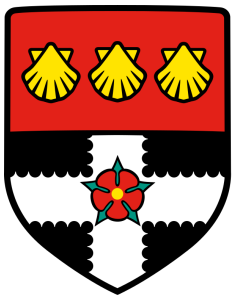
University of Reading

Department of Mathematics and Statistics



ITERATION OF INVERSE PROBLEMS AND DATA ASSIMILATION TECHNIQUES FOR NEURAL FIELD EQUATIONS

Jehan Alswaihli

A thesis submitted for the degree of Doctor of Philosophy

June 2020

Abstract

The need to understand the neural field activity for realistic living systems is a current challenging task in neuroscience. For several decades, neural fields have been studied and developed theoretically and numerically. However, to make practical use of the equations, we need to *determine their constituents* in practical systems. This includes the determination of parameters or the reconstruction of the underlying connectivity in biological tissue.

The thesis is part of the fields of *inverse problems* and *data assimilation* applied to neural field theory. Inverse problems deals with the *reconstruction of structural information or characteristics* of some natural system and data assimilation deals with the repeated *estimation of the dynamical state* of a system. Dealing with medical systems, both tasks are strongly related due to the fact that often structural information is missing or strongly incomplete, such that the *state estimation* needs to reconstruct the structural information and the *reconstruction* needs estimation of the states.

Our main achievement is building an iterative approach to determine both the neural states for simulations and the underlying parameter functions. This procedure takes in turn the state estimation or data assimilation problem and the inverse neural field problem. The thesis is concerned with these two basic tasks in the framework of a *neural field model*. In the three main chapters, we study the reconstruction of the kernel for the delay neural field equation, first providing an easy to follow proof for the existence and uniqueness of the solution of the direct problem. Then, we work on reconstruction of the state for the non-delay case using 3D-Var.

Subsequently, the iterative approach is introduced as a combination of three operators depending on the model itself (*Transport Operator T*), data assimilation (*Estimation Operator E*) and the inversion (*Kernel Reconstruction Operator K.*) As an analytic contribution, a proof of convergence for the iterative approach of data assimilation and inversion techniques is developed. Finally, we implement our algorithm to show the feasibility of the steps individually and as an iterative approach. The examples are built on standard types of medical *measurement data* which are usually functions of the neural excitation fields.

Dedication

To my mum who is the inspiration of each success I have ever had.
To my family for their love, patience and support.

Declaration

I declare that this thesis is my original authorial work. All sources, references and literature used are properly cited and listed in complete reference.

Acknowledgments

I would like to thank everyone who has supported me even with nice words and wishes.

First and foremost I'd like to thank my supervisor, Professor Roland Potthast. Thank you for accepting me as one of your students and for your guidance, patience and all support and understanding of my circumstances even in my struggles time. It has been an honor working with you and learning from you. Your enthusiasm and love of science gave me the inspiration and the motivation to think differently. Without you and your help, this work would not have been possible.

Secondly, many thanks to Professor Douglas Saddy for all of your encouraging feedback and patience. I have learned so much from you. I am also grateful to my monitoring committee Dr. Amos Lawless and Prof. Sarah Dance. You have been great monitors, providing honest feedback and suggestions over the years, and presenting so many opportunities to help me grow as a researcher.

Thanks to staff of the departments of mathematics and statistics, graduate school, DARC and CINN for their support and advice. I have greatly enjoyed and benefited from courses taught and seminars held at the university. Thanks to the wonderful friends for the amazing discussion in science, philosophy and politics.

Special thanks to Peta-Ann King, the former administrator in our department for her support, help and offering a friendly environment at our early start. Thanks also to Kristine Aldridge (Kris) for her continuous care and support.

Thank you all for the great help you gave me and for the nice memories we shared that enriched my life.

Contents

List of Figures

Chapter 1

Introduction

The importance of studying neural field equations is rooted in their role for understanding the dynamics of the neural system. Studying of its organization and how that related to its behaviour is one of the core challenges in neurobiology. Interest in developing both theory and applications of neural fields increased along with advances in recording and imaging of brain activity. Starting with the first recording of the human electroencephalogram (EEG) in 1924, increasing success in experimental neuroimaging dataobtained from different methods such as EEG, MEG, fMRI, has driven then for mathematical models to interpret this data avalanche.

The rich history of mathematical modeling in neuroscience was the first motivation of the present work [102, 103, 78, 53, 54, 70, 58, 4]. In particular, recent neural modeling research which address both the direct and inverse problems inspire us to contribute in this field of study, see for example [26, 17, 43, 77, 85, 28, 96].

The focus of this thesis is on four different aspects:

- Significantly extending recent results on neural fields to the case of delay.
- Studying the theory of both the direct and the inverse problem of the delayed neural field equations and its sensitive analysis.
- Introducing techniques of data assimilation to neural dynamics.
- Designing and investigating an iterative method based on inversion and data assimilation and proving its convergence reconstructing neural dynamics from given measurements.

In this introduction, we start with an overview section consisting of an introduction to both the biological and mathematical aspects of neural equations with

time delay. It is followed by two subsections describing previous related studies, the contributions of this thesis and a section with the outline of this thesis.

1.1 Overview

The development of applying mathematical theory and numerical techniques to understand neural processes has come under the spotlight since the second half of the last century, when important work in modeling neurodynamics started. That work aims to bridge the gap between neuroimaging data and neural activity. The different successful techniques of multi-sensor recordings and imaging methods measure signals of millions of neurons. Modeling aims to study and reveal the underlying features of the complex recorded activity itself and hopefully provide other researchers new tools for understanding brain based behaviours like working memory, epilepsy and hallucinations [33, 98, 90]. A core aim is explaining the relation between the firing dynamics in populations of neurons and the activity of an individual neuron.

Biologists, psychologists and clinicians have always been driven to understand the biology, organization and activity of the brain. Today, it is further believed that the improvements in neuroscience will take humanity a step further in the study and improvement of diverse other fields such as artificial intelligence, medical imaging and machine learning [100]. We now understand that neural networks are a key feature of brain activity. Developing simulation techniques that allow us to study and test hypotheses about the properties of individual networks and the interaction of multiple networks has become essential to our understanding of their in typical and atypical neurological behaviour [29].

Researchers from diverse fields within different disciplines, such as engineering, physiology and psychology were motivated by the potential and applications of this technology. Early interest started in 1940s. Then, during the following two decades, it lost its sheen. Later, since 1980s, it grew rapidly over time because of its successes in both theory and application. That scientific success has crucial impact in computational neuroscience and mathematical biology. In fact, that increased the possibility of applying computation to assimilate the human neural processes [100].

Over decades, studying the activity of neural tissue and development of mathematical and numerical techniques to understand neural processes has led to improved neural field models. Neural field models have not only become an effective tool in neuroscience, but also helped the community of modeling the activity of neural tissue to grow strongly in different aspects [17, 101, 36, 98, 15, 26, 95, 43, 81].

