underline{a}-\bar{b}), \triangle(\bar{a}-\underline{b})],\\
\boldsymbol{a}\times\boldsymbol{b} &= [\triangledown(\min(\underline{a}\,\underline{b}, \underline{a}\bar{b}, \bar{a}\underline{b}, \bar{a}\bar{b})), \triangle(\max(\underline{a}\,\underline{b}, \underline{a}\bar{b}, \bar{a}\underline{b}, \bar{a}\bar{b}))],\\
\boldsymbol{a}\div\boldsymbol{b} &= \boldsymbol{a}\times[\triangledown(1/\bar{b}), \triangle(1/\underline{b})].
\end{aligned}$$

The difference between real interval arithmetic and floating-point interval arithmetic, while being very important, is only a technical issue in our setting. We shall design our

algorithms to work with $\mathbb{IR}$, assuming that they are represented by $\mathbb{IF}$ and the operations are implemented using proper rounding modes by the interval software. The calculations on the computer, carried out in this manner, are called *rigorous calculations*. We refer to Tucker [36] for further reading.

1.4 Automatic Differentiation

The possibility of tighter inclusions or the numerical methods for differential equations – that are usually based on *Taylor expansions* – undoubtedly serve as a motivation to obtain derivatives of a function. It is well known that the complexity of the formulae for the derivatives of a function f may increase exponentially, making it practically impossible to calculate (and store) them. *Automatic Differentiation (AD)* is an extremely handy concept for calculating the value of a derivative at a point and not the formula itself. From a practical point of view, we usually evaluate the formulae at certain points only, so we might as well obtain those values instead. The theory of AD distinguishes between the so-called Forward Automatic Differentiation and Backward Automatic Differentiation. In the forward mode we propagate tangents, while in the backward mode we propagate gradients. The book of Griewank [16] discusses these topics in detail. In addition, we refer to Moore [27] and Tucker [36].

As an example of the forward mode, consider the differentiable function $f\colon \mathbb{R} \to \mathbb{R}$, $x \mapsto f(x)$. The goal is to evaluate the derivative $f'(x)$ at several points. We will represent every quantity during the calculation with a vector (q, q'), where q stands for the value of f and q' for the value of the derivative f'. This implies representing the variable x at x_0 with $(x_0, 1)$ and any constant c with $(c, 0)$. Consider the following arithmetic rules for vectors of this kind:

$$\begin{aligned}(a,a') \pm (b,b') &= (a \pm b, a' \pm b'),\\ (a,a') \times (b,b') &= (ab, a'b + ab'),\\ (a,a') \div (b,b') &= (a/b, (a'b - ab')/b^2).\end{aligned} \tag{1.3}$$

Assume that f is a rational function and pick a point x_0. Replace every constant and every variable x in the formula for f with the vectors given above. If we evaluate this expression following the rules (1.3), it is apparent that we obtain $f(x_0)$ in the first component and in addition, we will *automatically* have $f'(x_0)$ in the second. One may derive the appropriate rules for the standard functions and thus obtain the potential to calculate the derivative of any elementary function at a given point x_0.

Similar rules can be given for the second and higher order derivatives (resulting in a computation with higher dimensional vectors), but it gets rather tedious to write and implement them. This is the time to get greedy and aim for derivatives of *arbitrary* high order. Usually we encounter these quantities scaled, in the form of Taylor coefficients. It turns out that the rules for propagating these coefficients are rather nice and simple. Let $(a)_k$ and $(b)_k$ denote the k-th Taylor coefficient of a and b with respect to the variable x.

The rules for arithmetic operations are the following:

$$
\begin{aligned}
(a \pm b)_k &= (a)_k \pm (b)_k, \\
(a \times b)_k &= \sum_{i=0}^{k} (a)_i (b)_{k-i}, \\
(a \div b)_k &= \frac{1}{(b)_0} \left((a)_k - \sum_{i=0}^{k-1} (a \div b)_i (b)_{k-i} \right) \text{ if } (b)_0 \neq 0.
\end{aligned}
$$

With some effort, one can derive similar rules for evaluating the standard functions, we include these for `exp`, `sin` and `cos` as an example:

$$
(e^a)_k = \begin{cases} e^{(a)_0} & \text{if } k = 0, \\ \frac{1}{k} \sum_{i=1}^{k-1} i (a)_i (e^a)_{k-i} & \text{if } k > 0, \end{cases}
$$

$$
(\sin a)_k = \begin{cases} \sin(a)_0 & \text{if } k = 0, \\ \frac{1}{k} \sum_{i=1}^{k} i (a)_i (\cos a)_{k-i} & \text{if } k > 0, \end{cases}
$$

$$
(\cos a)_k = \begin{cases} \cos(a)_0 & \text{if } k = 0, \\ -\frac{1}{k} \sum_{i=1}^{k} i (a)_i (\sin a)_{k-i} & \text{if } k > 0. \end{cases}
$$

Note that in the implementation of an AD software, the trigonometric functions `sin` and `cos` must be computed in parallel.

The previously mentioned `C++` library `CAPD` [10] has built in Automatic Differentiation capabilities as well. We recommend the `FADBAD++` [3] library developed by Bendtsen and Stauning for more general purposes.

Chapter 2

Dynamical systems and data structures

In this chapter, we study *Difference Equations (DE)*, *Ordinary Differential Equations (ODE)* and *Differential Inclusions (DI)*. We adopt the concept that a *Dynamical System (DynSys)* acts on a data that is represented in the form of a *Dynamical Set (DynSet)*.

Thus, we start with studying different representations for the data. As we are working with rigorous numerics, it is ultimately given by a finite collection of intervals. We discuss the representation of *finite* data in Section 2.1. In order to fight the *wrapping-effect*, we favor the so-called Lohner-sets (see Lohner [24]), an overview of this concept is given in Section 2.1.2. Another widely used technique is the so-called Taylor-models that we will not discuss here, the reader is referred to Berz and Makino [26]. We comment on representing *infinite* dimensional data in Section 2.2.

In Section 2.3 we analyze how a map acts on the represented data. The continuous nature of an ODE is captured in the computer by using a certain timestep h and the corresponding time-h map of the flow. We discuss in Section 2.4 how to place this into the framework presented so far. Finally, we include a brief overview of the results for DIs by Zgliczyński and Kapela [21] in Section 2.5. Similar theories have been established for other type of dynamical systems such as *Partial Differential Equations (PDE)* and *Delay Differential Equations (DDE)* as well. We shall give a short introduction to the method of self-consistent bounds by Zgliczyński for dissipative PDEs in Chapter 5.

2.1 Data in *finite* dimension

2.1.1 Interval Boxes

Interval boxes are the higher dimensional analogues for intervals. $\boldsymbol{X}_b \in \mathbb{IR}^n$ is an interval box enclosure of the set $X \in \mathbb{R}^n$ if $X_i \subseteq (\boldsymbol{X}_b)_i$ for every coordinate $i = 1, \ldots, n$. They might be suitable in special cases, but in general they result in huge overestimation, due to the so-called *wrapping effect*. The following example demonstrates this phenomenon.

Consider a 2-dimensional interval box and the map ρ, a rotation by $\frac{\pi}{4}$ in the plane. After each rotation, the image of our set is enclosed in a box, therefore the area doubles at least. This will lead to a *blow-up* eventually, even though ρ is volume-preserving. The situation is depicted on Figure 2.1.

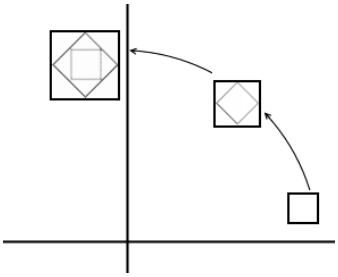

Figure 2.1: Rotation by $\frac{\pi}{4}$ on the plane.

It is vital to observe that the phenomenon is a consequence of the representation of the set itself and not the way we evaluate ρ.

2.1.2 Lohner-sets

We may store our data as an interval box, together with a local coordinate system. Numerous articles have been published on this topic, we refer to Lohner [24], Nedialkov *et al.* [29], Mrozek and Zgliczyński [28], Zgliczyński and Wilczak [38] and Zgliczyński [41].

One natural way to represent the set $A \in \mathbb{R}^n$ in such way is a *parallelepiped*. That is

$$X \subseteq \boldsymbol{X}_p = m + \boldsymbol{C} \cdot \boldsymbol{r},$$

where the vector $m \in \mathbb{R}^n$ represents the center of the set X, $\boldsymbol{C} \in \mathbb{IR}^{n\times n}$ and $\boldsymbol{r} \in \mathbb{IR}^n$. Consider now our previous example about ρ, the rotation by $\frac{\pi}{4}$, and the data represented as a parallelepiped $\boldsymbol{X}_p = m + \boldsymbol{C} \times \boldsymbol{r} \subseteq \mathbb{R}^2$. It is easily shown that

$$\rho(\boldsymbol{X}_p) = \rho(m) + \left(\begin{pmatrix} \cos\frac{\pi}{4} & \sin\frac{\pi}{4} \\ - \sin\frac{\pi}{4} & \cos\frac{\pi}{4} \end{pmatrix} \boldsymbol{C} \right) \cdot \boldsymbol{r},$$

is an enclosure of $\rho(X)$.

As a slight modification, we may require that the matrix $\boldsymbol{C}$ is in fact a thin, orthogonal matrix $Q \in \mathbb{R}^{n\times n}$; the resulting structure is called a *cuboid*. One may use the QR-decomposition, in order to obtain such structure.

Finally, we present the *doubleton*

$$\boldsymbol{X}_d = m + C \cdot \boldsymbol{r}_0 + \boldsymbol{r},$$

that turns out to be a rather good representation to use in many applications. The product $C \cdot \boldsymbol{r}_0$ is referred to as the *linear part*, with $C \in \mathbb{R}^{n\times n}$ and $\boldsymbol{r}_0 \in \mathbb{IR}^n$. The role of the last term, the *error part* $\boldsymbol{r}$, is to incorporate what remains after a computation. The set $\boldsymbol{r}$ is represented as another standard enclosure, for example an interval vector in $\mathbb{IR}^n$ or the product of an orthogonal matrix and an interval vector.

Remark 2.1. In a rigorous implementation, we do not use *real* or more precisely *floating-point* values, but thin intervals representing these quantities. The way of dealing with these changes is of rather technical nature and is not discussed here. The general idea behind the representations is apparent using real values as well.

When solving ODEs, we might be interested in propagating and storing higher order derivatives as we integrate the equations. Thus, the data will represent not only the value (C_0 information), but the Taylor-coefficients – or derivatives – up to order n. The generalized version of doubletons is called C_n-set. For further details, see Zgliczyński and Wilczak [38] and Zgliczyński [41].

2.2 Data in *infinite* dimension

In this Section we assume that our data is given as the infinite series $(x_k)_{k=0}^{\infty}$, where $x_k \in \mathbb{R}$ for all $k \in \mathbb{N}_0$. As an example think of the coefficients of a power- or Fourier series. Let $M \in \mathbb{N}$ and assume that there exists an interval valued function $F_M \colon \mathbb{N} \to \mathbb{IR}$ such that F_M may be described by a finite number of parameters $p_1, \dots p_n$; moreover $x_k \in F_M(k)$ is satisfied for $k \geq M+1$. If $\boldsymbol{X}$ is an enclosure of $\{x_1\} \times \ldots \times \{x_M\}$, then we obtain the finite representation $\{\boldsymbol{X}, p_1, \dots, p_n\}$ for an enclosure of $(x_k)_{k=0}^{\infty}$. The first M elements of the series are referred to as the *finite part* or *main part*, the remaining elements form the *tail*.

As an example, assume that $\boldsymbol{X}_b \in \mathbb{IR}^n$ is an interval box enclosure of the vector $X = (x_1, \dots, x_M) \in \mathbb{R}^n$ and $x_k \in F_M(k) = \frac{C}{k^s}[-1,1]$ is satisfied for all $k \geq M+1$, with the real parameters $C > 0$ and $s \in \mathbb{R}$. Based on $\{\boldsymbol{X}_b, C, s\}$, we are able to enclose each element of the series. We will encounter such objects in the method of self-consistent bounds, discussed in Chapter 5.

2.3 Difference Equations and Maps

2.3.1 Description of the *Dynamical System*

Consider the map

$$x_{k+1} = f(x_k), \quad k = 0, 1, \dots \tag{2.1}$$

where $f \colon \mathcal{D}_f \subseteq \mathbb{R}^n \to \mathbb{R}^n$ is a continuous function. Equation (2.1) is also referred to as a *Difference Equation (DE)* or a discrete equation. Let $f^{-1}(x) = \{y \in \mathcal{D}_f : f(y) = x\}$, for $x \in \mathbb{R}^n$. For $k \in \mathbb{N}_0$, f^k denotes the k-fold composition of f, i.e., $f^{k+1}(x) = f(f^k(x))$, and $f^0(x) = x$.

Definition 2.2. The *forward orbit* of the point x is

$$\Gamma^+(x) := \{f^k(x) : k \in \mathbb{N}\}.$$

The *backward orbit* is

$$\Gamma^-(x) := \{y : \exists k \in \mathbb{N} : f^k(y) = x\},$$

their union is the *orbit* of x, denoted by $\Gamma(x)$.

Definition 2.3. The point $x^* \in \mathcal{D}_f$ is called a *fixed point* of f if $f(x^*) = x^*$. The point $q \in \mathcal{D}_f$ is a *periodic point of f with minimal period m* if $f^m(q) = q$ and for all $0 < k < m : f^k(q) \neq q$; $q \in \mathcal{D}_f$ is *eventually periodic* if it is not periodic, but there is a k_0 such that $f^{k_0}(q)$ is periodic.

Besides fixed points and periodic orbits, we are interested in the following objects:

Definition 2.4. The set $A \subseteq \mathcal{D}_f$ is said to be *forward invariant* under f if

$$A = f(A),$$

backward invariant if

$$A = f^{-1}(A) \text{ and } \forall x \in A : f^{-1}(x) \neq \emptyset,$$

invariant if it is both backward and forward invariant. An invariant set A is called *attracting set* if there exists an open neighbourhood $\mathcal{U} \subseteq \mathcal{D}_f$ of A such that

$$(\forall \text{ open neighbourhood } V \supseteq A)\ (\exists M = M(V) \in \mathbb{N}) \text{ such that } \forall m \geq M : f^m(\mathcal{U}) \subseteq V.$$

This neighbourhood $\mathcal{U}$ is called a *fundamental neighbourhood* of A. The *basin of attraction* of A is $\cup_{k \in \mathbb{N}_0} f^{-k}(\mathcal{U})$. If $\mathcal{A}$ is compact, invariant and has the whole domain $\mathcal{D}_f$ as a basin of attraction, then we call it the *global attractor*. The point $q \in \mathcal{D}_f$ is a *non-wandering point* of (2.1), if for every neighbourhood U of q and for all $M \geq 0$, there exists an integer $m \geq M$ such that $f^m(U \cap \mathcal{D}_f) \cap U \cap \mathcal{D}_f \neq \emptyset$.