The most famous formulations of mathematical models of neural fields have been introduced in the 1970s by Wilson and Cowan, Amari and Nunez [102, 103, 78, 53, 54]. Neural fields models are considered as neural mass models that describe the activity of neural population at spatio-temporally coarse-grained scales [78, 103], in activity based [103] and voltage based [78, 54] models. A variety of solutions is studied such as periodic patterns, bumps and waves as well as their existence and uniqueness with smooth sigmoidal firing rates or smoothed Heaviside firing rate functions [38, 23, 85, 24, 28, 81]. Other studies have considered different perspectives such as stability [6], inverse problems [12, 82], data assimilation [77], dendritic processing [17], synaptic depression [18] and the presence of delay [78, 99, 40, 96, 47, 97].

Neural problems belong to the class of complex systems due to the sophisticated interconnections between the large number of neurons where each of these neurons is a complex biological system itself [17, 26, 14]. For that, simplifying the analysis of neural activity is a challenging task of various research. We note that neural tissue has a large variety of properties in space and time, and whatever level of approximation and complexity is employed, the task to determine the structural information and the initial states are indispensable.

Neural networks occurring in nature are typically complex systems sporting a large variety of properties in space and time as we mentioned earlier. Simplifying their analysis is generally difficult – in particular when one considers the many billions of neurons of the entire human nervous system, where each of these neurons can be considered as a complex biological system by and in itself, cf.[17, 14]. However, neural field models describe these complicated system mathematically in a few equations, essentially by using the large number of neurons to achieve simplification in terms of mass action. Thus, these models consider averages of neural activity as a dynamical variable, and averages of neural properties as parameters. The derivation of neural models from properties of single neurons and their networks, and the analysis of the resulting activity, remains a major focus of current research [17, 77, 47, 26, 98, 40, 6].

The essential motivation of the neural modeling task is a results of the progress of different imagining techniques such as EEG and MEG. Further new and important imaging technologies have been MRI and DTI. Here, the response of the brain with respect to particular stimulation or tasks of human probands is studied. However, in the absence of any stimulation or tasks, the experimental results often show some kind of fluctuations. The question of the cause of these is not fully settled. Scientists argue that this could be an evidence of the role of time delays and study these data with respect to different aspects, e.g. the role of neural delay for the resting state of the brain, see for example [32, 46, 45, 75].

This section is split into five subsections. We provide a brief introduction to the biological motivation in the first Section 1.1.1, followed by a discussion of the development of the mathematical modeling in Section 1.1.2. In the third part of this section, we introduce the delayed neural field equation which is the main interest in this work. Then, we provide a review of some more related studies that elucidate my work further in Section 1.1.4. The main ideas of the contributions of this thesis are introduced in the Section 1.1.5.

1.1.1 The Biological Aspect

The challenge of modeling and simulating neural system activity is described in recent studies of mathematical biology and computational neuroscience. There is a huge body of literature studying both the simple neuron modeling and neural networks theory [70, 4, 58, 60, 36]. The neural system consists of the brain as the main component, the spinal cord and branching neural tissues to connect this organ with the whole body.

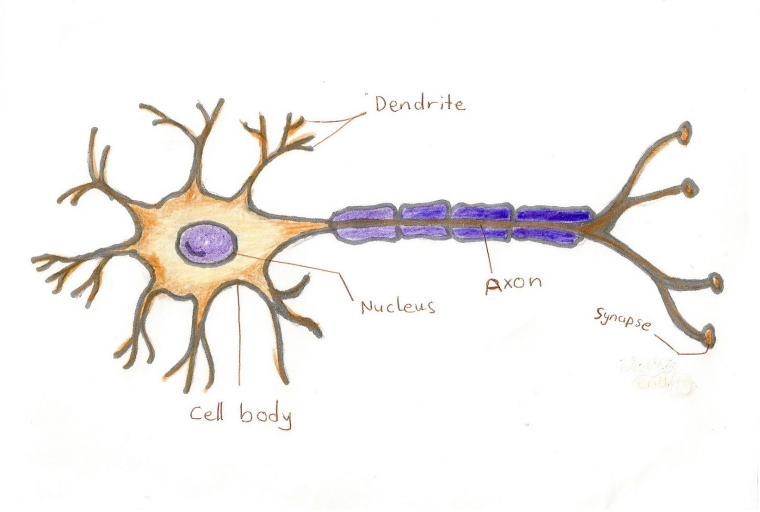


Figure 1.1: The biological structure of a neural cell indicating each of the structural components considering that they vary in shape and size depending on their function. They demonstrate two kind of processes, receiving chemical messages through dendrites and transmitting electrochemical signals through axons

Neural tissue presents various properties in space and time. Although we are interested in the average of the activity of a neural population, we need to consider their complicated interconnection and we know that each neuron has its own

properties and nearby units will respond differently. That makes the simplification of the analysis of neural activity very challenging considering the large number of neurons, as many as 100 billion neurons, their variety of shape and size, and their complicated interconnections in human nerve system. The common neuron structure has some characteristics of the other body cells, consisting of soma (a cell body), dendrites (branched process) that connect the neurons to each other, synapses (connection points) which conduct the input to the cell body, and an axon, with a special ability of interacting with other neurons by electrochemical signals, see Figure 1.1 for a general structure of a neural unit. The neurons can receive, process, and transmit signals through the transmission paths.

In neural dynamics, neurons have connections that allow them to exchange information with each other at synapses, compare Figure 1.1. There are two core processes. Receiving messages from other neurons through the synapses, or from sensing cells by dendrites. The second is sending electrical spikes to other neurons, muscles or gland cells through axons. The inputs either excite or inhibit the targeted neuron. In the case of excitation, the neuron fires when the excitation level reaches a threshold. Although neural networks present a diversity in structures, modeling their activity mostly follows the following similar approaches.

A set of input values is established to represent the initial state of neurons about to send a signal to a set of sets of downstream neurons to which they are connected. the input values are multiplied by different weights representing the differing excitatory and inhibitory synaptic strengths in the network. The summation of these weighted inputs models the neural activity. In reality, the fundamental neural reaction is quite sophisticated, for more details see for example [14, 4, 58, 70]. It is currently not possible to stimulate the neural behaviour in its full complexity. Simplifying assumptions and structures to reconstruct and investigate neural dynamics are imployed to keep the model as simple as possible while capturing the essential features [17, 100].

1.1.2 The Mathematical Modeling

The development of multi-sensor recording and imaging techniques of neural activity provides an opportunity to further develop neural modeling. Researchers, from different disciplines such as neuroscientists, biologists, physicists and mathematicians, show an interest in modeling the brain behaviour each from their own prospective. Although the process is described biologically to start as information reaches the soma through the dendrites, where it is integrated and sent through axon to other neuron, models consider the process to start from an axon of the

post-synaptic neuron to the the dendrite of the presynaptic one. In general, most of the models are designed to be simple while they retain some important properties such as firing rates, membrane potential and synaptic connection. However they mostly ignore other features such as dendritic processing and delay. Usually, two different models are distinguished, following basic assumptions.

First, there is the voltage-dependent (current-based) model. In this case the membrane properties governs the decay and the membrane time constant is dominant in comparison to the decay time of the synapses. To build the model, the post-synaptic potential is assumed to be a linear convolution of pre-synaptic spiking input. In this case it can be formulated as a linear differential operator with constant coefficient.

Second, there is the activity-based (conductance-based) model. The time constant then is dependent on the synaptic decay due to the longer synaptic time than the small membrane time. Although the latter is widely used, the former is more popular because of their simplicity, analytical and numerical advantages, For more details see [38, 21, 25].

There have been significant developments in theory, application and time estimation of neural fields since the main work of Wilson, Cowan, Nunez and Amari in the 1970s. Amari and Nunez analysed neural fields, and suggested that cortical neural tissues can be mathematically modelled as neural fields. Then, they formulated a continuous neural field equation propagating both excitation and inhibition. Their mathematical concept leads to a nonlinear integro-differential equations (1.4), that describes the large- scale dynamics of spatially structured networks of neurons. It is an important milestone for theoretical and practical reasons, e.g. because it is consistent with the measurements EEG. For more details we refer to [54, 78, 76, 68].

Stating that neurons send their electrical spikes to each other through axons terminating in synapses. Let $u(r_j, t)$ denote the average membrane potential of the j-th neuron or population of neurons located at position r_j at time t in a network of N units. Let $W(r_j, r_i)$ be the average connectivity strength between the neurons at position r_i and those neurons at position r_j. The function f is the activation rate or firing rate function, which describes the conversion of the membrane potential $u(r_j, t)$ into a spike train $S(r_i, t) = f(u(r_i, t))$. It is then leading to an excitation of neurons at location r_j with strength $W(r_j, r_i)S(r_i, t)$. The dynamics of the excitation is now described by the ODEs

$$\tau \frac{du}{dt}(r_j, t) = -u(r_j, t) + \sum_{i=1}^{N} W(r_j, r_i) f(u(r_i, t)), \tag{1.1}$$

which, with their exponential decay term and sum of excitation terms, is also called a *leaky integrator model*. Here, τ is a time constant. The sum represents the net-input to unit j, i.e. the weighted sum of activity delivered by all units i that are connected to unit j with a connection strength

$$w_{ij} := W(r_j, r_i),$$

which is sometimes referred to as the *synaptic footprint* or the *connectivity function*, see [64, 15, 28, 96, 88]. Here, W gives three different pieces of information: existence of connectivity between two neurons i and j if $w_{ij} \neq 0$, the effect if the neuron is either excitatory ($w_{ij} > 0$) or inhibitory ($w_{ij} < 0$) and the strength of the synapse $|w_{ij}|$, compare [85, 25, 76].

The activation function f is considered frequently in different choices of functions. One of them is a Heaviside step function

$$f(s) := \begin{cases} 0, & s < \eta, \\ 1, & s \geq \eta, \end{cases} \tag{1.2}$$

with $s \in \mathbb{R}$ and an activation threshold η. The second choice is to use a continuous activation functions $f(s) = Prob(s \geq \eta)$. This function describes the probability of the neuron to fire when its membrane potential reaches the threshold η. This probability is approximated by the sigmoid function:

$$f(s) = \frac{1}{1 + e^{-(s-\eta)}}, \quad s \in \mathbb{R}. \tag{1.3}$$

Written as a field equation, the Amari and Nunes ansatz receives the form

$$\tau \frac{\partial u}{\partial t}(r,t) = -u(r,t) + \int_\Omega w(r,r') f(u(r',t))\, dr' \tag{1.4}$$

with *initial condition*

$$u(r,0) = u_0(r), \quad r \in \Omega. \tag{1.5}$$

Here $u(r,t)$ is the membrane potential of the population of neurons at position $r \in \Omega$ on the cortex and at time t. The nonlinear function f interprets the neural firing rate, while the kernel or the connectivity function $w(r,r')$ represents how the neurons separated by a distance $|r - r'|$ interact with each other. The kernel is often assumed to be homogeneous for simplicity, i.e. only depending on $|r - r'|$. compare [54, 78, 12, 88, 28].

1.1.3 Delayed Neural Field Equations

Although, the neural field equation (1.4) represents several biological mechanism, this form still neglects any delay between spatial locations. It was Nunez [78, 79] who adds transmission delay to Amari equation to be more realistic. This delay is due to the finite propagation speed of action potential and dendritic integration. Delays in neural models have been studied widely. For activity-based models Coombes [27] show how the delays contribute to the generation of network rhythms. More studies will be mentioned later in section 1.1.4. In reality, there are different sources of delays could occur during the neural activity. Some delays could happen inside the neural cell itself and some could occur due to internal features [19]. We can summarize the biological mechanisms which cause delays briefly in the following points:

- The propagation of action potentials along the axon with finite transmission speed.
- The variety of conduction velocities.
- The action potential at the end of the axon.
- The distance between the cell body and the synapses.
- The transmission of the electrical signal across the synapses.

The finite transmission speed in axons, synapses and dendrites cause a functionally significant delay in neural system. Introducing a delay term to the equation (1.4), the neural field equation involving delayed interactions becomes

$$\tau \frac{\partial u}{\partial t}(r,t) = -u(r,t) + \int_{\Omega} w(r,r') f(u(r', t - D(r,r'))) \, dr', \tag{1.6}$$

with initial condition

$$u(r,t) = u_0(r,t), \quad (r,t) \in \Omega \times [-|\Omega|, 0], \tag{1.7}$$

as the delay is bounded by $|\Omega|$. The delay is typically assumed to be $D(r,r') \simeq \tilde{D}(r,r')/v$, i.e., the total length of the neural fibers $\tilde{D}$ connecting locations r and r', divided by v, the finite transmission speed of neural signals (action / post-synaptic potentials) along those fibers. In Figure 1.2 we show the connection between neurons in a simple way.