Remark 2.5. For a more throughout introduction to maps, see Devaney [14].

We often restrict our analysis to a compact subset of the space, especially in the case of computer-aided proofs. We shall utilize the concept of relative objects from Hohmann, Dellnitz [12] and Galias [15]. In the following, let K be a compact subset of $\mathcal{D}_f$.

Definition 2.6. The *invariant part of K* is the largest invariant set contained in K, and is denoted by $\mathrm{Inv}(f;K)$. The *non-wandering part of K* is the subset of $\mathrm{Inv}(f;K)$, formed by the non-wandering points

$$\mathrm{NonW}(f;K) = \{x \in \mathrm{Inv}(f;K) : x \text{ is non-wandering}\}.$$

Let $\mathcal{A}$ be the global attractor of (2.1). The *global attractor relative to K* is

$$\mathcal{A}_K = \{x \in \mathcal{A} : f^{-k}(x) \cap K \neq \emptyset, \text{ for all } k \geq 0\}.$$

The set of fixed points and periodic orbits in K are denoted by $\mathrm{Fix}(f;K)$ and $\mathrm{Per}(f;K)$, respectively. In addition, $\mathrm{Per}_{\leq m}(f;K)$ denotes the set of periodic points in K with a period not larger than m.

Remark 2.7. It is obvious from the definition that $\mathcal{A}_K \subseteq \mathcal{A}$ and $\mathcal{A}_K$ is backward invariant. $\mathcal{A}_K$ is compact if f has a continuous inverse, since $\mathcal{A}$ is compact. However, $\mathcal{A}_K$ is not necessary invariant.

It is not always true that $\mathcal{A}_K = \mathcal{A} \cap K$. See Figure 2.2, the periodic orbit $x \to y \to z \to v \to x$ and the fixed point q are parts of the global attractor $\mathcal{A}$, but no points from the periodic orbit will be part of $\mathcal{A}_K$, even though $y, z \in K$.

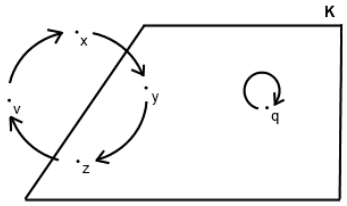

Figure 2.2: Global attractor relative to K.

2.3.2 Propagation of enclosures

Let $\boldsymbol{X}$ be a finite dimensional enclosure of the set $X \subset \mathbb{R}^n$ and $F\colon \mathbb{IR}^n \to \mathbb{IR}^n$ an interval extension of the function f. We will enclose $f(X)$ with taking into consideration the representation of $\boldsymbol{X}$ and the smoothness of f. Since F is an interval extension, we have

$$f(X) \subseteq F(\boldsymbol{X}_b).$$

Assume that f is a differentiable function and an interval extension of $\mathrm{D}f$ is given by $\mathrm{D}F : \mathbb{IR}^n \to \mathbb{IR}^{n\times n}$. As we have seen in (1.2), we may use the mean value theorem to show that

$$f(X) \subseteq f(\mathrm{mid}(\boldsymbol{X}_b)) + \mathrm{D}F(\boldsymbol{X}_b) \cdot \mathrm{symrad}(\boldsymbol{X}_b).$$

Using this, we obtain the enclosure formulae for parallelepiped and doubleton representations as follows.

1. For the parallelepiped representation $\boldsymbol{X}_p = m + \boldsymbol{C} \cdot \boldsymbol{r}$, it holds that

   $$f(X) \subseteq f(m) + \boldsymbol{C}' \cdot \boldsymbol{r},$$

   with $\boldsymbol{C}' = \mathrm{D}F(\boldsymbol{X}_p) \cdot \boldsymbol{C}$.

2. Having the doubleton representation $\boldsymbol{X}_d = m + C \cdot \boldsymbol{r}_0 + \boldsymbol{r}$, we obtain

   $$f(X) \subseteq f(m) + C' \cdot \boldsymbol{r}_0 + \boldsymbol{r}',$$

   where $C' = \mathrm{mid}(\mathrm{D}F(\boldsymbol{X}_p) \cdot C)$; and $\boldsymbol{r}'$ is such that the result gives an enclosure.

Remark 2.8. The *error part* in a doubleton is, in principle, supposed to be smaller than the *linear part*. If it becomes too large, it is advised to *rearrange* the representation based on a particular *reorganization policy*. The reader is referred to the article by Mrozek and Zgliczyński [28].

2.4 Ordinary Differential Equations

2.4.1 Description of the *Dynamical System*

We shall not discuss the standard terminology here, as we did for maps in Section 2.3, since we shall work with the time-h map of an *Ordinary Differential Equation (ODE)* in practice. For an introduction to the theory of ODEs, the reader is referred to Arnold [2], Hirsch, Smale and Devaney [19] or Boyce and DiPrima [7].

Let $f : \mathbb{R}^n \to \mathbb{R}^n$ and consider the autonomous ODE

$$y'(t) = f(y(t)). \tag{2.2}$$

If we couple equation (2.2) with the *initial condition* $y(t_0) = y_0 \in \mathbb{R}^n$, we obtain

$$\begin{cases} y'(t) = f(y(t)), \\ y(t_0) = y_0, \end{cases} \tag{2.3}$$

an *Initial Value Problem (IVP)*. As we know, if f is Lipschitz-continuous, then there exists a unique solution locally for all initial conditions. We denote the solution of (2.3) by $y_{y_0;t_0}(t)$. The function $y_{y_0;t_0}$ is defined in a neighbourhood of t_0, satisfies $y_{y_0;t_0}(t_0) = y_0$ and $y'_{y_0;t_0}(t) = f(y_{y_0;t_0}(t))$.

2.4.2 The time-h map

For a given $h > 0$, we consider the *time-h map* φ_h corresponding to the ODE (2.2):

$$\varphi_h \colon (y_0, t_0) \mapsto (y_{y_0;t_0}(h), t_0 + h) = (y_h, t_h).$$

The question is how to obtain the first component y_h – we shall do this using the Taylor method that is based on computing the Taylor expansion of the solution at time t_0. Using the notation from Section 1.4 and assuming that f is an analytic function gives us

$$y_h = \sum_{i=0}^{\infty} (y_{y_0;t_0}(t_0))_i h^i.$$

We can list the first two Taylor coefficients of the solution at once

$$\begin{aligned} (y_{y_0;t_0}(t_0))_0 &= y_0, \\ (y_{y_0;t_0}(t_0))_1 &= f(y_0). \end{aligned}$$

In order to obtain them up to an arbitrary order, we utilize that

$$\frac{1}{(k+1)!} \frac{d^{k+1}}{dt^{k+1}} y_{y_0;t_0}(t) \bigg|_{t=t_0} = \frac{1}{k+1} \left(\frac{1}{k!} \frac{d^k}{dt^k} f(y_{y_0;t_0}(t)) \right) \bigg|_{t=t_0},$$

therefore

$$(y_{y_0;t_0}(t_0))_{k+1} = \frac{1}{k+1} (f(y_{y_0;t_0}(t_0)))_k.$$

Notice that in order to obtain the k-th Taylor coefficient of $f(y_{y_0;t_0}(t))$, we need to know the Taylor coefficients of order $0,\ldots,k$ of $y_{y_0;t_0}(t)$. Thus, in theory, we can calculate $(y_{y_0;t_0}(t_0))_k$ up to an arbitrary order using a recursive procedure. With automatic differentiation, this becomes possible in practice as well. The procedure may be highly optimized by building a *Directed Acyclic Graph (DAG)* that represents the evaluation of the function f. The vertices of the graph may be repeatedly filled up with the higher and higher order Taylor coefficients of the (intermediate) quantities that they represent in the evaluation of f. The reader is referred for a throughout analysis to Griewank [16] and for an actual implementation to Bendtsen and Stauning [3].

2.4.3 Rigorous time-h map

Let $h > 0$ and assume that the (2.3) has a unique solution that exists for $t \in [t_0, t_0 + h]$. We obtain an enclosure – by using interval analysis – of the time-h map centered at y_0, by truncating the Taylor expansion at a given $N \in \mathbb{N}$ and adding the remainder to the polynomial enclosure. Let ϕ_h^N be an interval extension of this truncated Taylor expansion

$$\sum_{i=0}^{N} (y_{y_0;t_0}(t_0))_i \, h^i \in \phi_h^N(y_0, t_0). \tag{2.4}$$

The remainder $(y_{y_0;t_0}(\xi))_{N+1} \, h^{N+1}$ contains an unknown $\xi \in [0,h]$. Thus, enclosing it using intervals is straightforward:

$$(y_{y_0;t_0}(\xi))_{N+1} \, h^{N+1} \in \mathrm{Rem}_{h;N+1}(y_0, t_0) = (y_{y_0;t_0}([0,h]))_{N+1} \, h^{N+1}. \tag{2.5}$$

Formulae (2.4) and (2.5) result in the following enclosure of the time-h map:

$$y_h \in \phi_h^N(y_0, t_0) + \mathrm{Rem}_{h;N+1}(y_0, t_0). \tag{2.6}$$

Observe that in order to compute (2.6), we need to establish that the solution does exist and give an enclosure in advance for $y_{y_0;t_0}(t)$ on $[t_0, t_0 + h]$ as well, since it is used when we evaluate the remainder. This is called the *a-priori* enclosure of the solution and is obtained by finding – through an iterative procedure – an interval box $\boldsymbol{Y}$ containing y_0 such that

$$y_0 + F(\boldsymbol{Y})h \subset \mathrm{int}(\boldsymbol{Y}).$$

If we do not succeed in finding such box, then we abort the evaluation with an error message or try to decrease the timestep h. For further details, the reader is referred to Moore [27], Lohner [24] and Tucker [36].

Remark 2.9. Note that if we replace h by the interval $[0,h]$ in formulae (2.4) and (2.6), then we obtain a set that is an *enclosure* of the trajectory for $t \in [t_0, t_0 + h]$. This means that for all $t \in [0,h]$ the inclusion

$$y_t \in \sum_{i=0}^{N} (y_{y_0;t_0}(t_0))_i \, [0,h]^i + (y_{y_0;t_0}([0,h]))_{N+1} \, [0,h]^{N+1}$$

is satisfied.

2.4.4 Propagating doubletons

If the data is represented as a doubleton, we need the rigorous Jacobi matrix $\mathrm{Jac}_y \phi_h^N(y_0,t_0)$ of $\phi_h^N(y_0,t)$ at $t=t_0$. By using automatic differentiation in the space variable as well, this may computed together with $\phi_h^N(y_0,t_0)$. The value y_h is then contained in the doubleton

$$y_h \in \varphi_h^N(m,t_0) + C' \cdot \boldsymbol{r}_0 + \boldsymbol{r}',$$

where $C' = \mathrm{mid}(\mathrm{Jac}_y \phi_h^N(y_0,t_0) \cdot C)$, and $\boldsymbol{r}'$ is such that the result gives an enclosure.

2.5 Integration of a Differential Inclusion

Consider the perturbed ordinary differential equation

$$\begin{cases} x'(t) = f(x(t),y(t)), \\ x(0) = x_0, \end{cases} \tag{2.7}$$

where $x_0 \in \mathbb{R}^n$, the function $f\colon \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is continuously differentiable and the *perturbation* is given by $y\colon \mathbb{R} \to \mathbb{R}^m$. Equations of type (2.7) arise in various problems such as in control theory or in the rigorous integration of dissipative PDEs. The following Theorem by Zgliczyński and Kapela [21] describes how to obtain a rigorous solution for (2.7). For the proof and a through introduction, see the paper referred above.

Theorem 2.10. *Assume that $t_0, h \in \mathbb{R}$ and $h > 0$. Let $f\colon \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \to \mathbb{R}^{n_1}$ be a continuously differentiable function. For a fixed $y_c \in \mathbb{R}^{n_2}$ and a bounded and continuous function $y\colon [t_0,t_0+h] \to \mathbb{R}^{n_2}$ consider*

$$x'(t) = f(x(t),y_c), \qquad x(t_0) = x_0, \tag{2.8}$$

$$x'(t) = f(x(t),y_c) + (f(x(t),y(t)) - f(x(t),y_c)), \qquad x(t_0) = x_0. \tag{2.9}$$

Let $x_1, x_2\colon [t_0,t_0+h] \to \mathbb{R}^{n_1}$ be solutions of (2.8) and (2.9), respectively. We assume that

- *$W_y \subset \mathbb{R}^{n_2}$ is a convex set such that $y([t_0,t_0+h]) \subset W_y$,*
- *$W_1 \subset W_2 \subset \mathbb{R}^{n_1}$ are convex and compact sets such that for $s \in [t_0,t_0+h]$ the inclusions $x_1(s) \in W_1$ and $x_2(s) \in W_2$ are satisfied.*

Then for $t \in [t_0,t_0+h]$ the inequality (the subscript i denotes the i-th component)

$$|x_{1,i}(t) - x_{2,i}(t)| \le \left(\int_{t_0}^{t} e^{J(t-s)} C ds \right)_i, \quad i = 1,\dots,n_1 \tag{2.10}$$

holds, provided that $C \in \mathbb{R}^{n_1}$ and $J \in \mathbb{R}^{n_1 \times n_1}$ satisfy the conditions

$$C_i \ge \sup\{|f_i(x,y_c) - f_i(x,y)|, x \in W_1, y \in W_y\},\ i = 1,\dots,n_1, \tag{2.11}$$

$$J_{ij} \ge \begin{cases} \sup \frac{\partial f_i}{\partial x_j}(W_2,W_y) & \text{if } i = j, \\ \sup \left| \frac{\partial f_i}{\partial x_j}(W_2,W_y) \right| & \text{if } i \ne j. \end{cases} \tag{2.12}$$

Remark 2.11. The sets W_1, W_2 and W_y are called *a-priori* enclosures. They are rough, but rigorous estimates that we have to obtain in advance. This is a similar situation to the one we have seen in the previous Section for integrating ODEs. As a matter of fact, W_1 is an a-priori enclosure for the ODE (2.8) in the sense of the discussion therein.

Assume now that we obtained these rough enclosures. We get the solution $x_1(t)$ by rigorously integrating the corresponding ODE. Evaluating (2.11) and (2.12) using rigorous computations, we immediately obtain good candidates for C_i and J_{ij}, thus we may give an enclosure for $x_2(t)$ using (2.10).