If we look at the Figure 1.2, we can understand that the time delay will have occurred when the action potential from neuron at the position r reaches the

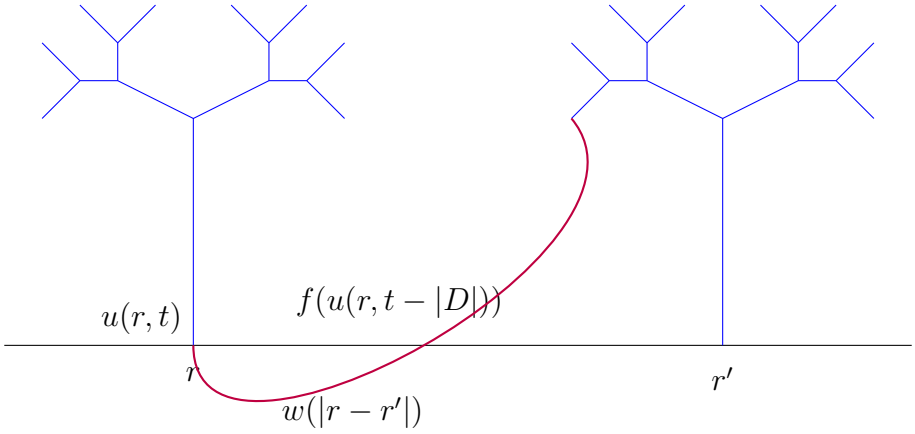


Figure 1.2: A simplification of the connection between two neurons, and showing the average membrane potential u, the average connectivity strength w and the activation rate f in presence of a delay D, the time delay due to finite transmission speed. This graph is presenting a simplified structure of neurons which are usually part of a complex network.

synapse, and there is some time before the neuron in r' initiates an action potential itself. The finite transmission speeds in axons, synapses, and dendrites cause a functionally significant delay in neural system which is represented in equation (1.6). We note that most of the previous important work and results in this topic studied the non- delayed neural field in a general case ignoring any delay endowed with the firing rate and employing the transmission velocity $v = \infty$. The effect of introducing a delay has been studied in several resources. Many of the studies are following the introductory book [34]. In their book, Diekmann and et al. introduce a sun-star calculus for delay differential equations, which opened the door for further development. In our work, we follow their treatment of the initial condition and the equation (1.6) for $t \geq 0$. In addition to the general study of delay equations, there are some recent studies considering the delay precisely for neural field equations.

Since 2000, the study of delay neural field equations has grown rapidly. Hutt et al. [51] studied the stability analysis of a neuronal field model. In 2005, Atay and Hutt [6, 50] discussed the stationary and non-stationary bifurcation in presence of delay, in addition to their instabilities. Later, Faugeras and Faye [40] investigated existence and stability properties of the stationary solutions and numerical approximation. They also mentioned some MATLAB solvers which solve this equations

in case of constant delays. On the other hand, Bojak and Liley [15] introduced propagation PDEs to approximate the neural model and provided a discussion of the relation between estimated velocities and their distribution. Then, Faugeras and Veltz provided new conditions for the stability of their stationary solutions depending on the delay [97]. Meanwhile, Van Gils et al. [47] computed normal form coefficients for bifurcation using the functional analytic sun-star framework. Then, Dijkstra et al. [35] provided an expansion to their analysis in the Pitchfork-Hopf bifurcation and Nogaret et al. [77], under consideration of delay, built a model construction method using an optimization technique to assimilate neural data. for more details see [67, 92]

1.1.4 Related Work

In the 21^{st} century, there are many papers on the neural field equation with and without delays. Some of the studies provide a framework for the existence, uniqueness and stability of the solutions of the neural field equation such as [98, 85, 40, 47, 6, 99, 96, 97], while others consider building effective methods to investigate and assimilate the neural field activities, see for example [12, 44, 82, 84, 76, 26] with techniques of *data assimilation* and *inverse problems* applied to the case without delays. In addition, recent introductory books into inverse and data assimilation methods [26, 76] provide chapters on neural fields with techniques of data assimilation and inverse problems applied to the case without delays, where [76] contains sample codes for the inversion.

The interest in using inverse problem and data assimilation techniques in neuroscience is increasing rapidly. Inverse problem methods are effective in various applications such as medical imaging. Hence, the concept of studying the clinical and experimental data to understand the physiology of neural system is an *inverse problem*, because it is looking for the cause by studying the effects. Inverse problems techniques provide powerful tools to determine the unknown parameters of EEG/MEG modeling. EEG technology is valuable especially in clinical diagnosis. It has an advantage of capturing the time course of brain activity despite the lack of localizing the signals, and it is improving due to the advancement of digitization see [72, 104, 9]. Further, the ability of data assimilation techniques to initialize predictions brings a lot of potential to neural systems. Further, data assimilation techniques are used to estimate unobserved variables and unknown parameters of neural models see for details [74, 73].

An important and significant investigation on the solvability and inversion of the neural field equation has been carried out by Potthast and beim Graben.

They firstly studied the neural field equation (1.4) under general assumptions on the kernel and the activation function. that using the Banach fixed point theorem existence of global solutions can be carried out in an elementary way [85]. They also described how the regularity of the integral kernel w leads to the ill-posedness of the inverse problem, in the sense that for continuous or weakly singular kernels the equations for w are ill- posed.

The ill-posedness of the problem means either its solution does not exist, the solution is not unique, or it is not a continuous function of the data, i.e. if small error of the data can cause a large error of the results. However, they remark that their assumptions are not applicable with a Heaviside-activation function. They provide some reasoning of how to obtain results in this case using regularity on the kernel function w, see [85, 12] for more details. In addition, Coombes and Schmidt [28] dealt with approximating stationary and traveling wave solutions of neural field model. They studied some specific smoothed Heaviside functions, as a firing rate approximation. They highlighted the success of the method by comparing the results with direct numerical simulation of the field. They also suggested that applying an appropriate fixed point theorem could help prove the convergence of such a scheme. Later, Oleynik et al. [81] introduced two iterative schemes for localized stationary solutions of Wilson-Cowan model with a smooth firing rate function. The first one is formulated as a fixed point approach while the second is built on the excitation width of a bump. In addition to the work of [28], they studied convergence of their approaches.

In [82] Potthast stated that applying inverse problems techniques on neural field equations aims to determine either the connectivity kernel w or both the kernel and parameters of the activity function f. Furthermore, it can aim to determine the parameter η. The inverse problem to find f is a nonlinear inverse problem, but it can be reformulated as a problem to find an unknown steepness parameter a constant in the parametrization of the function f.

Recently, Nogaret et al. [77] built a model construction method using an optimization technique to assimilate neural data to determine parameters in a detailed neural model including delay.

The existence of solutions to the delayed neural field equation (1.6) has been investigated in several papers already, see [2, 85, 40]. The aim of this thesis is following the previous results of the *neural field equation* without delay, inspired by the current interest and promising results in the delayed case.

1.1.5 The contribution of this work

As we can see from the previous research results in Section 1.1.4, a challenge often encountered in the study of living systems is to estimate a *spatial connectivity kernel* w. In a neural system, this connectivity kernel usually corresponds to the synaptic footprint, i.e., the connections from a neuron to others by synapses forming between its branching axon and their dendritic trees. Typically, measurements are available for the activity function u at particular spatial locations, e.g.,

- where neurons are patch-clamped or
- electrodes are placed in the extracellular medium.

The task then becomes to derive the spatial connectivity from these experimental data. This approach limits the estimation of connectivity to the set of spatial locations of measurements.

In our Chapter 3, we propose to improve this conventional approach by studying the inverse problem where the *full activity function* u is given at each location in a given spatial domain and the underlying spatial connectivity is derived. In this step, we focus on the problem to reconstruct the kernel w when u is known.

The problem of having limited measurements is part of subsequent work in Chapter 4 combining inverse techniques with state estimation techniques. This second key task is to determine *initial states* for simulations from measurements, such as EEG measurements carried out on the human scalp or during surgery. Clearly, the measurements do not determine fully the underlying excitation fields, even if these are already smoothed out by a neural field model approach. The task to determine states of a dynamical problem is known as the *data assimilation problem*, see for example [76]. Based on a dynamical model and its structural information (e.g. the neural kernel under consideration), we obtain forecasts of the system dynamics starting from the estimated initial value and carrying out a numerical simulation of the dynamical system.

Usually, data assimilation algorithms are based on some initial guess for the state under consideration which is calculated from an earlier state estimate using the dynamical model.

- Here, the data assimilation problem is based on the solution to the inverse problem to reconstruct structural information about the dynamical system.
- On the other hand, the inverse problem as formulated in the above literature uses the dynamical behavior of the states, i.e. it relies on the state estimate.

Altogether, we obtain a coupled *inverse data assimilation problem*.

We work out the above approach to the reconstruction of neural connectivity based on measurement data and the forecasting of neural activity based on reconstructed kernels and states estimated from measured data. Measurement data usually are functions of the neural excitation fields. For example, an electrode measures the integrated activity of a large number of neurons. In general, we are aiming to build a complete iterative procedure of applying inverse problem and data assimilation technique for delay neural network activity. We structure our work into separate steps. Firstly, inspired by promising results studying delayed neural field equations as provided in [96, 40, 99], and secondly, taking into account and building on the previous studies of inversion theory and algorithm introduced by Potthast et al. [85, 12, 82], we develop results for *delay* neural field equations. Here, we are taking advantage of applying regularization approaches to deal with the ill-posedness of inverse problem.

Our second step is to apply data assimilation techniques to estimate the state. In this procedure, we will employ a 3D-VAR type algorithm for data assimilation, based on a (potentially rough) approximation to the dynamical system, which is then used to calculate better structural information by solving the inverse problem. 3D-VAR or 3-dimensional variational assimilation is well-known as an effective tool in numerical weather prediction. As a variational approach, 3D-VAR used to assimilate and analyse observed quantities from different sources and to impose a dynamic balance explicitly through the use of balance equations. We will discuss it in details in Chapter 4.

Later,we built an iterative process by applying the approach of Potthast and beim Graben (2009) for the inverse problem and 3D-Var to estimate the state. This process can be repeated and, under suitable conditions, converges to the full solution of the coupled problem. In addition, numerical results are carried out for the delay neural field equations, developing a complete iterative technique which will use both inversion and data assimilation theory to reconstruct the delay neural field.

1.2 Outline of This Thesis

The thesis is structured into seven chapters, with research contributions focussed within Chapter 3, 4 and 5 and numerical examples in Chapter 6.

We first study the reconstruction of the kernel (or connectivity) w from the full excitation field, i.e. the full neural field u. Then, we go to a more realistic problem

where we do not have the full field u, but we still need to construct the kernel w. Assuming we are given a function Hu of u, we first reconstruct u from Hu using *data assimilation techniques*, where the observation operator H is linear. Then, we develop the idea to iterate this estimation and reconstruction We investigate the convergence of this iterative approach.

1. Chapter 1 is the introduction into the applications and techniques of this thesis, including an overview of the history of development of neural modeling, its biological and mathematical aspects, followed by the related previous research of neural field studies.

2. Chapter 2 collects fundamental theories, basic methods and results for further use in the subsequent chapters.

3. In Chapter 3, we study the reconstruction of the kernel (or connectivity) w from the full excitation field, i.e. the full neural field u. First, we provide the existence proof of the delayed neural field equation. Then, we provide the formulation of the integral equations, the regularization method and the sensitivity analysis.

4. In Chapter 4 we go to a more realistic problem. We actually do not have the full field u but we still need it to construct the kernel w. Assuming we are given some measurements as a function Hu of u with some linear so-called observation operator H, we first reconstruct u from Hu using *data assimilation*. We describe a variational approach to data assimilation in a neural field setup, which is basically a three-dimensional variational assimilation 3D-VAR equipped with a Gaussian covariance matrix.

5. Chapter 5 is the study of the convergence of our iterative approach of inversion and data assimilation.

6. Chapter 6 shows our results in *numerical examples* which is developed for the delayed neural field equation, its state estimation and the kernel reconstruction problem. Those examples were built on the theoretical discussions from the main Chapters 3, 4 and 5.

7. In Chapter 7, we provide our conclusion and further possible studies building on our work and results. We summarize our thesis and provide the conclusion and further possible studies building on our work and results.

Chapter 2

Basic Material and Method

Our goal of building an iterative procedure based on studying of the kernel reconstruction and the state estimation for the delayed neural field equation requires a knowledge of mathematical and functional analysis. That is fundamental to clarify the problems, capture their properties and analyze their solutions. This chapter is a presentation of essential material and tools from mathematical and functional analysis which is needed as a basis for the discussion of inverse problems and the study of data assimilation in the following chapters. The material of this chapter can be found in most of introductory books in functional and numerical analysis. To support our discussion, this material has been reviewed from different resources, see [22, 61, 83, 76].

2.1 Basic Functional Analysis

The purpose of this section is laying ground to the theory and methods of subsequent chapters. We start with the definitions of vector and normed spaces, compactness and completeness, alongside with the definitions of fixed points and the Fréchet derivative. These tools are required to understand the discussion in Chapter 3.

2.1.1 Vector Spaces

The vector space is one of the essential concepts that is required for further basic tools in next subsections and later for our discussion. We note that in particular the data assimilation community usually works with a focus on $\mathbb{R}^n$, such that here we provide some basic material to make our research more widely accessible.

DEFINITION 2.1.1. *Let X be a set that is closed under finite vector addition and scalar multiplication. In order for X to be a vector space, the following axioms must be satisfied for all vectors $x, y, z \in X$ and scalars $r, s \in \mathbb{R}, \mathbb{C}$:*

- *Commutativity*
$$x + y = y + x$$
- *Associativity of Vector Addition*
$$(x + y) + z = x + (y + z)$$
- *Additive Identity*
$$0 + x = x + 0 = x$$
- *Existence of additive inverse*
there exists a $-x \in X$ such that
$$x + (-x) = 0$$
- *Associativity of Scalar Multiplication*
$$r(sx) = (rs)x$$
- *Distributivity of Scalar Sums*
$$(r + s)x = rx + sx$$
- *Distributivity of Vector Sums*
$$r(x + y) = rx + ry$$
- *Scalar Multiplication Identity*
$$1x = x$$

Based on the concept of vector spaces, we next need to introduce the concept of normed spaces to carry our discussion on the Banach fixed point theorem.

2.1.2 Normed Spaces

To review the concept of normed spaces, we need to have a knowledge of norms. Since norms are an effective tool to measure the distance between functions, they are essential to study the difference between the exact and approximation solutions.