Chapter 3

Evaluating Multivariate Derivatives

Let $f\colon \mathbb{R}^n \to \mathbb{R}$ be a sufficiently smooth function such that all appearing derivatives exist and are continuous. Our goal in this Chapter is to evaluate certain higher order derivatives of f. For this purpose, one may use higher order automatic differentiation techniques for multivariate functions, the reader is referred to Berz [4] and Danis [11]. We shall present here a different approach by Griewank, Utke and Walther [17] and Griewank [16]. We will rely on the techniques mentioned in the short introduction to AD in Section 1.4. We obtain the sought values by interpolating from univariate, directional Taylor coefficients. We have used the formulae presented in this Chapter in Paper A in order to give a feasible way to compute elementary differentials and B-series (see Butcher [8] and Hairer [18]).

First we introduce the notations for *multi-indices* and the *seed matrix* in Section 3.1. Using these, we derive a closed formula for higher order *derivatives of polynomials* in Section 3.2. We take on this and in Section 3.3, we obtain an expression for higher order *derivatives of smooth functions* by interpolating from certain univariate Taylor coefficients. In Section 3.4, we derive our final formula for interpolation. The number of nonvanishing coefficients in this expression is discussed in Section 3.5.

3.1 Multi-indices and the seed matrix

Fix the integer $p \geq 1$ for the time being. We shall not denote explicitly that the dimension of a quantity is dependent on p, it is rather straightforward to use the formulae with different p-s later on.

Let $\mathbf{i} = (\mathbf{i}_1, \ldots, \mathbf{i}_p) \in \mathbb{N}_0^p$ be a *multi-index* with the norm $|\mathbf{i}|$ defined as $|\mathbf{i}| = \sum_{r=1}^{p} \mathbf{i}_r$. The multi-indices $\mathbf{i}$ and $\mathbf{j}$ satisfy $\mathbf{j} \leq \mathbf{i}$ if the relation is satisfied componentwise. Consequently, $\mathbf{j} < \mathbf{i}$ is true if $\mathbf{j} \leq \mathbf{i}$ and $\mathbf{j} \neq \mathbf{i}$ stand. We denote by $\mathbf{0}$ and $\mathbf{1}$ the multi-indices that contain only zeros or ones, respectively. Naturally, a multi-index is a real vector in $\mathbb{R}^p$ as well, therefore we may use them in the standard algebraic operations.

Let $\mathbf{s_r} \in \mathbb{R}^n$ be a real vector for all $r = 1, \ldots, p$. The $\mathbf{S} \in \mathbb{R}^{n \times p}$ matrix that has the column vectors $\mathbf{s_j}$,

$$\mathbf{S} = \left[\begin{array}{c} \mathbf{s_1}; \ldots; \mathbf{s_p} \end{array} \right]$$

is called the *seed matrix*.

Our goal is to evaluate $\nabla^d_{\mathbf{S}} f(\mathbf{x})$, the d-th derivative tensor of $f(\mathbf{x}+\mathbf{Sz})$ with respect to $\mathbf{z}$ at $\mathbf{z}=\mathbf{0}$. This means that we have to obtain partial derivatives of the form

$$f_{\mathbf{i}}(\mathbf{x}) = \left.\frac{\partial^{|\mathbf{i}|} f(\mathbf{x}+z_1\mathbf{s_1}+\ldots+z_p\mathbf{s_p})}{\partial z_1^{\mathbf{i}_1}\ldots\partial z_p^{\mathbf{i}_p}}\right|_{\mathbf{z}=\mathbf{0}}, \tag{3.1}$$

where $\mathbf{i}\in\mathbb{N}_0^p$ is a multi-index with $1\le|\mathbf{i}|\le d$.

3.2 Higher order derivatives of polynomials

Proposition 3.1. *Let P be a polynomial of degree p or less, $\mathbf{S}\in\mathbb{R}^{n\times p}$ a seed matrix and $\mathbf{z}\in\mathbb{R}^p$ a vector. It holds that*

$$\frac{\partial^p P(\mathbf{Sz})}{\partial z_1\ldots\partial z_p} = \sum_{\mathbf{i}_1=0}^{1}\ldots\sum_{\mathbf{i}_p=0}^{1} P(\mathbf{i}_1\mathbf{s_1}+\ldots+\mathbf{i}_p\mathbf{s_p})(-1)^{p-(\mathbf{i}_1+\ldots+\mathbf{i}_p)} = \sum_{\mathbf{0}\le\mathbf{i}\le\mathbf{1}} P(\mathbf{Si})(-1)^{p-|\mathbf{i}|}. \tag{3.2}$$

Proof. Since the left hand side is a constant, integrating it over the p-dimensional unit cube doesn't change its value, thus

$$\frac{\partial^p P(\mathbf{Sz})}{\partial z_1\ldots\partial z_p} = \int_0^1\ldots\int_0^1 \frac{\partial^p P(\mathbf{Sz})}{\partial z_1\ldots\partial z_p}\mathrm{d}z_1\ldots\mathrm{d}z_p.$$

By integrating with respect to z_p, we obtain

$$\begin{aligned}\int_0^1\ldots\int_0^1 \frac{\partial^p P(\mathbf{Sz})}{\partial z_1\ldots\partial z_p}\mathrm{d}z_1\ldots\mathrm{d}z_p = \\ \int_0^1\ldots\int_0^1 \frac{\partial^{p-1} P(\mathbf{s_1}z_1+\ldots+\mathbf{s_{p-1}}z_{p-1}+1\cdot\mathbf{s_p})}{\partial z_1\ldots\partial z_{p-1}} - \\ \frac{\partial^{p-1} P(\mathbf{s_1}z_1+\ldots+\mathbf{s_{p-1}}z_{p-1}+0\cdot\mathbf{s_p})}{\partial z_1\ldots\partial z_{p-1}}\mathrm{d}z_1\ldots\mathrm{d}z_{p-1}.\end{aligned}$$

As we continue to integrate the expression with respect to the other variables, we arrive to the form on the right hand side of (3.2). □

Remark 3.2. Observe that (3.2) implies that in order to obtain the mixed derivative with respect to every z_i, we only need to know the values of the polynomial at the corners of the parallelepiped $\{\mathbf{Sz}: 0\le z_i\le 1\}$.

Now let us consider multiple derivations with respect to each z_i. More precisely, given a multi-index $\mathbf{i}$, we differentiate $\mathbf{i}_r$ times with respect to z_r for $r=1,\ldots,p$. The analogue of (3.2) for a polynomial P of degree at most $|\mathbf{i}|$ is

$$\frac{\partial^{|\mathbf{i}|} P(\mathbf{Sz})}{\partial z_1^{\mathbf{i}_1}\ldots\partial z_p^{\mathbf{i}_p}} = \sum_{\mathbf{k}_1=0}^{\mathbf{i}_1}\ldots\sum_{\mathbf{k}_p=0}^{\mathbf{i}_p}\binom{\mathbf{i}_1}{\mathbf{k}_1}\ldots\binom{\mathbf{i}_p}{\mathbf{k}_p}(-1)^{|\mathbf{i}-\mathbf{k}|}P(\mathbf{Sk}). \tag{3.3}$$

Substituting the binomial coefficient notation for multi-indices

$$\begin{pmatrix} \mathbf{i} \\ \mathbf{k} \end{pmatrix} = \begin{pmatrix} \mathbf{i}_1 \\ \mathbf{k}_1 \end{pmatrix} \cdots \begin{pmatrix} \mathbf{i}_p \\ \mathbf{k}_p \end{pmatrix},$$

into (3.3), we obtain the following Lemma.

Lemma 3.3. *Let $\mathbf{S} \in \mathbb{R}^{n\times p}$ be a seed matrix, $\mathbf{i} \in \mathbb{N}_0^p$ a multi-index and P a polynomial of degree at most $|\mathbf{i}|$. It holds that*

$$\frac{\partial^{|\mathbf{i}|} P(\mathbf{Sz})}{\partial z_1^{\mathbf{i}_1} \dots \partial z_p^{\mathbf{i}_n}} = \sum_{\mathbf{0}\le\mathbf{k}\le\mathbf{i}} \begin{pmatrix} \mathbf{i} \\ \mathbf{k} \end{pmatrix} (-1)^{|\mathbf{i}-\mathbf{k}|} P(\mathbf{Sk}).$$

3.3 Higher order derivatives of smooth functions

Let $F_k(\mathbf{x};\mathbf{v})$ denote the k-th Taylor coefficient of the univariate function

$$f_{\mathbf{x};\mathbf{v}} \colon \mathbb{R} \to \mathbb{R}, \quad t \mapsto f(\mathbf{x}+t\mathbf{v})$$

at $t = 0$ for the vectors $\mathbf{x}, \mathbf{v} \in \mathbb{R}^n$. The Taylor expansion for $f_{\mathbf{x};\mathbf{v}}$ is given by

$$f_{\mathbf{x};\mathbf{v}}(t) = f(\mathbf{x}+t\mathbf{v}) = F_0(\mathbf{x};\mathbf{v}) + F_1(\mathbf{x};\mathbf{v})t + \ldots + F_k(\mathbf{x};\mathbf{v})t^k + \ldots,$$

up to an order determined by the smoothness of f. By considering $\mathbf{x}$ as a constant and $\mathbf{v}$ as a variable, the function

$$\mathbf{v} \mapsto F_k(\mathbf{x};\mathbf{v}) = \frac{1}{k!}\frac{\mathrm{d}^k}{\mathrm{d}t^k} f(\mathbf{x}+t\mathbf{v})\bigg|_{t=0}$$

is a polynomial of degree k for $k \in \mathbb{N}_0$. Moreover, this polynomial is homogeneous, since $F_k(\mathbf{x};r\mathbf{v}) = r^k F_k(\mathbf{x};\mathbf{v})$ for $r \in \mathbb{R}$ and $k \in \mathbb{N}$.

Let $\mathbf{i} \in \mathbb{N}_0^p$ be a multi-index, $\mathbf{S} \in \mathbb{R}^{n\times p}$ a seed matrix and let $\mathbf{v} = \mathbf{Sz}$, where now we consider $\mathbf{z} \in \mathbb{R}^p$ as the variable vector. In the Taylor expansion for $f_{\mathbf{x};\mathbf{v}}$ of order $|\mathbf{i}|$

$$f(\mathbf{x}+\mathbf{Sz}) = f_{\mathbf{x};\mathbf{v}}(1) = \sum_{k=0}^{k=|\mathbf{i}|} F_k(\mathbf{x};\mathbf{Sz}) + \mathrm{Rem}_{1;|\mathbf{i}|+1}(f_{\mathbf{x};\mathbf{v}}, 0),$$

the remainder is an infinite polynomial of $\mathbf{z}$ consisting of monomials of degree at least $|\mathbf{i}|+1$. Therefore

$$\frac{\partial^{|\mathbf{i}|} f(\mathbf{x}+\mathbf{Sz})}{\partial z_1^{\mathbf{i}_1} \dots \partial z_p^{\mathbf{i}_p}}\bigg|_{\mathbf{z}=\mathbf{0}} = \frac{\partial^{|\mathbf{i}|} F_{|\mathbf{i}|}(\mathbf{x};\mathbf{Sz})}{\partial z_1^{\mathbf{i}_1} \dots \partial z_p^{\mathbf{i}_p}}. \tag{3.4}$$

Using Lemma 3.3, the homogeneity property and (3.4), we obtain the following.

Lemma 3.4. *Let $\mathbf{i} \in \mathbb{N}_0^p$ be a multi-index, $\mathbf{S} \in \mathbb{R}^{n\times p}$ a seed matrix and let $\mathbf{x} \in \mathbb{R}^n$.*

$$\frac{\partial^{|\mathbf{i}|} f(\mathbf{x}+\mathbf{Sz})}{\partial z_1^{\mathbf{i}_1} \dots \partial z_p^{\mathbf{i}_p}}\bigg|_{\mathbf{z}=\mathbf{0}} = \sum_{\mathbf{0}<\mathbf{k}\le\mathbf{i}} \begin{pmatrix} \mathbf{i} \\ \mathbf{k} \end{pmatrix} (-1)^{|\mathbf{i}-\mathbf{k}|} F_{|\mathbf{i}|}(\mathbf{x};\mathbf{Sk}).$$

3.4 Interpolating higher order derivatives

Lemma 3.4 provides a procedure to calculate $f_{\mathbf{i}}(\mathbf{x})$ (recall the notation from (3.1)) through interpolation from univariate Taylor coefficients. Our goal in this Section is to find the same value by using univariate Taylor coefficients in directions obtained as $\mathbf{Sj}$, where $|\mathbf{j}| = d$. In order to achieve this, we have to analyze the term $F_{|\mathbf{i}|}(\mathbf{x};\mathbf{Sk})$ on the right hand side of Lemma 3.4.

Let $\mathbf{k}$ be a multi-index satisfying $\mathbf{0} < \mathbf{k} \leq \mathbf{i}$ and $m \in \mathbb{N}$. Using the homogeneity of F_m, we get

$$F_m(\mathbf{x};\mathbf{Sk}) = \left(\frac{|\mathbf{k}|}{d}\right)^m F_m\left(\mathbf{x};\mathbf{S}\left(\frac{d}{|\mathbf{k}|}\mathbf{k}\right)\right). \tag{3.5}$$

Note that the vector $\mathbf{z}' = \frac{d}{|\mathbf{k}|}\mathbf{k} \in \mathbb{R}^p$ satisfies $z'_1 + \ldots + z'_p = d$.

Let us recall that $F_m(\mathbf{x};\mathbf{Sz})$ is a polynomial of degree m in $\mathbf{v} = \mathbf{Sz}$, therefore it is a polynomial of degree m in $\mathbf{z}$ as well. Note that $\begin{pmatrix} \mathbf{z} \\ \mathbf{j} \end{pmatrix} F_m(\mathbf{x};\mathbf{Sj})$ is a polynomial of degree $|\mathbf{j}|$ in $\mathbf{z}$. If we assume that $\mathbf{z} \in \mathbb{N}_0^p$ is a multi-index such that $|\mathbf{z}| = |\mathbf{j}|$, then the expression $\begin{pmatrix} \mathbf{z} \\ \mathbf{j} \end{pmatrix} F_m(\mathbf{x};\mathbf{Sj})$ is nonzero if and only if $\mathbf{z} = \mathbf{j}$ and in that case we obtain $\begin{pmatrix} \mathbf{j} \\ \mathbf{j} \end{pmatrix} F_m(\mathbf{x};\mathbf{Sj}) = F_m(\mathbf{x};\mathbf{Sj})$. These considerations lead to the formula

$$F_m(\mathbf{x};\mathbf{Sz}) = \sum_{|\mathbf{j}|=d} \begin{pmatrix} \mathbf{z} \\ \mathbf{j} \end{pmatrix} F_m(\mathbf{x};\mathbf{Sj}), \tag{3.6}$$

where $\mathbf{z} \in \mathbb{R}^p$ is an arbitrary real vector satisfying $z_1 + \ldots + z_p = d$ and $m \leq d$. To see that (3.6) holds, note that both sides are polynomial in $\mathbf{z}$ and equality is satisfied if $\mathbf{z} = \mathbf{j}$ for all multi-indices $|\mathbf{j}| = d$. As a consequence of (3.5) and (3.6), we obtain

$$F_{|\mathbf{i}|}(\mathbf{x};\mathbf{Sk}) = \sum_{|\mathbf{j}|=d} \left(\frac{|\mathbf{k}|}{d}\right)^{|\mathbf{i}|} \begin{pmatrix} d\mathbf{k}/|\mathbf{k}| \\ \mathbf{j} \end{pmatrix} F_{|\mathbf{i}|}(\mathbf{x};\mathbf{Sj}), \tag{3.7}$$

for all multi-indices $\mathbf{k}$ satisfying $\mathbf{0} < \mathbf{k} \leq \mathbf{i}$ and $1 \leq |\mathbf{i}| \leq d$.

We define the quantities $\gamma(\mathbf{i},\mathbf{j})$, where $\mathbf{i} \in \mathbb{N}_0^p$ and $\mathbf{j} \in \mathbb{N}_0^p$ are multi-indices, as follows

$$\gamma(\mathbf{i},\mathbf{j}) = \sum_{\mathbf{0}<\mathbf{k}\leq\mathbf{i}} (-1)^{|\mathbf{i}-\mathbf{k}|} \begin{pmatrix} \mathbf{i} \\ \mathbf{k} \end{pmatrix} \left(\frac{|\mathbf{k}|}{|\mathbf{j}|}\right)^{|\mathbf{i}|} \begin{pmatrix} |\mathbf{j}|\mathbf{k}/|\mathbf{k}| \\ \mathbf{j} \end{pmatrix}. \tag{3.8}$$

Lemma 3.4 together with (3.7) and (3.8) gives us the following Theorem.

Theorem 3.5. *Fix $d \in \mathbb{N}$. Let $f\colon \mathbb{R}^n \to \mathbb{R}$ be at least d-times continuously differentiable function at the point $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{S} \in \mathbb{R}^{n\times p}$ be a seed matrix. Then for all multi-indices $\mathbf{i} \in \mathbb{N}_0^p$ with $1 \leq |\mathbf{i}| \leq d$, the partial derivative $f_{\mathbf{i}}(\mathbf{x}) = \left.\frac{\partial^{|\mathbf{i}|} f(\mathbf{x}+z_1\mathbf{s_1}+\ldots+z_p\mathbf{s_p})}{\partial z_1^{\mathbf{i}_1}\ldots\partial z_p^{\mathbf{i}_p}}\right|_{\mathbf{z}=\mathbf{0}}$ is given by*

$$\left.\frac{\partial^{|\mathbf{i}|} f(\mathbf{x}+z_1\mathbf{s_1}+\ldots+z_p\mathbf{s_p})}{\partial z_1^{\mathbf{i}_1}\ldots\partial z_p^{\mathbf{i}_p}}\right|_{\mathbf{z}=\mathbf{0}} = \sum_{|\mathbf{j}|=d} \gamma(\mathbf{i},\mathbf{j}) F_{|\mathbf{i}|}(\mathbf{x};\mathbf{Sj}). \tag{3.9}$$

3.5 The coefficients $\gamma(\mathbf{i},\mathbf{j})$

It is also shown in Griewank, Utke and Walther [17] that for $\mathbf{i},\mathbf{j} \in \mathbb{N}_0^p$, $1 \le |\mathbf{i}| \le d$ and $|\mathbf{j}| = d$, the number of nonvanishing coefficients $\gamma(\mathbf{i},\mathbf{j}) \neq 0$ is less than or equal to

$$\#(d,p) = \sum_{m=1}^{d} \left(\begin{array}{c} p \\ m \end{array} \right) \left(\begin{array}{c} d \\ m \end{array} \right) \left(\begin{array}{c} m+d-1 \\ d \end{array} \right),$$

Note that upon applying the formula (3.9) for different vectors $\mathbf{x} \in \mathbb{R}^n$, it is recommended to precompute the values $\gamma(\mathbf{i},\mathbf{j})$ in advance, possibly with higher accuracy.

Chapter 4

Graph representations of maps

Different directed graphs can be associated with a given map. These graphs reflect the behavior of the map up to a given resolution. The vertices of these graphs are sets and the edges correspond to transitions between them. We can derive properties of our dynamical system through the study of the graphs. These techniques appeared in many articles, in both rigorous and non-rigorous computations, for example by Hohmann and Dellnitz [12], Hohmann, Dellnitz, Junge and Rumpf [13], Galias [15], Luzzatto and Pilarczyk [25], and computations for the time evolution of a continuous system with a given timestep by Wilczak [37], without attempting to be comprehensive. We summarize the main algorithms and give a uniform framework in this Chapter. The procedure for enclosing non-wandering points was described in [15] in a similar setting, however without the proof of its correctness. This we give in Paper B together with an application, proving a conjecture for the 2-dimensional Ricker map (see Ricker [31], Levin and May [23]). In Paper C, we use the same method and give a necessary and sufficient condition for global asymptotic stability of the fixed point of a certain class of delay difference equations.

In Section 4.1 we introduce *graph representations* of maps. We discuss several *enclosure algorithms* that use these representations in Section 4.2. We comment on their *convergence* properties in Section 4.3. A more efficient scheme for enclosing *fixed points* and an algorithm for the *inner enclosure of the basin of attraction* are given in Sections 4.4 and 4.5, respectively. In Section 4.6, we comment on the analysis of two *topological properties*, *transitivity* and *mixing*. We finish this Chapter with an Appendix that contains some simple and well known algorithms for directed graphs.

4.1 Covers and graph representations

Definition 4.1. $\mathcal{S}$ is called a *cover* of $\mathcal{D} \subseteq \mathbb{R}^n$ if it is a collection of subsets of $\mathbb{R}^n$ such that $\cup_{s\in\mathcal{S}} s \supseteq \mathcal{D}$. We denote the closure of their union relative to $\mathcal{D}$ by

$$|\mathcal{S}| = \mathrm{cl}\left(\bigcup_{s\in\mathcal{S}} s\right) \cap \mathcal{D}$$

in the following. We define the *diameter* or *outer resolution* of the cover $\mathcal{S}$ by

$$\mathcal{R}^+(\mathcal{S}) = \operatorname{diam}(\mathcal{S}) = \sup_{s\in\mathcal{S}} \operatorname{diam}(s),$$

where

$$\operatorname{diam}(s) = \sup_{x,y\in s} \|x-y\|.$$

A cover $\mathcal{S}_2$ is said to be *finer* than the cover $\mathcal{S}_1$ if

$$(\forall s_1 \in \mathcal{S}_1)\ (\exists\{s_{2,i}, i\in\mathcal{I}\}\subseteq\mathcal{S}_2)\ \text{ such that } \bigcup_{i\in\mathcal{I}} s_{2,i} = s_1.$$

We denote this relation by $\mathcal{S}_2 \preccurlyeq \mathcal{S}_1$. The *inner resolution* of a cover $\mathcal{S}$ is the following:

$$\mathcal{R}^-(\mathcal{S}) = \sup\{r\geq 0 : \forall x\in\mathcal{D}, \exists s\in\mathcal{S} : \mathrm{B}(x;r)\subseteq s\}.$$

We mean by $\mathrm{B}(x;r)$ the open ball with radius r around x in the Euclidean-norm. A cover $\mathcal{S}$ is *essential* if $\mathcal{S}\setminus s$ is not a cover anymore for all $s\in\mathcal{S}$. The cover $\mathcal{S}$ is called an *open cover* if all elements are open subsets of $\mathbb{R}^n$. The cover $\mathcal{P}$ is called a *partition* if it consists of closed sets such that $|\mathcal{P}| = \mathcal{D}$ and $\forall p_1, p_2 \in \mathcal{P} : p_1\cap p_2 \subseteq \operatorname{bd}(p_1)\cup\operatorname{bd}(p_2)$, where $\operatorname{bd}(p)$ is the boundary of the set p. Consequently, for a partition $\mathcal{P}$ the inner resolution $\mathcal{R}^-(\mathcal{P})$ is zero.

In the following we will always work with essential and finite covers that are open covers or partitions. As a consequence, the supremum in the definition of the diameter $\mathcal{R}^+(\mathcal{S})$ becomes a maximum.

Definition 4.2. A *directed graph* $\mathcal{G} = \mathcal{G}(\mathcal{V},\mathcal{E})$ is a pair of sets representing the vertices $\mathcal{V}$ and the edges $\mathcal{E}$, that is: $\mathcal{E}\subseteq\mathcal{V}\times\mathcal{V}$, and $(u,v)\in\mathcal{E}$ means that $\mathcal{G}$ has a directed edge going from u to v. We say that $v_1\to v_2\to\ldots\to v_k$ is a *directed path* if $(v_i,v_{i+1})\in\mathcal{E}$ for all $i=1,\ldots,k-1$. If $v_k = v_1$, then it is a *directed cycle*. If the greatest common divisor of the lengths of all the directed cycles in the graph is 1, then $\mathcal{G}$ is called *aperiodic*.

A directed graph $\mathcal{G}$ is *strongly connected* if for all $u,v\in\mathcal{V}$, $v\neq u$ there is a directed path from u to v and from v to u as well. The *Strongly Connected Components (SCC)* of a directed graph $\mathcal{G}$ are its maximal strongly connected subgraphs. It is easy to see that u and v are in the same SCC if and only if there is a directed cycle going through both u and v. Every directed graph $\mathcal{G}$ can be decomposed into the union of strongly connected components and directed paths between them. If we contract each SCC to a new vertex, we obtain a directed acyclic graph, that is called the *condensation* of $\mathcal{G}$.

Definition 4.3. Let $f : \mathcal{D}_f\subseteq\mathbb{R}^n\to\mathbb{R}^n$, $\mathcal{D}\subseteq\mathcal{D}_f$ and $\mathcal{S}$ be a cover of $\mathcal{D}$. We say that the directed graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ is a *graph representation of f on $\mathcal{D}$ with respect to* $\mathcal{S}$, if there is a $\imath : \mathcal{V}\to\mathcal{S}$ bijection such that the following implication is true for all $u,v\in\mathcal{V}$:

$$f(\imath(u)\cap\mathcal{D})\cap\imath(v)\cap\mathcal{D}\neq\emptyset \Rightarrow (u,v)\in\mathcal{E},$$

and we denote it by $\mathcal{G}\propto(f,\mathcal{D},\mathcal{S})$.

Having a graph representation $\mathcal{G}$ of f on $\mathcal{D}$ with respect to $\mathcal{S}$, we take the liberty to handle the elements of the cover as vertices and vice versa, omitting the usage of ι. It is important to emphasize that in general $(u,v) \in \mathcal{E}$ does not imply that $f(u \cap \mathcal{D}) \cap v \cap \mathcal{D} \neq \emptyset$. If we have $(u,v) \in \mathcal{E} \Leftrightarrow f(u \cap \mathcal{D}) \cap v \cap \mathcal{D} \neq \emptyset$, then we call $\mathcal{G}$ an *exact* graph representation.

Example 4.4. *In order to easily understand what a graph representation is, consider a two-dimensional map and a compact set on the plane. Create a mesh on the set by dividing into smaller parts, these are the vertices of $\mathcal{G}$. We obtain the directed edges by drawing an arrow from one set into another. An edge must be drawn if there is an orbit corresponding to it. If only such edges are drawn, the representation is exact. See Figure 4.1.*

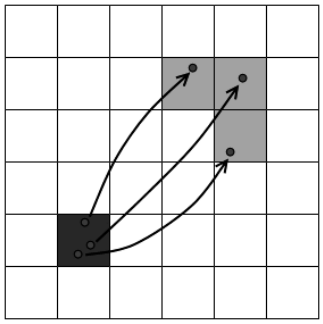

Figure 4.1: The edges starting from a vertex are induced by the function.

4.2 Enclosure algorithms

Instead of directly studying the map (2.1), we may derive conclusions through analyzing different graph representations of f. Let $K \subseteq \mathcal{D}_f$ be a compact set, $\mathcal{S}$ a cover of K and $\mathcal{G}$ a graph representation of f with respect to $\mathcal{S}$. We summarize some trivial statements in the following theorem.

Theorem 4.5. *Let $s \in \mathcal{S}$ be an element of the cover. Then it holds that*

a) if $\mathrm{Fix}(f;K) \cap s \neq \emptyset$, *then* $(s,s) \in \mathcal{E}$,

b) if $\mathrm{Per}_{\leq m}(f;K) \cap s \neq \emptyset$, *then there is a directed cycle in $\mathcal{G}$ that contains s and the cycle is of length at most m,*

c) if $\mathcal{A}_K \cap s \neq \emptyset$, *then there exists $r \in \mathcal{S}$ such that $(r,s) \in \mathcal{E}$,*

d) if $\mathrm{Inv}(f;K) \cap s \neq \emptyset$, *then there are $r,t \in \mathcal{S}$ satisfying $(r,s),(s,t) \in \mathcal{E}$.*

These are immediate consequences of the definitions. Now we present a general algorithm that will enclose the formerly mentioned objects, depending on the choice of the property P. Assume that we have implemented the following functions already:

1. `Cover`(K,δ_0) returns a cover of the compact set K such that the a diameter of this cover is not larger than δ_0.

2. `Transitions`$(\mathcal{V},f)$ returns the possible transitions, induced by f in $\mathcal{D}_f$, between the elements of the cover $\mathcal{V}$.