Definition 2.1.2. *Let X be a vector space. A norm on X is a real-valued function, denoted by $||.||$, which satisfies the following properties for vectors $x, y \in X$ and scalars $r \in \mathbb{R}, \mathbb{C}$:*

- *Positivity*
$$||x|| > 0$$
- *Definiteness*
$$||x|| = 0 \text{ if and only if } x = 0$$
- *Homogeneity*
$$||rx|| = |r|||x||$$
- *Triangle Inequality*
$$||x + y|| \leq ||x|| + ||y||$$

A vector space combined with a norm is called a normed space.

2.1.3 Convergence, Compactness and Completeness

In this part we come to the definition of convergence, compactness and completeness. We need these concepts in our further basic knowledge in next section as well as for our study in Chapters 3 and 5.

Definition 2.1.3. *A sequence $[x_n]$ in a normed space is called to converge to a point x if*

$$\lim_{n\to\infty} ||x_n - x|| = 0,$$

$n \in \mathbb{N}$.

Definition 2.1.4. *A subset U in a normed space X is called* compact *if each sequence in U has a subsequence that converges to a point in U.*

Definition 2.1.5. *A Cauchy sequence is a sequence $[x_n]$ in a normed space X satisfying*

$$\lim_{n\to\infty} \sup_{i\geq n, j\geq n} ||x_i - x_j|| = 0,$$

.

Definition 2.1.6. *If every Cauchy sequence in the space X is convergent, then the space X is called* complete.

A complete normed space is called a *Banach space*, which is one of the most important structures for our theoretical analysis in Chapter 3.

2.1.4 The Banach Fixed Point Theorem

The Banach fixed point theorem is an affective tool to study the existence and uniqueness of solutions to dynamical systems and also to find fixed points of certain self-maps of normed spaces. We are using it to prove the existence and uniqueness of the solution to the neural field equation in Section 3.2.

Definition 2.1.7. *Let U be a subset of a normed space X. An operator $A : U \to X$ is called a contraction operator if there exists a constant $q \in [0, 1)$ such that:*

$$||Ax - Ay|| \leq q||x - y||,$$

for all $x, y \in U$.

Each constant q satisfying this inequality is called a *contraction number* of the operator A.

Definition 2.1.8. *An element x of a normed space X is called a* fixed point *of an operator $A : U \subset X \to X$ if*

$$Ax = x.$$

Theorem 2.1.9 (Banach Fixed Point Theorem.). *Let U be a complete subset of a normed space X and let $A : U \to U$ be a contraction operator. Then A has a unique fixed point.*

Proof. This proof is from [61], p.44. We recommend this book for more details. Starting from an arbitrary element $x_0 \in U$ we define a sequence $(x_n)_{n\in\mathbb{N}}$ in U by the recursion

$$x_{n+1} := Ax_n, n = 1, 2, ...$$

Then, we have

$$||x_{n+1} - x_n|| = ||Ax_n - Ax_{n-1}|| \leq q||x_n - x_{n-1}||,$$

and from this we deduce by induction that

$$||x_{n+1} - x_n|| \leq q^n ||x_1 - x_0||, n = 1, 2, ...$$

Hence, for $m \geq n$, by the triangle inequality and the geometric series it follows that

$$\begin{aligned} ||x_n - x_m|| &\leq ||x_n - x_{n+1}|| + ||x_{n+1} - x_{n+2}|| + ... + ||x_{m-1} - x_m|| \\ &\leq (q^n + q^{n-1} + ... + q^{m-1})||x_1 - x_0|| \\ &\leq \frac{q^n - q^m}{1-q}||x_1 - x_0|| \\ &\leq \frac{q^n}{1-q}||x_1 - x_0||. \end{aligned} \tag{2.1}$$

Since $q^n \to 0, n \to \infty$, this implies that (x_n) is a Cauchy sequence, and therefore because U is complete there exists an element $x \in U$ such that $x_n \to x, n \to \infty$. Finally, the continuity of the contraction operator A yields

$$\begin{aligned} x &= \lim_{n\to\infty} x_{n+1} \\ &= \lim_{n\to\infty} Ax_n \\ &= Ax, \end{aligned} \tag{2.2}$$

i.e. x is a fixed point of A. □

This theorem is important for different iterative approaches. In our work, this theorem is an essential tool in our proof of the solvability of our direct delayed neural field problem in Section 3.2.

2.1.5 The Fréchet Derivative

In this part we introduce the Fréchet differentials as an example of derivatives for mappings between Banach spaces which we used for our later discussion in Section 3.5. Fréchet differentials are more general than partial derivatives because they are an extension of the derivative from real-valued functions to a general normed space. We start with the definition of the bounded linear operator.

DEFINITION 2.1.10. *Let X and Y be normed linear spaces, and let A be a map from X to Y. A is linear if:*
(a) $A(x_1 + x_2) = Ax_1 + Ax_2$,
(b) $A(\alpha x) = \alpha Ax$
for all $x_1, x_2, x \in X$.

A is bounded if there exists $C > 0$ such that

$$||Ax|| \leq C||x||$$

for all $x \in X$.

DEFINITION 2.1.11. *Let $f : U \to Y$ be a mapping from an open set U in a normed space X into a normed space Y. Let $x \in U$, if there is a bounded linear map $A : X \to Y$ such that*

$$\lim_{||h||\to 0} \frac{||f(x+h) - f(x) - Ah||}{||h||} = 0,$$

uniformly for all directions, then f is called Fréchet differentiable *at x, A is called the* Fréchet derivative *of f at x. If f is differentiable at all $x \in U$, we say that f is Fréchet differentiable in U.*

The concepts introduced in this section are available in any introductory book on functional analysis. For more details we refer to [22, 61, 83, 76].

2.2 Methods of Approximation

2.2.1 Collocation Method

Since fifty years ago, the collocation method has been developed to be an effective tool to solve ordinary and partial differential equations, integral equations and integro-differential equations. This method is a weighted residual method approximating the solution of an equation at a finite number of points called the collocation points using a linear combination of trial functions which are usually chosen to be polynomials or cubic splines. The advantages of this method are :

- its simple concept,
- its broad applications,
- its ease to implement.

Let us look at the collocation method for approximately solving a linear operator equation

$$Au = g \tag{2.3}$$

with an operator A between Banach spaces X and Y, $u \in X$ and $g \in Y$. To approximate the value of the solution u by a finite series approximation $\tilde{u}$, we assume that

$$u \cong \tilde{u} = \sum_{i=1}^{n} l_i V_i, \tag{2.4}$$

where $l_i, i = 1, 2, ..., N$ are unknown coefficients. The set of functions $V_i, i = 1, 2, ..., N$ are the selected trial functions which satisfy any conditions associated with the equation (2.3), for example boundary conditions. By substituting the approximate solution $\tilde{u}$ into the equation (2.3), we obtain the residual in the form:

$$R = A\tilde{u} - g = \sum_{i=1}^{n} l_i AV_i - g. \tag{2.5}$$

If $R = 0$, then $\tilde{u}$ is the exact solution. Otherwise, the l_i's need to be chosen to make $|R|$ as small as possible. If the modulus $|R|$ of R is minimal, then the equation (2.4) is the best approximation. To achieve this goal, we could try to minimize the integral of $|R|$

$$\int_t \int_V |R| \, dV dt = \min. \tag{2.6}$$

To satisfy (2.3) approximately and determine the n unknown coefficients l_i, $i = 1, ..., n$, we take the scalar product of (2.3) with functions w_j, which in the case of linearly independent functions $\{w_j, j = 1, ..., N\}$ is leading to N equations

$$\sum_{i=1}^{n} l_i \int_t \int_V w_j AV_i \, dV dt = \int_t \int_V w_j g \, dV dt, \quad j = 1, 2, ..., N. \tag{2.7}$$

This equation is describing the weighted residual methods. For the collocation method, the weighting functions w_i's are chosen to be the Dirac delta function. This choice makes the integration vanish and guarantees N equations to evaluate the N unknown coefficients l_i's by only calculating the residual at the selected points in the domain. For more details see [61, 65, 22, 76].