Algorithm 1 General enclosure algorithm

```
1:  procedure GENERAL_ENCLOSURE_ALGORITHM(f, K, δ_0; P)
2:      k ← 0
3:      V_0 ← Cover(K, δ_0)                          ▷ V_0 is a cover of K, diam(V_0) ≤ δ_0.
4:      loop
5:          E_k ← Transitions(V_k, f)        ▷ The possible transitions (extra edges may occur).
6:          G_k ← GRAPH(V_k, E_k)                                  ▷ G_k ∝ (f, |V_k|, V_k)
7:          ready ← TRUE
8:          repeat
9:              for all v ∈ V_k do
10:                 if v does not have property P then
11:                     remove v from G_k
12:                     ready ← FALSE
13:                 end if
14:             end for
15:         until ready
16:         if STOP(k, V_k, ε_k) then
17:             return V_k
18:         else
19:             δ_{k+1} ← δ_k/2
20:             V_{k+1} ← Cover(|V_k|, δ_{k+1})      ▷ V_{k+1} is a cover of |V_k|, diam(V_{k+1}) ≤ δ_{k+1}.
21:             k ← k+1
22:         end if
23:     end loop
24: end procedure
```

We start the algorithm with the map f, the initial compact region $K \subseteq \mathcal{D}_f$, the initial maximal resolution δ_0 and the property P. In line 3 we construct the initial cover of K, then in lines 5 and 6 the corresponding graph representation. The cycle starting from line 9 removes all the vertices from the graph that does not possess the property P. This is then repeated until no more vertices can be removed – removing a vertex v can make another vertex u loose the property P. We return the remaining elements of the cover, if any, when a certain stopping condition is satisfied in line 16; for example if $\delta_k < \Delta$, where $\Delta > 0$ is a small positive number given in advance. Otherwise, we take a cover with a smaller diameter of the remaining covered compact part of K and repeat the process.

As a result, Algorithm 1 generates tighter and tighter enclosures of a certain object depending on the property P. For example, if having the property P for a vertex v means

that $(v,v) \in \mathcal{E}$, then we get an enclosure of the fixed points. If P is given by v having both in- and out-edges, then we get an enclosure of the invariant set.

We may use the same algorithm to enclose the non-wandering points with open covers and partitions. The appropriate property P shall be that v is in a directed cycle. The proof of this statement for partitions can be found in Paper B. The proof for open covers is analogous with the trivial part of the proof in Paper B, since every point is contained in the interior of some cover element.

Remark 4.6. Though using the property $P =$ 'v is in a directed cycle' will result in an enclosure of the non-wandering points, it is not guaranteed that every partition element that contains a non-wandering point is kept.

Table 4.1 collects the discussed *objects* and the corresponding *properties*.

Table 4.1: Objects and corresponding properties.

i	Object ($\mathcal{O}_i$)	Vertex property (P_i)
1	$\mathrm{Fix}(f;K)$	$(s,s) \in \mathcal{E}$
2	$\mathrm{Per}_{\leq m}(f;K)$	there is a directed cycle in $\mathcal{G}$ containing s and of length at most m
3	$\mathcal{A}_K$	there exists $r \in \mathcal{S}$ such that $(r,s) \in \mathcal{E}$
4	$\mathrm{Inv}(f;K)$	there are $r,t \in \mathcal{S}$ such that $(r,s),(s,t) \in \mathcal{E}$
5	$\mathrm{NonW}(f;K)$	s is in a directed cycle

4.3 Convergence

During the entire general algorithm, $|\mathcal{V}_k|$ is a closed, compact set. Suppose that we never stop and let $\delta_k \to 0$. Then, $|\mathcal{V}_k|$ is a nested sequence of closed and compact sets, thus we may define

$$\mathcal{V}_\infty = \bigcap_{k \in \mathbb{N}_0} |\mathcal{V}_k|.$$

From Theorem 4.5, by using the notations of Table 4.1 and considering $\mathcal{V}_k$ given by Algorithm 1, we obtain Theorem 4.7.

Theorem 4.7. *If we apply the – non stopping – Algorithm 1 to K with P_i as a property, then $\mathcal{O}_i \subseteq \mathcal{V}_\infty$.*

Assume that we work with as exact graph representations as we can get; overestimations due to the usage of intervals may occur (recall Sections 1.2 and 2.1.1). The natural question is the following. Is it true that $\lim_{k\to\infty} \mathcal{V}_k = \mathcal{O}_i$, that is $\mathcal{V}_\infty = \mathcal{O}_i$, or not? We shall show that for $i = 1,\ldots,4$, this is satisfied in the ideal case, when every graph representation created during the Algorithm is exact. Thus, the subsequent enclosures of the fixed points, periodic points, relative attractor or the invariant set are converging to the corresponding object. We will establish this result through a series of lemmata.

Lemma 4.8. *Let $m \in \mathbb{Z}^+$ and $\varepsilon > 0$. There exists a $0 < \delta = \delta(\varepsilon,m) < \varepsilon$ such that for all finite series of points $x_0,x_1,\ldots,x_m \in K$ satisfying $x_{k+1} = f(x_k)$, for all **exact** graph*

representations $\mathcal{G} \propto (f, K, \mathcal{S})$ *with* $diam(\mathcal{S}) < \delta$ *and for any series of cover elements* $\{s_0, \ldots, s_l\} \subseteq \mathcal{S}$ *that satisfies*

1. $0 \leq l \leq m$
2. $x_0 \in s_0$,
3. *if* $l > 1$, *then* $s_0 \to \ldots \to s_l$ *is a directed path in* $\mathcal{G}$,

it is true that $s_k \subseteq \mathrm{B}(x_k; \varepsilon)$, *for* $k = 0, \ldots, l$.

Proof. The case $l = 0$ is trivial. For $l > 0$ we proceed as follows. f is continuous on the compact K, therefore it is uniformly continuous. We will prove the claim by induction, consider that $m = 1$:

Fix $\varepsilon > 0$, there exists $0 < \delta < \frac{\varepsilon}{2}$, such that if $\|x - y\|_2 < \delta$, then $\|f(x) - f(y)\|_2 < \frac{\varepsilon}{2}$, for any $x, y \in K$. Assume now that $\mathrm{diam}(\mathcal{S}) < \delta$ and $x_0, x_1 \in K$, $s_0, s_1 \in \mathcal{S}$ are as described above. We have $x_0 \in s_0 \subseteq \mathrm{B}(x_0; \delta) \subseteq \mathrm{B}(x_0; \varepsilon)$ and

$$f(s_0) \subseteq \mathrm{B}\left(x_1; \frac{\varepsilon}{2}\right),$$
$$s_1 \subseteq \mathrm{B}(f(s_0); \delta),$$

therefore

$$s_1 \subseteq \mathrm{B}\left(x_1; \frac{\varepsilon}{2} + \delta\right) \subseteq \mathrm{B}(x_1; \varepsilon).$$

Now assume that the statement is true for $1 \leq k \leq m$ and consider it for $m + 1$. Let $\varepsilon > 0$ and obtain $\delta(\varepsilon, 1) =: \delta_1 < \varepsilon$. Now, by induction, we get

$$\delta(\delta_1, l - 1) =: \delta_2 < \delta_1 < \varepsilon.$$

It is easy to see that δ_2 satisfies the requirements. □

Remark 4.9. That means that any m-orbit of the system and any m-path in the graph, where the first point of the orbit is contained in the first vertex of the path, are 'ε-close' if the resolution is small enough.

Lemma 4.10. *Let* $\mathcal{G} \propto (f, K, \mathcal{S})$ *be an exact representation and fix* $x \in K$.

1. *If* $f(x) \notin K$, *then there exists a* $\delta > 0$ *such that if* $\mathrm{diam}(\mathcal{S}) < \delta$, *then for any* $s \in \mathcal{S}$ *such that* $x \in s$, *there is no edge leaving* s.
2. *If* $f^{-1}(x) \cap K$ *is empty, then there exists a* $\delta > 0$ *such that if* $\mathrm{diam}(\mathcal{S}) < \delta$, *then for any* $s \in \mathcal{S}$ *such that* $x \in s$, *there is no edge incoming to* s.

Proof. In the first case choose $\varepsilon > 0$ such that $\|f(x) - K\|_2 > \varepsilon$. This is possible, since K is compact and f is continuous. Now exists is a $\delta \in (0, \frac{\varepsilon}{4})$ such that if $\|x - y\|_2 < \delta$, then $\|f(x) - f(y)\|_2 < \frac{\varepsilon}{4}$. Therefore

$$f(y) \notin \mathrm{B}\left(K; \frac{\varepsilon}{4}\right).$$

This implies that if $diam(\mathcal{S}) < \delta$, then for any $s \in \mathcal{S}$ such that $x \in s$, the set $f(s) \cap |\mathcal{S}|$ is empty. In other words, there is no outgoing edge from s.

In the second case, again because of the continuity of f, there exists an $\varepsilon > 0$ such that $f|_K$ has no inverse in $\mathrm{B}(x;\varepsilon)$. If $diam(\mathcal{S}) < \varepsilon$, then any $s \in \mathcal{S}$ such that $x \in \mathcal{S}$, will have no incoming edge, since every edge represents at least one real orbit because $\mathcal{G}$ is an exact representation. □

Lemma 4.11. *If we work with exact representations and run our – non stopping – Algorithm 1 with the property P_i and the compact set K, then we obtain $\mathcal{V}_\infty = \mathcal{O}_i$, for $i = 1, 2$.*

Proof. It is enough to show it for $i = 2$, that is for periodic orbits. Let $x \in K \setminus \mathrm{Per}_{\leq m}(f;K)$ and define

$$\varepsilon = \min\left\{ \|f^k(x) - f^j(x)\|_2 : k \neq j \text{ and } k, j = 0, \ldots, m \right\}.$$

Since $x \notin \mathrm{Per}_{\leq m}(f;K)$, ε is positive. By Lemma 4.8 we obtain a $\delta(\frac{\varepsilon}{2}, m)$ such that if $\mathrm{diam}(\mathcal{S}) < \delta$, then any path, starting from a vertex containing x and of length l not greater than m, will follow the first l iterates of x, not being further than $\frac{\varepsilon}{2}$ from the actual value in any step. This means that it cannot form a directed cycle through the starting vertex. Thus, if $\mathrm{diam}(\mathcal{S}) < \delta$, then every $s \in \mathcal{S}$ such that $x \in s$ will be removed by our algorithm, therefore $x \notin \mathcal{V}_\infty$. □

Lemma 4.12. *If we work with exact representations and run our – non stopping – Algorithm 1 with the property P_4 and the compact set K, then we obtain $\mathcal{V}_\infty = \mathcal{O}_4$.*

Proof. Let $x \notin \mathrm{Inv}(f;K)$. This means that there exists $k \in \mathbb{Z}^+$ such that

$$\begin{aligned} &f^{k-1}(x) \in K, && \text{but } f^k(x) \notin K, \text{ or} \\ &f^{-k+1}(x) \cap K \neq \emptyset, && \text{but } f^{-k}(x) \cap K = \emptyset. \end{aligned}$$

From Lemma 4.10 we know that if the diameter of the cover is small enough, every vertex containing $f^{k-1}(x)$ is a sink in the first case or if we are in the second case, then every vertex containing $f^{-k+1}(x)$ is a source. Therefore, these vertices will be removed by Algorithm 1 after a finite number of steps. Suppose that this happens when the cover is given as $\mathcal{V}_m$.

Now we may repeat the argument for x, with the compact $|\mathcal{V}_m|$ taking the role of K. We get a $k' < k$ such that analogous equations hold with k' as above. We obtain by induction that every box containing any iterate of x will be removed after a finite number of steps, thus $x \notin \mathcal{V}_\infty$. □

Let us turn our attention to the global attractor relative to K. The following argument was presented by Hohmann and Dellnitz [12].

Lemma 4.13. *If we work with exact representations and run our – non stopping – Algorithm 1 with the property P_3 and the compact set K, then the obtained set $\mathcal{V}_\infty$ is backward invariant in K, that is, for all $x \in \mathcal{V}_\infty$ there exists $y \in \mathcal{V}_\infty$ such that $f(x) = y$.*

Proof. If there exists $k \in \mathbb{Z}^+$ such that $f^{-k+1}(x) \cap K \neq \emptyset$ but $f^k(x) \cap K = \emptyset$, then, by a similar argument as in Lemma 4.12, any vertex containing $f^{-k+1}(x)$ will be removed when the resolution is small enough. Using induction, we obtain that in finite number of steps, every vertex that contains a non-positive iterate of x will be removed as well. If such k does not exist, then $f^{-m}(x) \subseteq \mathcal{V}_\infty$, for all $m \in \mathbb{N}$. □

Lemma 4.14. *Any set B that is backward invariant in K is contained in $\mathcal{A}_K$.*

Proof. We have $B \subseteq K$ and $B \subseteq f(B)$, therefore

$$B \subseteq f^k(B) \tag{4.1}$$

for $k \in \mathbb{Z}^+$. Let $\mathcal{A}$ be the global attractor of (2.1) and consider its fundamental neighbourhood $\mathcal{U}$ from the definition. Recall that $\mathcal{U}$ satisfies

$$\bigcup_{k\in\mathbb{N}} f^{-k}(\mathcal{U}) = \mathcal{D}_f.$$

Since K is compact, therefore there exists $k_0 \in \mathbb{N}_0$ such that

$$B \subseteq K \subseteq \bigcup_{0\leq k\leq k_0} f^{-k}(\mathcal{U}).$$

This leads to $f^{k_0}(B) \subseteq \mathcal{U}$. Equation (4.1) implies that $B \subseteq f^{k_0+k}(B) \subseteq f^k(\mathcal{U})$ for all $k \in \mathbb{N}$, thus

$$B \subseteq \bigcap_{k\in\mathbb{N}} f^k(\mathcal{U}) = \mathcal{A}.$$

Together with the backward invariance in K, this implies that $B \subseteq \mathcal{A}_K$. □

We may summarize Theorem 4.7 and Lemmata 4.11, 4.12, 4.13, 4.14 as:

Theorem 4.15. *If we work with exact representations and run our – non stopping – Algorithm 1 with the property P_i and the compact set K, then for $i = 1, \ldots, 4$, we obtain $\mathcal{V}_\infty = \mathcal{O}_i$.*

4.4 Fixed points, periodic orbits

We have seen a convergent enclosure procedure for periodic orbits and fixed points. By another approach, we transform the task of enclosing fixed points and periodic orbits into enclosing the zeros of a certain function. The fixed point equation $f(x) = x$ may be reformulated as $f(x) - x = 0$ and finding a periodic orbit of period m is equivalent to finding zeros of $f^m(x) - x$. If we apply the bisection method for these reformulated problems – see for example in Moore [27] or Tucker [36] – we obtain the same procedure essentially that was described with graph representations. On the other hand, we may apply superior zero-finding techniques such as the Newton-method or the Krawczyk-method. These give faster convergence and the possibility to prove uniqueness. As seen

in Galias [15], we may gain additional speed by the following construction in the case of periodic points.

Let the function $F\colon (\mathbb{R}^n)^m \to (\mathbb{R}^n)^m$, $z \mapsto F(z)$ be given as

$$F_k(x_0, x_2, \ldots, x_{m-1}) = x_{(k+1) \bmod m} - f(x_k).$$

It is clear that a zero of F corresponds to an m-periodic orbit of f. Instead of using the general Krawczyk-method, we modify it as follows. When we do the bisection, we exploit that we are not really searching for a zero of a function in $n \times m$ dimension, but for an n-dimensional periodic point. This is summarized in Algorithm 2.

Algorithm 2 Find periodic orbits

```
1:  procedure FIND_PERIODIC_ORBITS(f, x, m; V)
2:      x_0 ← x                                   ▷ We search for a periodic point in x.
3:      for i = 1 to m − 1 do
4:          x_i ← f(x_{i−1})                      ▷ We find the orbit of x.
5:      end for
6:      z ← (x_0, ..., x_{m−1})                   ▷ The orbit is transformed into the new variable z.
7:      if Krawczyk(z)_F ⊂ z then                 ▷ z contains a unique fixed point.
8:          V ← V ∪ {x}
9:          return
10:     else if Krawczyk(z)_F ∩ z = ∅ then        ▷ z contains no fixed point.
11:         return
12:     end if
13:     divide x into {y_i}                       ▷ Otherwise we subdivide x ∈ R^n and not z ∈ R^{n×m}
14:     for all i do
15:         Find_Periodic_Orbits(f, y_i, m; Q, V)  ▷ The recursive call for the new regions.
16:     end for
17: end procedure
```

Here $\text{Krawczyk}(z)_F$ is the Krawczyk operator corresponding to F applied to the set z

$$\text{Krawczyk}(z)_F = \check{z} - F(\check{z})DF(\check{z})^{-1} - \left(1 - DF(z)DF(\check{z})^{-1}\right)[-\text{rad}(z), \text{rad}(z)],$$

where $\check{z} \in z$. We remark that $\check{z}$ is usually chosen to be the midpoint. Note that lines 7 to 16 give us the Krawczyk algorithm (see Galias [15]), the only difference is in the subdivision.

The elements of V are sets, each of them contains exactly one zero of F, that is one m-periodic point of f. Since every periodic point is a non-wandering point as well, we may first use our procedure to enclose the non-wandering points in $\mathcal{D}$ and then use the resulting enclosure as a starting set for our search.

4.5 Inner enclosure of the basin of attraction

Assume now that $\mathcal{O}$ is an attracting invariant set for (2.1) restricted to the compact K. Thus, there exists a neighbourhood U such that $\mathcal{O} \subseteq U \subseteq K$ and U is contained in the

basin of attraction of $\mathcal{O}$. We want to find a – possibly – larger set B, that is still inside the basin of attraction of $\mathcal{O}$.

We will use the following algorithm from Galias [15]. We consider a cover of K and the empty list W. We shall collect into W such elements that are inside the basin of attraction of $\mathcal{O}$. In practice, this means that a vertex is moved from the actual cover to W if it is inside or mapped into U or the other elements of W. We refine our remaining cover to have diameter half as before and repeat the procedure. Since in the beginning W was empty, it will only contain sets that are inside the basin of attraction of $\mathcal{O}$. Thus, after each cycle, $|W|$ is an *inner* enclosure for the basin of attraction of $\mathcal{O}$. We stop the iteration if a certain stopping condition is satisfied; for example $\delta_k < \Delta$, where Δ is a small positive number given in advance.

Algorithm 3 Inner enclosure of the basin of attraction

procedure BASIN_OF_ATTRACTION($f, K, \delta_0; U$) ▷ U is attracted by $\mathcal{O}$.
  $k \leftarrow 0$
  $W \leftarrow \emptyset$ ▷ We collect the vertices in the basin of attraction into W.
  $\mathcal{V}_0 \leftarrow$ `Cover`(K, δ_0) ▷ $\mathcal{V}_0$ is a cover of K, $\mathrm{diam}(\mathcal{V}_0) \le \delta_0$.
  loop
    $\mathcal{E}_k \leftarrow$ `Transitions`$(\mathcal{V}_k \cup W, f)$ ▷ The possible transitions (extra edges may occur).
    $\mathcal{G}_k \leftarrow$ `GRAPH`$(\mathcal{V}_k \cup W, \mathcal{E}_k)$ ▷ $\mathcal{G}_k \propto (f, |\mathcal{V}_k \cup W|, \mathcal{V}_k \cup W)$
    repeat
      ready $\leftarrow$ TRUE
      for all $v \in \mathcal{V}_k$ **do**
        if $v \subseteq U \cup |W|$ **or** $f(v) \subseteq U \cup |W|$ **then**
          move v from $\mathcal{V}_k$ to W ▷ v is attracted by $\mathcal{O}$.
          ready $\leftarrow$ FALSE
        end if
      end for
    until ready ▷ The remaining vertices are not attracted at this resolution.
    if `STOP`$(k, \mathcal{V}_k, W, \delta_k)$ **then** ▷ Some stopping condition.
      return W
    end if
    $\delta_{k+1} \leftarrow \delta_k/2$
    $\mathcal{V}_{k+1} \leftarrow$ `Cover`$(|\mathcal{V}_k|, \delta_{k+1})$ ▷ $\mathcal{V}_{k+1}$ is a cover of $|\mathcal{V}_k|$, $\mathrm{diam}(\mathcal{V}_{k+1}) \le \delta_{k+1}$.
    $k \leftarrow k+1$
  end loop
end procedure

4.6 Topological transitivity and mixing

We may analyze topological properties of maps using similar techniques as presented so far. We give the definitions and a brief discussion on how to check if the map satisfies these properties as seen in Luzzatto and Pilarczyk [25]. The corresponding simple and well known algorithms are included as an Appendix after this Section.

Definition 4.16. The map f is called *topologically transitive on* $\mathcal{D}$ if for all open sets $U, V \subseteq \mathcal{D}$, there exists a $k = k(U,V) \geq 0$ such that $f^k(U) \cap V \neq \emptyset$. The map is called *topologically mixing*, if the k in the former definition is independent of V.

Remark 4.17. A map that is mixing, is transitive as well.

Consider graph representations of f on $\mathcal{D}$ with respect to various covers. We shall formulate *necessary conditions* for the representations. If they are not satisfied, then the map cannot possess the corresponding topological properties.

Theorem 4.18. *Let $\mathcal{G} \propto (f, \mathcal{D}, \mathcal{S})$. If f is topologically transitive on $\mathcal{D}$, then $\mathcal{G}$ is a strongly connected graph. If f is topologically mixing, then $\mathcal{G}$ is strongly connected and aperiodic.*

Proof. If f is transitive, then taking any $s_1, s_2 \in \mathcal{S}$, there exists a $k \in \mathbb{N}$ such that $f^k(s_1) \cap s_2 \neq \emptyset$. Thus, $\mathcal{G}$ must contain a directed route from s_1 to s_2. Since these two vertices were chosen arbitrarily, the graph is strongly connected.

If f is mixing, then for any $s \in \mathcal{S}$, there is $k \in \mathbb{N}$ such that

$$(s \cap \mathcal{D}) \subseteq (|\mathcal{S}| \cap \mathcal{D}) \subseteq f^k(s)$$

holds. Since a mixing map is transitive, which in turn implies that $\mathcal{G}$ is strongly connected, we obtain that this is true for $f^{k+1}(s)$ as well. In particular, this shows that there are directed cycles of length k and $k+1$ through s, therefore $\mathcal{G}$ is aperiodic. □

Appendix : Graph Algorithms

Tarjan's Algorithm

In order to implement some of the methods presented in Chapter 4, we need algorithms to find the strongly connected components and the period of a directed graph $\mathcal{G}$. For the first problem, we will use the Algorithm by Tarjan [34].

Algorithm 4 Tarjan's algorithm / I

```
1: procedure TARJAN(G = (V, E))                    ▷ The main procedure.
2:     global index ← 0                            ▷ This holds the lowest unused index.
3:     global stack T ← empty   ▷ The stack T holds the visited but not categorized vertices.
4:     for all v ∈ V do
5:         if index of v is undefined then         ▷ We haven't analyzed the vertex v yet.
6:             Strong_Connect(G, v)
7:         end if
8:     end for
9: end procedure
```

This is the main routine, that goes through the vertices in a cycle and starts the procedure `Strong_Connect`$(\mathcal{G}, v)$ if it finds an unvisited vertex. We use the stack T to keep track of the way we traverse the graph. The recursive procedure `Strong_Connect` is called for each vertex exactly once and it visits each edge at most twice. Therefore

the algorithm is very fast, it runs in *linear time*, meaning that the number of operations required is of order $\mathcal{O}(|\mathcal{V}|+|\mathcal{E}|)$. This is very important in large calculations.

Algorithm 5 Tarjan's algorithm / II

```
 1: procedure STRONG_CONNECT(G, v)                          ▷ The recursive procedure.
 2:     v.index ← index                                     ▷ The index of v is the smallest unused index.
 3:     v.lowlink ← index        ▷ lowlink of v holds the lowest index in the SCC containing v.
 4:     index ← index +1
 5:     T.push(v)                                           ▷ We collect the visited vertices into T.
 6:     for all w such that (v,w) ∈ E do
 7:         if index of w is undefined then
 8:             Strong_Connect(G, w)      ▷ The recursive call with an unvisited neighbour.
 9:             v.lowlink ← min(v.lowlink, w.lowlink)       ▷ Updating the lowest index.
10:         else if w ∈ T then                              ▷ For a visited neighbour,
11:             v.lowlink ← min(v.lowlink, w.index)         ▷ we just update the lowest index.
12:         end if
13:     end for
14:     if v.lowlink = v.index then                         ▷ v is the root for the current SCC.
15:         Start registering a new SCC
16:         repeat          ▷ We record the vertices contained in T and belong to this SCC.
17:             w ← T.pop()
18:             add w to the current SCC
19:         until w = v
20:     end if
21: end procedure
```

Period of a graph

Having established that $\mathcal{G}$ is strongly connected, we may consider finding its period. The following function, `Find_Period` called with $p = 0$ and any vertex v with index 0, does exactly this; returns the period of the graph. The reader is referred to Luzzatto and Pilarczyk [25] and Jarvis [20].

Algorithm 6 Finding the period of $\mathcal{G}$

```
function FIND_PERIOD(G, v, p)
    for all w ∈ V : (v,w) ∈ E do
        if index of w is undefined then                     ▷ w is not visited.
            w.index ← v.index + 1
            Find_Period(G, w, p)          ▷ A recursive call with an unvisited neighbour.
        else                                                ▷ w is visited.
            p ← GCD(p, w.index − v.index − 1)  ▷ The greatest common divisor of the indices.
        end if
    end for
    return p
end function
```

Chapter 5

The method of Self-consistent Bounds for PDEs

In this Chapter we give an introduction to the method of *self-consistent bounds* for *dissipative PDEs* developed by Zgliczyński and Mischaikow [44] and Zgliczyński [40, 42, 43]. We apply these techniques to a certain destabilized Kuramoto-Sivashinsky equation (see Wittenberg [39]) in Paper D.

In Section 5.1 we introduce the concept of *self-consistent bounds* for a certain *class of dissipative PDEs*. First we build up the proper setting to handle the problem. We discuss the spaces, solutions, projections and coefficients involved. In the remaining part of the Chapter we include certain results from the aforementioned papers. The reader is referred to therein for the proofs. We obtain a solution of the PDE by integrating finite dimensional Galerkin-projections, we comment on the *properties of such solutions* in Section 5.2. For a fixed projection, we transform the equation to a *differential inclusion*, this is discussed in Section 5.3.

5.1 The method of Self-Consistent Bounds

The method is introduced in an abstract setting. Consider the following Hilbert space of square- integrable functions $\mathcal{H}_0 = L^2(\mathbb{R} \times \mathbb{R}^d)$. The elements of $\mathcal{H}_0$ are of the form $(t, \mathbf{x}) \mapsto u(t, \mathbf{x})$, where the variables represent time and space, respectively. Assume that $\mathcal{I} \subset \mathbb{Z}^d$ and $\mathcal{B}_{\mathcal{H}_0} = \{\xi_{\mathbf{k}}(\mathbf{x})\}_{\mathbf{k} \in \mathcal{I}}$ is an orthonormal basis of $\mathcal{H}_0$. Consider the evolution equation for $u \in \mathcal{H}_0$

$$\begin{cases} \frac{\mathrm{d}u}{\mathrm{d}t} = F(u), \\ u(t_0, \mathbf{x}) = u_0(\mathbf{x}), \quad u_0 \in \mathcal{H}_0, \end{cases} \tag{5.1}$$

where $F : \mathcal{H}_0 \to \mathcal{H}_0$ is a differential operator. We require that F is of the following form

$$u_t = Lu + N(u, \mathrm{D}u, \ldots, \mathrm{D}^r u),$$

where L is a *linear operator* that is diagonal in $\mathcal{B}_{\mathcal{H}_0}$, N is a *polynomial*, $\mathrm{D}^s u$ is the collection of all s-th order spatial partial derivatives of u. We require in addition that the eigenvalues of L, given by $L\xi_{\mathbf{k}}(\mathbf{x}) = \lambda_{\mathbf{k}} \xi_{\mathbf{k}}(\mathbf{x})$ for $\mathbf{k} \in \mathcal{I}$, are of the form

$$\lambda_{\mathbf{k}} = -\nu(\|\mathbf{k}\|)\|\mathbf{k}\|^p, \tag{5.2}$$

where $p > r$, $v\colon \mathbb{R}_0^+ \to \mathbb{R}$, and there exists $k_0 \in \mathbb{R}^+$ such that $v(z)$ is positive, uniformly bounded away from zero and from above for $z > k_0$. Under a *solution* of (5.1) we understand a differentiable function $u\colon [0, t_{max;u_0}) \times \mathbb{R}^d \to \mathcal{H}_0$ that satisfies (5.1) for all $t \in [0, t_{max;u_0})$. Note that expanding u in $\mathcal{B}_{\mathcal{H}_0}$ yields

$$u(t, \mathbf{x}) = \sum_{\mathbf{k} \in \mathcal{I}} u_{\mathbf{k}}(t) \xi_{\mathbf{k}}(\mathbf{x}),$$

thus we work with time-dependent coefficients, $t \mapsto u_{\mathbf{k}}(t) \in \mathcal{H}$, where $\mathcal{H} = l^2(\mathcal{I})$. However, when it does not affect the understanding, we omit the variable t and simply write $u_{\mathbf{k}}$. Note that $\mathcal{H}$ is a Hilbert space and $\mathcal{H}_0 \cong \mathcal{H}$. Accordingly, we consider the operators F, L and N to act on $\mathcal{H}$, identify u with its coefficient vector in the following. Thus, instead of (5.1), we consider the evolution equation for $u \in \mathcal{H}$

$$\begin{cases} \frac{\mathrm{d}u}{\mathrm{d}t} = F(u), \\ u(t_0) = u_0, \quad u_0 \in \mathcal{H}. \end{cases} \tag{5.3}$$

As one might expect, a solution of (5.3) is defined as a function $u\colon [0, t_{max;u_0}) \to \mathcal{H}$ that satisfies (5.3) for all $t \in [0, t_{max;u_0})$

Let $\mathcal{B} = \{e_{\mathbf{k}}\}_{\mathbf{k} \in \mathcal{I}}$ be the standard orthonormal basis of $\mathcal{H}$. Converting equation (5.3) onto $\mathcal{H}$ results in the *infinite ladder of ODEs*

$$\frac{\mathrm{d}u_{\mathbf{k}}}{\mathrm{d}t} = \lambda_{\mathbf{k}} u_{\mathbf{k}} + N_{\mathbf{k}}(u, \mathrm{D}u, \ldots, \mathrm{D}^r u), \quad \mathbf{k} \in \mathcal{I}, \tag{5.4}$$

where $N_{\mathbf{k}}$ is the $\mathbf{k}$-component of N.