2.3 Fundamental Knowledge of Regularization

In this section, we start with introducing the meaning of ill-posedness. Then, we provide the main concept of regularization techniques as a method to deal

with the instability of ill-posed equations. In particular, we pay attention to the Tikhonov regularization scheme, as we are using it in Section 3.4.2.

2.3.1 Ill-Posedness Problems

There is an essential difference between direct and inverse problems. Inverse problems are often *ill-posed problems*. This term is related to three properties defining well-posed problems as postulated by Hadamard in 1923. These properties are formulated mathematically in our subsequent definition.

DEFINITION 2.3.1. *Let X and Y be normed spaces, $A : X \to Y$ an operator. The equation*

$$A\phi = f, \tag{2.8}$$

is called well-posed *if it satisfies:*

1. *Existence of a solution. For every $f \in Y$ there is a solution $\phi \in X$ for (2.8).*
2. *Uniqueness of the solution. For every $f \in Y$ there is at most one solution $\phi \in X$ for (2.8).*
3. *Stability or continuous dependence of the solution on the data. The solution ϕ depends continuously on f; that is, for every sequence*

   $$(\phi_n) \subset X \text{ with } A\phi_n \to A\phi \text{ for } n \to \infty,$$

   it follows that

   $$\phi_n \to \phi \text{ for } n \to \infty.$$

Any problem does not satisfy at least one of these properties is called *ill-posed*. For more details we refer to the books [76, 59].

2.3.2 Regularization Schemes

Solving ill-posed problems needs to apply a regularization method to approximate its solution. In this section we introduce the concept of requlariza tion.

In the equation (2.8), the right-hand side f is usually given by some measurements. Let it be a function $f^{(\xi)} \in Y$ with

$$||f - f^{(\xi)}|| \leq \xi, \tag{2.9}$$

where ξ is the measurement error.

Assume that the inverse of the operator A is unbounded. This assumption means that the solution of (2.8) is unstable with respect to variations of the right-hand side, which is an ill-posed case described in Definition 2.3.1. For unbounded A^{-1} there is a sequence $\psi_n \in Y$ with $||\psi_n|| = 1$ such that

$$\phi_n := A^{-1}\psi_n$$

satisfies

$$||\phi_n|| \to \infty, \text{for } \ n \to \infty,$$

the solution $\phi^{(\xi)}$ in the equation

$$A\phi^{(\xi)} = f^{(\xi)}, \tag{2.10}$$

can have an arbitrary distance to the true solution ϕ, depending on the choice of $f^{(\xi)}$.

Let us define

$$f_n^{(\xi)} := f + \psi_n, \ \phi_n^{(\xi)} := \phi + \phi_n, \tag{2.11}$$

for arbitrary fixed $\xi > 0$, and obtain solutions of (2.10), (2.11) with

$$||\phi_n^{(\xi)}|| \to \infty, \ n \to \infty. \tag{2.12}$$

The idea of solving the ill-posed equations is trying to find a bounded approximation R_α to the unbounded operator A^{-1} depending on some parameter α which, when the observation error ξ tends to zero or is zero, guarantees the convergence of the corresponding approximate solution ϕ_α to the true solution of the ill-posed equation.

DEFINITION 2.3.2. *Let $A : X \to Y$ be an injective bounded operator between normed spaces X, Y. A regularization scheme is a family of bounded linear operators $R_\alpha : Y \to X, \alpha > 0$, such that*

$$R_\alpha A\phi \to \phi, \ \alpha \to 0, \tag{2.13}$$

for all $\phi \in X$.

The limit (2.13) means that R_α tends to A^{-1} pointwise, but the convergence does not hold in norm, compare [76]. The construction of the function ϕ has an error. This error is estimated from:

$$\phi_\alpha^{(\xi)} = R_\alpha f^{(\xi)} \text{ with } ||f^{(\xi)} - f|| \leq \xi, \tag{2.14}$$

by

$$\begin{aligned} ||\phi_\alpha^{(\xi)} - \phi|| &= ||R_\alpha f^{(\xi)} - R_\alpha f + R_\alpha f - \phi|| \\ &\leq ||R_\alpha f^{(\xi)} - R_\alpha f|| + ||R_\alpha f - \phi|| \\ &\leq ||R_\alpha||\xi + ||R_\alpha A\phi - \phi||. \end{aligned} \tag{2.15}$$

For the second term, the estimation does not depend on the measurement error, but only on the approximation of A^{-1} by R_α. This is called the regularization error, which is arising from the approximation of the operator A^{-1}.

The first term involves the data error terms of size ξ and is called the data error. It is magnified by the application of R_α. However, because of the boundedness of the operator for every fixed $\alpha > 0$, its influence can be controlled.

These errors have a characteristic behaviour. The regularization error tends to zero for $\alpha \to 0$, which means the approximation tends to the true solution if no observation error is present. On the other hand, the data error tends to infinity for $\alpha \to 0$, depending on a particular measurement $f^{(\xi)}$. For further investigation of the convergence, we need to consider $\xi \to 0$. In this case, the appropriate regularization parameter $\alpha = \alpha(\xi)$ needs to be chosen to ensure that the approximate solution $\phi^{(\xi)}$ tends to ϕ by keeping the data error controlable and tend to zero as well when $\xi \to 0$.

DEFINITION 2.3.3. *The function $\alpha = \alpha(\xi)$ is called a strategy for a regularization scheme R_α if $\alpha \to 0$ for $\xi \to 0$. Such a strategy is called regular, if*

$$R_{\alpha(\xi)} f^{(\xi)} \to A^{-1} f, \ \xi \to 0, \tag{2.16}$$

for each $f^{(\xi)}$ with $||f^{(\xi)} - f|| \leq \xi$

We refer to the literature, in particular ch.3 p107 of [76].

2.3.3 Tikhonov Regularization.

In our work, we apply the Tikhonov requlariza tion method to approximately solve our ill-posed kernel reconstruction problem. In the next theorem, we provide an introduction of the method as a preparation for Chapter 3.