Remark 5.1. Formula (5.2) implies that for $\|\mathbf{k}\|$ large enough, the term $\lambda_{\mathbf{k}} u_{\mathbf{k}}$ *dominates* (5.4) , moreover $\lambda_{\mathbf{k}}$ is negative, thus the components of a solution are expected to decay to zero at least at a *polynomial speed*.

Definition 5.2. Consider the decomposition of $\mathcal{H}$ into the direct sum of the mutually orthogonal subspaces $H_{\mathbf{k}} \subset \mathcal{H}$, again, indexed by $\mathcal{I}$

$$\mathcal{H} = \overline{\oplus_{\mathbf{k} \in \mathcal{I}} H_{\mathbf{k}}}. \tag{5.5}$$

Assume that for all $\mathbf{k} \in \mathcal{I}$ there exist an $l = l(\mathbf{k}) \in \mathbb{N}$ and basis vectors $e_{\mathbf{k}_1}, \ldots, e_{\mathbf{k}_l}$ such that $H_{\mathbf{k}} = \mathrm{span}\{e_{\mathbf{k}_1}, \ldots, e_{\mathbf{k}_l}\}$ and let $H_{\mathbf{k},s} = \mathrm{span}\{e_{\mathbf{k}_s}\}$. Assume in addition, that there is an $M > 0$ such that $\dim H_{\mathbf{k}} = 1$ for $\|\mathbf{k}\| > M$. Having these assumption fulfilled, we refer to (5.5) as a *block decomposition* of $\mathcal{H}$.

Remark 5.3. Note that these assumptions imply that $\dim H_{\mathbf{k}}$ is bounded uniformly.

In the following part of this Section, we assume that a block-decomposition $\mathcal{H} = \overline{\oplus_{\mathbf{k} \in \mathcal{I}} H_{\mathbf{k}}}$ is given. Let $m \in \mathbb{N}$ and $\mathcal{I}_m = \{\mathbf{k} \in \mathcal{I} : \|\mathbf{k}\| > m\}$ and define the spaces

$$X_m = \bigoplus_{\mathbf{k} \notin \mathcal{I}_m} H_{\mathbf{k}},$$
$$Y_m = X_m^{\perp}.$$

Note that $\mathcal{H} = X_m \oplus Y_m$. It is convenient to denote an element of X_m by $\mathbf{x}$. This is not the space variable in (5.1), due to moving to the coefficient space, that one is long gone, this should not cause confusion. Consider the orthogonal projections $A_{\mathbf{k}} \colon \mathcal{H} \to H_{\mathbf{k}}$, $A_{\mathbf{k},s} \colon \mathcal{H} \to H_{\mathbf{k},s}$, $P_m \colon \mathcal{H} \to X_m$ and $Q_m \colon \mathcal{H} \to Y_m$.

Definition 5.4. For an $m \in \mathbb{N}$, the *m-Galerkin projection* of (5.4) is the finite dimensional ODE

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = P_m F(\mathbf{x} \oplus \mathbf{0}), \tag{5.6}$$

where $\mathbf{x} \in X_m$ and $\mathbf{0}$ is the corresponding zero vector in Q_m. We denote by $\varphi^m(t, \mathbf{x})$, the flow on X_m induced by (5.6).

The natural question is if we are able to analyze (5.3) through the Galerkin projections. Obviously, some consistency conditions needs to be fulfilled in order to do this. This leads to the definition of self-consistent bounds.

Definition 5.5. Assume that $P_n F \colon X_n \to X_n$ is a C^1 function for all $n \in \mathbb{N}$ and let $0 < m \leq M < \infty$. Consider the structure

$$S = W \oplus \prod_{\mathbf{k} \in \mathcal{I}_m} B_{\mathbf{k}},$$

where $W \subset X_m$ and $B_{\mathbf{k}} \subset H_{\mathbf{k}}$ for $\mathbf{k} \in \mathcal{I}_m$ are compact sets. Assume in addition, that M is large enough to have $\dim H_{\mathbf{k}} = 1$ for $\mathbf{k} \in \mathcal{I}_M$.

Let us define the conditions **C1**, **C2**, **C3** and **C4** as follows.

C1 $0 \in B_{\mathbf{k}}$ for $\mathbf{k} \in \mathcal{I}_M$.

C2 $\sum_{\mathbf{k} \in \mathcal{I}_M} a_{\mathbf{k}}^2 < \infty$, where $a_{\mathbf{k}} = \max_{a \in B_{\mathbf{k}}} \|a\|$ for $\mathbf{k} \in \mathcal{I}_M$. This implies that $S \subset H$.

C3 $u \mapsto F(u)$ is continuous on S and $\sum_{\mathbf{k} \in \mathcal{I}} f_{\mathbf{k}}^2 < \infty$, where $f_{\mathbf{k}} = \max_{u \in S} \|A_{\mathbf{k}} F(u)\|$.

C4 For each $\mathbf{k} \in \mathcal{I}_m$ the set $B_{\mathbf{k}}$ is given either as an interval box $\prod_{s=1}^{l(\mathbf{k})} [\underline{a_{\mathbf{k},s}}, \overline{a_{\mathbf{k},s}}]$ or as a closed $l(\mathbf{k})$-ball $\overline{\mathrm{B}(c_{\mathbf{k}}; r_{\mathbf{k}})}$ with $r_{\mathbf{k}} > 0$ and $c_{\mathbf{k}} \in H_{\mathbf{k}}$. In addition, for $u \in S$ and $\mathbf{k} \in \mathcal{I}_m$ it holds that

- if $B_{\mathbf{k}}$ is given as a box, then

$$A_{\mathbf{k},s} u = \underline{a_{\mathbf{k},s}} \Rightarrow A_{\mathbf{k},s} F(u) > 0,$$
$$A_{\mathbf{k},s} u = \overline{a_{\mathbf{k},s}} \Rightarrow A_{\mathbf{k},s} F(u) < 0.$$

- if $B_{\mathbf{k}}$ is given as a sphere, then

$$A_{\mathbf{k}} u \in \mathrm{bd}_{H_{\mathbf{k}}} B_{\mathbf{k}} \Rightarrow \langle A_{\mathbf{k}} u - c_{\mathbf{k}}, A_{\mathbf{k}} F(u) \rangle < 0,$$

where $\mathrm{bd}_{H_{\mathbf{k}}}$ gives the boundary relative to $H_{\mathbf{k}}$.

We say that the set S forms *self-consistent bounds* if the conditions **C1**, **C2** and **C3** are satisfied. If **C4** holds in addition, we speak about *topologically self-consistent bounds*. We call W the *main part* and

$$T = \prod_{\mathbf{k}\in\mathcal{I}_m} B_{\mathbf{k}} \subset Y_m \tag{5.7}$$

the *tail*. In addition we refer to $\prod_{\mathbf{k}\in\mathcal{I}_M} B_{\mathbf{k}}$ as the *far-tail* and to $\prod_{\mathbf{k}\in\mathcal{I}_m\setminus\mathcal{I}_M} B_{\mathbf{k}}$ as the *mid-tail*. Note that **C4** means that the vector field points inwards on the boundary of the tail.

In the remaining part of the Chapter, we include certain results from Zgliczyński [40, 42, 43]. The reader is referred to these papers for the proofs.

Lemma 5.6. *Let $W \oplus T$ form self-consistent bounds for (5.4). Then*

- *$W \oplus T$ is a compact subset of $\mathcal{H}$,*
- *$\lim_{n\to\infty} P_n F(u) = F(u)$ uniformly for $u \in W \oplus T$.*

The connection between solutions of the Galerkin projections and of the evolution equation (5.3) is described by Lemma 5.7.

Lemma 5.7. *Let $W \oplus T$ form self-consistent bounds for (5.4). Let $m_n \in \mathbb{N}$ for all $n \in \mathbb{N}$. Assume that $\lim_{n\to\infty} m_n = \infty$, and that $\upsilon_n : [t_1,t_2] \to W \oplus T$ is a solution of the m_n-Galerkin projection (5.6) for all $n \in \mathbb{N}$.*

There exists a subsequence $(m_{n_i})_{i=0}^{\infty}$ such that $\lim_{i\to\infty} \upsilon_{n_i} = \upsilon^* \colon [t_1,t_2] \to W \oplus T$ uniformly on $[t_1,t_2]$ and υ^* satisfies (5.3).

Let us recall that Remark 5.1 implies polynomial decay rate for the coefficients. Let $s_0 = p + d + 1$ and consider tails that satisfy

$$B_{\mathbf{k}} \subseteq \frac{C}{\|\mathbf{k}\|^s}[-1,1], \tag{5.8}$$

for all $\mathbf{k} \in \mathcal{I}_M$, with some $C > 0$ and $s > s_0$. We say that such tail is a *polynomial tail* and when we speak about its *decay rate*, we mean s. Note that if $W \subset X_m$ is compact, then $W \oplus \prod_{\mathbf{k}\in\mathcal{I}_m} B_{\mathbf{k}}$ automatically satisfies conditions **C1** and **C2**. It may be shown that **C3** is satisfied as well; therefore, we obtain self-consistent bounds.

5.2 Existence, classical and analytic solutions

The following theorem states that starting from self-consistent bounds with polynomial tail and decay rate s, solutions exists for a certain time and they may be enclosed in self-consistent bounds with polynomial tail. Moreover, the enclosing tail has the same decay rate as the initial bounds. This is crucial for the method to work in practice. Furthermore, the block-decomposition will allow us to establish *qualitative* properties of the solutions.

Theorem 5.8. *Let $Z_0 \oplus T_0$ form self-consistent bounds with polynomial tail and decay rate s for (5.4). Then there exist $h > 0$, $n_0 \in \mathbb{N}$, and $W \oplus T$ self-consistent bounds with polynomial tail for (5.4) such that for all $n > n_0$ and initial condition $\mathbf{x} \in P_n(Z_0 \oplus T_0)$ it holds that*

$$\varphi^n([0,h],\mathbf{x}) \subset W \oplus T.$$

Moreover, T has the same decay rate as T_0.

Heuristically, a block-decomposition is σ-*smooth* if it may be lifted back to $\mathcal{H}_0$ and the norms of the partial derivatives of the lift at $u \in H_{\mathbf{k}}$ are bounded by a function of the form $R\|\mathbf{k}\|^{\sigma}\|u\|$ for all $\mathbf{k} \in \mathcal{I}$. We will not include the rather technical definition as a whole but we remark that the Fourier decomposition of $L_2([0,2\pi],\mathbb{R}^d)$ is a σ-smooth decomposition for all σ.

According to the following theorem, having a sufficiently smooth decomposition will imply that the solutions we obtain through the method are classical and analytic solutions.

Theorem 5.9. *Let $\upsilon\colon [t_1,t_2] \to W \oplus T$, where $W \oplus T$ are self-consistent bounds with polynomial tail and decay rate s for (5.4). Assume that υ is a solution of (5.4) and that the decomposition $H = \overline{\oplus_{\mathbf{k}\in\mathcal{I}} H_{\mathbf{k}}}$ is s-*smooth*. Then υ is a classical solution of (5.1) and it is analytic for all $t \in (t_1,t_2]$.*

5.3 Time integration

Assume that we have self-consistent bounds with polynomial tail $Z_0 \oplus T_0$ for (5.4), enclosing the solution $\mathbf{x}(t) \oplus T(t)$ at time t_0. We may obtain an enclosure after time h that is of the same structure. By considering the tail to be constant T_c, we obtain the m-dimensional ODE

$$\begin{aligned} \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} &= P_m F(\mathbf{x}(t) \oplus T_c), \\ \mathbf{x}(t_0) &= Z_0, \end{aligned}$$

where $\mathbf{x} \in X_m$. By including the perturbation caused by the tail, this becomes the differential inclusion

$$\begin{aligned} \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} &\in P_m F(\mathbf{x}(t) \oplus T_c) + \left(P_m F(\mathbf{x}(t) \oplus T(t)) - P_m F(\mathbf{x}(t) \oplus T_c)\right), \\ \mathbf{x}(t_0) &= Z_0. \end{aligned} \tag{5.9}$$

In order to use the theory described in Section 2.5, we need to obtain a rough enclosure for the whole trajectory in advance. This is referred to as the generation of *a-priori* bounds. Once more, the reader is referred to Zgliczyński [43] for a detailed analysis.

Chapter 6

Overview of the papers

Paper A:
Computing of B-series by Automatic Differentiation
Ferenc A. Bartha and Hans Z. Munthe-Kaas

We consider a B-series, named after John C. Butcher, that is a fundamental tool for the study of certain numerical integration methods [8, 18] for the ordinary differential equation $y' = f(y)$. A B-series is traditionally given by the infinite sum

$$B_f(\beta) = \sum_{t \in T} \frac{h^{|t|}}{\sigma(t)} \beta(t) \mathcal{F}_f(t),$$

where T denotes the set of rooted trees, $\sigma(t) \in \mathbb{Z}$ is the tree symmetry function, $|t|$ denotes the number of nodes in t, $h > 0$ is the timestep of the numerical integrator and $\beta : T \to \mathbb{R}$ is a given function. The terms $\mathcal{F}_f(t)$ are called elementary differentials and are defined recursively as certain higher order partial derivatives of f. We study this recursion and the isomorphisms of rooted trees. We provide an algorithm for computing the elementary differentials for all rooted trees in T_d, that is for all rooted trees with at most d nodes. We use Automatic Differentiation to evaluate the higher order derivatives defining the elementary differentials. In particular, we follow the method of Griewank *et al.* [17] and propagate univariate Taylor series in certain directions, as we have seen in Chapter 3.

Paper B:
Local stability implies global stability for the 2-dimensional Ricker map
Ferenc A. Bartha, Ábel Garab and Tibor Krisztin

In this paper we consider the delay difference equation $x_{k+1} = x_k e^{\alpha - x_{k-d}}$, where α is a positive parameter and d is a nonnegative integer. The case $d = 0$ was introduced by W.E. Ricker in 1954 as a population model [31]. For the delayed version $d \geq 1$ of the equation S. Levin and R. May conjectured in 1976 [23] that local stability of the nontrivial equilibrium implies its global stability. We consider $d = 1$ and introduce the

equivalent two-dimensional map

$$F\colon \mathbb{R}^2 \to \mathbb{R}^2,\ F(x,y) = F_{\alpha,m}(x,y) = (y, my - \alpha\varphi(x)).$$

In this case (α,α) is locally stable given $\alpha \in (0,1)$ and at $\alpha = 1$ a Neimark–Sacker bifurcation occurs. We prove the conjecture for $d = 1$ by showing that the fixed point (α,α) is globally asymptotically stable for $\alpha \in (0,1]$, that is even for the bifurcation parameter $\alpha = 1$.

The proof consists of the following three major steps:

1. The construction of $S^{(\alpha)}$, a compact, attracting, invariant, trapping region around the fixed point.
2. The construction of $N^{(\alpha)}$, a neighbourhood of the fixed point that is contained in the basin of attraction of (α,α).
3. Showing that any orbit starting from $S^{(\alpha)}$ eventually enters the neighbourhood $N^{(\alpha)}$, thus it is attracted by the fixed point.

The set $S^{(\alpha)}$ is obtained using elementary calculations. We derive formulae for $N^{(\alpha)}$ both from the linearized equation and from the bifurcation normal form. Here we use the help of Wolfram Mathematica to do certain symbolic calculations. Note that using the normal form is crucial, as we get closer to $\alpha = 1$, the size of the neighbourhood obtained from the linearization goes to zero. After we have derived uniform expressions for these sets, we use graph representations, as described in Chapter 4, based on rigorous computations to establish our claim. We also give the proof of correctness of the algorithm for enclosing non-wandering points, as it is relevant to our case.

Paper C:
Necessary and sufficient condition for the global stability of a delayed discrete-time single neuron model
Ferenc A. Bartha and Ábel Garab

We study the global asymptotic stability of the trivial fixed point of the delay difference equation $x_{n+1} = mx_n - \alpha\varphi(x_{n-1})$, where $(\alpha,m) \in \mathbb{R}^2$ and φ is a real function that satisfies $0 \le x\varphi(x) \le x^2$ for all $x \in \mathbb{R}$. Using elementary calculations, we show that $(\alpha,m) \in (|m|-1, 1/(1+|m|)) \times (-1,1)$ is a sufficient condition for the global asymptotic stability of 0.

We consider the special sigmoid type feedback $\varphi(x) = \tanh(x)$, commonly used in neural networks. We introduce the two-dimensional map

$$F\colon \mathbb{R}^2 \to \mathbb{R}^2,\ F(x,y) = F_{\alpha,m}(x,y) = (y, my - \alpha\varphi(x))$$

and use the same techniques as in Paper B to investigate the global stability of the fixed point. We prove that the condition $(\alpha,m) \in [|m|-1,1] \times [-1,1]$, $(\alpha,m) \neq (0,-1),(0,1)$ is necessary and sufficient for global asymptotic stability.

Paper D:
Fixed point of a destabilized Kuramoto-Sivashinsky equation
Ferenc A. Bartha and Warwick Tucker

Various forms of the Kuramoto-Sivashinsky equation have been considered in the literature. Instead of picking a special one, we work with $u_t + \nu u_{xxxx} + \beta u_{xx} + \gamma u u_x = \alpha u$, assuming one spatial dimension. Thus, by appropriate choices of parameters, one obtains the most common members of the family of the KS-equations. For $\alpha > 0$, the equation is destabilized by αu. We will study the L-periodic stationary solutions of our model, and as an example, we take the parameter values $\alpha = 0.5$, $\beta = 2$, $\gamma = -1$, $\nu = 1$ and $L = 30$. These choices are motivated by Zgliczyński [40, 44] and Wittenberg *et al.* [30, 39]. In the last two papers referred, shock-like, odd, stationary solutions have been observed numerically.

We will use the framework of self-consistent bounds by Zgliczyński [40, 42–44] to validate the existence of a stationary solution of this kind. We put special emphasis on the transformation of the equation in order to give a better understanding of the method. An introduction to self-consistent bounds is given in Chapter 5.

Bibliography

[1] ALEFELD, G. Introduction to interval analysis. *SIAM Rev. 53*, 2 (2011), 380–381. 1.1, 1.2

[2] ARNOLD, V. I. *Ordinary differential equations*. Universitext. Springer-Verlag, Berlin, 2006. Translated from the Russian by Roger Cooke, Second printing of the 1992 edition. 2.4.1

[3] BENDTSEN, C., AND STAUNING, O. FADBAD, a flexible C++ package for automatic differentiation –- using the forward and backward methods. 1.4, 2.4.2

[4] BERZ, M. Algorithms for higher derivatives in many variables with applications to beam physics. In *Automatic differentiation of algorithms (Breckenridge, CO, 1991)*. SIAM, Philadelphia, PA, 1991, pp. 147–156. 3

[5] BIRKELAND, T., AND NEPSTAD, R. Pyprop. `http://pyprop.googlecode.com`.

[6] BOGÁR, F., BARTHA, F., BARTHA, F. A., AND MARCH, N. H. Pauli potential from Heilmann-Lieb electron density obtained by summing hydrogenic closed-shell densities over the entire bound-state spectrum. *Phys. Rev. A 83* (Jan 2011), 014502, doi: 10.1103/PhysRevA.83.014502.

[7] BOYCE, W. E., AND DIPRIMA, R. C. *Elementary differential equations and boundary value problems*. John Wiley & Sons Inc., New York, 1965. 2.4.1

[8] BUTCHER, J. An algebraic theory of integration methods. *Math. Comp 26*, 117 (1972), 79–106. 3, 6

[9] CAPA: COMPUTER-AIDED PROOFS IN ANALYSIS GROUP. `http://www2.math.uu.se/~warwick/CAPA/`. University of Uppsala, University of Bergen.

[10] COMPUTER ASSISTED PROOFS IN DYNAMICS GROUP. CAPD Library. `http://capd.ii.uj.edu.pl`. a C++ package for rigorous numerics. 1.2, 1.4

[11] DANIS, A. PhD Thesis. Parameter estimation, set valued numerics – in preparation. *Uppsala University* (2012). 3

[12] DELLNITZ, M., AND HOHMANN, A. A subdivision algorithm for the computation of unstable manifolds and global attractors. *Numer. Math. 75*, 3 (1997), 293–317, doi: 10.1007/s002110050240. (document), 2.3.1, 4, 4.3

[13] DELLNITZ, M., HOHMANN, A., JUNGE, O., AND RUMPF, M. Exploring invariant sets and invariant measures. *Chaos 7*, 2 (1997), 221–228, doi: 10.1063/1.166223. 4

[14] DEVANEY, R. L. *An introduction to chaotic dynamical systems*. Studies in Nonlinearity. Westview Press, Boulder, CO, 2003. Reprint of the second (1989) edition. 2.5

[15] GALIAS, Z. Rigorous investigation of the Ikeda map by means of interval arithmetic. *Nonlinearity 15*, 6 (2002), 1759–1779, doi: 10.1088/0951-7715/15/6/304. (document), 2.3.1, 4, 4.4, 4.4, 4.5

[16] GRIEWANK, A. *Evaluating derivatives*, vol. 19 of *Frontiers in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000. Principles and techniques of algorithmic differentiation. 1.4, 2.4.2, 3

[17] GRIEWANK, A., UTKE, J., AND WALTHER, A. Evaluating higher derivative tensors by forward propagation of univariate Taylor series. *Math. Comp. 69*, 231 (2000), 1117–1130, doi: 10.1090/S0025-5718-00-01120-0. (document), 3, 3.5, 6

[18] HAIRER, E., LUBICH, C., AND WANNER, G. *Geometric numerical integration: Structure-preserving algorithms for ordinary differential equations*, vol. 31. Springer, 2006. 3, 6

[19] HIRSCH, M. W., SMALE, S., AND DEVANEY, R. L. *Differential equations, dynamical systems, and an introduction to chaos*, second ed., vol. 60 of *Pure and Applied Mathematics (Amsterdam)*. Elsevier/Academic Press, Amsterdam, 2004. 2.4.1

[20] JARVIS, J. P., AND SHIER, D. R. Graph-theoretic analysis of finite markov chains. 4.6

[21] KAPELA, T., AND ZGLICZYŃSKI, P. A Lohner-type algorithm for control systems and ordinary differential inclusions. *Discrete Contin. Dyn. Syst. Ser. B 11*, 2 (2009), 365–385, doi: 10.3934/dcdsb.2009.11.365. 2, 2.5

[22] LERCH, M., TISCHLER, G., GUDENBERG, J. W. V., HOFSCHUSTER, W., AND KRÄMER, W. Filib++, a fast interval library supporting containment computations. *ACM Trans. Math. Softw. 32*, 2 (June 2006), 299–324, doi: 10.1145/1141885.1141893. 1.2

[23] LEVIN, S. A., AND MAY, R. M. A note on difference-delay equations. *Theoret. Population Biology 9*, 2 (1976), 178–187. 4, 6

[24] LOHNER, R. J. Enclosing the solutions of ordinary initial and boundary value problems. In *Computerarithmetic*. Teubner, Stuttgart, 1987, pp. 255–286. 2, 2.1.2, 2.4.3

[25] LUZZATTO, S., AND PILARCZYK, P. Finite resolution dynamics. *Found. Comput. Math. 11*, 2 (2011), 211–239, doi: 10.1007/s10208-010-9083-z. 4, 4.6, 4.6

[26] MAKINO, K., AND BERZ, M. Taylor models and other validated functional inclusion methods. *Int. J. Pure Appl. Math. 6*, 3 (2003), 239–316. 2

[27] MOORE, R. E. *Methods and applications of interval analysis*, vol. 2 of *SIAM Studies in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, Pa., 1979. 1.1, 1.2, 1.4, 2.4.3, 4.4

[28] MROZEK, M., AND ZGLICZYŃSKI, P. Set arithmetic and the enclosing problem in dynamics. *Ann. Polon. Math. 74* (2000), 237–259. Dedicated to the memory of Bogdan Ziemian. 2.1.2, 2.8

[29] NEDIALKOV, N. S., JACKSON, K. R., AND CORLISS, G. F. Validated solutions of initial value problems for ordinary differential equations. *Appl. Math. Comput. 105*, 1 (1999), 21–68, doi: 10.1016/S0096-3003(98)10083-8. 2.1.2

[30] RADEMACHER, J. D. M., AND WITTENBERG, R. W. Viscous shocks in the destabilized Kuramoto-Sivashinsky equation. *Journal of Computational and Nonlinear Dynamics 1*, 4 (2006), 336–347, doi: 10.1115/1.2338656. 6

[31] RICKER, W. E. Stock and recruitment. *Journal of the Fisheries Research Board of Canada 11*, 5 (1954), 559–623, doi: 10.1139/f54-039. 4, 6

[32] RUMP, S. INTLAB - INTerval LABoratory. In *Developments in Reliable Computing*, T. Csendes, Ed. Kluwer Academic Publishers, Dordrecht, 1999, pp. 77–104. `http://www.ti3.tu-harburg.de/rump/`. 1.2

[33] SIEK, J. G., LEE, L.-Q., AND LUMSDAINE, A. *The Boost Graph Library User Guide and Reference Manual (With CD-ROM)*. 2002.

[34] TARJAN, R. Depth-first search and linear graph algorithms. *SIAM Journal on Computing 1*, 2 (1972), 146–160, doi: 10.1137/0201010. 4.6

[35] TUCKER, W. A rigorous ODE solver and Smale's 14th problem. *Found. Comput. Math. 2*, 1 (2002), 53–117. 1.1, 1.2

[36] TUCKER, W. *Validated numerics*. Princeton University Press, Princeton, NJ, 2011. A short introduction to rigorous computations. 1.1, 1.2, 1.3, 1.4, 2.4.3, 4.4

[37] WILCZAK, D. Uniformly hyperbolic attractor of the Smale-Williams type for a Poincaré map in the Kuznetsov system. *SIAM J. Appl. Dyn. Syst. 9*, 4 (2010), 1263–1283, doi: 10.1137/100795176. With online multimedia enhancements. 4

[38] WILCZAK, D., AND ZGLICZYŃSKI, P. Cr-Lohner algorithm. *Schedae Informaticae 20* (2011), 9–46. 2.1.2, 2.1.2

[39] WITTENBERG, R. W. Dissipativity, analyticity and viscous shocks in the (de)stabilized Kuramoto–Sivashinsky equation. *Physics Letters A 300*, 4–5 (2002), 407 – 416, doi: 10.1016/S0375-9601(02)00861-7. 5, 6

[40] ZGLICZYŃSKI, P. Attracting fixed points for the Kuramoto-Sivashinsky equation: a computer assisted proof. *SIAM J. Appl. Dyn. Syst. 1*, 2 (2002), 215–235 (electronic), doi: 10.1137/S111111110240176X. (document), 5, 5.1, 6

[41] ZGLICZYŃSKI, P. C^1 Lohner algorithm. *Found. Comput. Math. 2*, 4 (2002), 429–465, doi: 10.1007/s102080010025. 2.1.2, 2.1.2

[42] ZGLICZYŃSKI, P. Rigorous numerics for dissipative partial differential equations. II. Periodic orbit for the Kuramoto-Sivashinsky PDE—a computer-assisted proof. *Found. Comput. Math. 4*, 2 (2004), 157–185, doi: 10.1007/s10208-002-0080-8. (document), 5, 5.1, 6

[43] ZGLICZYŃSKI, P. Rigorous numerics for dissipative PDEs III. An effective algorithm for rigorous integration of dissipative PDEs. *Topol. Methods Nonlinear Anal. 36*, 2 (2010), 197–262. (document), 5, 5.1, 5.3

[44] ZGLICZYŃSKI, P., AND MISCHAIKOW, K. Rigorous numerics for partial differential equations: the Kuramoto-Sivashinsky equation. *Found. Comput. Math. 1*, 3 (2001), 255–288, doi: 10.1007/s10208-002-0080-8. (document), 5, 6