THEOREM 2.3.4. *Let $A : X \to Y$ be an injective compact operator between Hilbert spaces X, Y. Then for each $\alpha > 0$ the operator $\alpha I + A^*A$ is boundedly invertible and the operator*

$$R_\alpha := (\alpha I + A^*A)^{-1}A^*, \tag{2.17}$$

where A^* is the adjoint operator of A. This defines a regularization scheme for A with

$$||R_\alpha|| \leq \frac{1}{2\sqrt{\alpha}}, \tag{2.18}$$

known as Tikhonov regularization.

Instead of solving the equation (2.8), the Tikhonov regularization solves the approximation equation:

$$\alpha\phi_\alpha + A^*A\phi_\alpha = A^*f, \tag{2.19}$$

which is the equation (2.8) after multiplying by A^*, then adding $\alpha\phi$.

This method keeps the residual $||A\phi_\alpha - f||_2^2$ small and stable by keeping ϕ_α small using the term $\alpha||\phi_\alpha||_2^2$, see [76, 61].

To summarize, this chapter introduced general analytical knowledge and important tools that lays the ground for our further discussion and the solution of the inverse neural field problem.

Chapter 3

Inverse Delay Neural Field Equation

Understanding the dynamics and architecture of the brain is the way to understand its functions and behaviour which is the role of neural modeling. Since the early work of Wilson, Cowan, Nunez and Amari [102, 103, 78, 54] in the 1970s neural field models have become an effective tool in computational neuroscience and mathematical biology. Neural models describe essential properties and relate them to the experimental results of different recordings such as EEG, MEG and fMRI.

Neural networks belong to the class of complex systems. Each neural tissue has a large variety of properties in space and time. That makes simplifying of its analysis very challenging due to the large number of 100 billion neurons and their connectivity in the human nerve system, where each of these neurons is a complex biological structure itself. Many researchers have discussed the neural field and its structure and properties, see [17, 14, 26].

This chapter considers neural field models that involve delayed spatial interactions and where the delay may depend on the distance between spatial locations [6, 96, 49]. We will assume that the delay function $D(r, r')$ between spatial locations r, r' is known. For instance, this is the case when the delay is linked to the *geometry* of the problem, e.g., when $D(r, r') \sim ||r - r'||$, the distance between the points r and r' in some domain Ω. This assumption is common in practice, since for direct neural connections the delay is essentially the distance divided by the signal propagation speed, which as a basic approximation can be assumed to be a universal constant. Clearly, it would be desirable to use more biologically realistic propagation fields, but this task would be beyond the scope of this work and here we work with the constant speed approximation.

Neural field models consider spatially non-local interactions, which may be expressed equivalently either by partial differential, integro-differential or integral equations [49, 30]. In this century, there are many papers on the neural field equation with and without delays. Some of the studies provide a framework for the existence, uniqueness and stability of the solutions of the neural field equation such as [98, 85, 40, 47, 6, 99, 96, 97], while others consider building effective methods to investigate and assimilate the neural field activities, see for example [12, 44, 82, 84, 76] with techniques of *data assimilation* and *inverse problems* applied to the case without delays. Recently, Nogaret et al. [77] built a model construction method using an optimization technique to assimilate neural data to determine parameters in a detailed neural model including delay.

We start the chapter with introducing the idea behind applying inverse problem techniques to the *delayed neural field equation*. In the second section, we will show how the methods used in [85] can be extended to study existence and stability of solutions in a neural field model with delay. Potthast and Graben study the solvability of Amari equation 1.4. For smooth firing rate functions f, they show the existence of global solutions using the Banach fixed point theorem [85]. We follow their procedure to prove the solvability of the delayed Amari equation. The basic idea is to split the integral operators under consideration into parts with positive and negative temporal arguments to deal with delay. Using this step, we have two operators. The positive operator where $t > 0$ is the same in [85] approach. The negative one presents the delay where we apply the initial condition. As a result we obtain a direct and flexible basic existence proof for a delay neural field equation, which includes a constructive method based on integral equations only. These results have been derived by other authors with more sophisticated technique. Studying the solutions of delayed neural field equations, Faye and Faugeras investigated the existance and uniqueness of the solution using the theory of delay differential equations and for stability, they use Lyapunov analysis. More authors contribute to bifurcation theory. For example, Van Gils et al. [47] investigate the stability and the bifurcation of steady state using the theory of sun-star calculus framework. Atay and Hutt in [6] analyze the stability of the solutions giving sufficient conditions, while Dikstre et al. show how symmetry arguments and residue calculus can be used to simplify the computations. It is non-trivial that the arguments used for neural fields without delay are applicable to the delay case, and the approach in Section 3.2 to show the existence and uniqueness of the solution based on several relatively simple functional analytic arguments, is of interest by itself.

In Section 3.3, we define both the direct and inverse problem for delayed neural field. Then, we will show in Section 3.4 that the kernel reconstruction problem

for the delayed neural field equation can be reformulated into a *family of integral equations* of the *first kind*. When several trajectories of neural activity are given, the family of integral equations is vector valued. This turns out to be an ill-posed problem, for smooth neural activity it is even exponentially ill-posed. To formulate stable numerical methods for its solution, we need to employ regularization. Here, we use a spectral approach to classical Tikhonov regularization, see [37, 48, 62], we later provide an analysis of ill-posedness of the kernel reconstruction approach.

Finally, considering the importance of the sensitivity, we add a sensitivity analysis. We study the *sensitivity* of the mapping $u \mapsto w$ of the excitation function u to the reconstructed kernel w in Section 3.5 showing that the regularized version of the kernel w is Fréchet differentiable with respect to the excitation function u. Further, we explicitly derive the derivative by means of integral equations.

3.1 Introduction to Inverse Problems.

Since the last century, inverse problems started to be very popular in modern science because of their important applications in different areas of physics and industrial mathematics. Although the first appearance of the term 'inverse problems' was in 1960s. The inversion idea is dated in the history to different examples.

- The reconstruction of the orbit of a comet from earlier data using the method of least squares by Gauss in 1800
- The discovery of the planet Neptune in 1846 by Le Verrier using computations of the movement of Uranus was a solving of inverse problem.
- The transformation that was studied by Radon in 1917 which is the basic of the Xray tomography.

In their use of measured data from the surface to understand the internal behaviour of the Earth, geophysicists contributed significantly to the development of the inversion theory. The pioneering work of Backus and Gilbert [7, 8] provided a start of the fame of inverse problems. In 1980s, Tarantola [94] contributed to the theory studying geophysical data by means of probabilistic models, see the new version of his book *Inverse Problem Theory [93].*

In fact, the topic of inverse problems is a connection between the general mathematical theory and special applications. In some dynamical systems, the knowledge about parameters and properties is insufficient to apply forward problem

techniques in real-world settings. In this case, using the inversion is necessary to estimate the missing state or parameters of the system based on a set of measurements.

Inverse problems aim to reconstruct the model or parts of it from a set of measurements. These could be any known measurements that have been taken previously or at different time steps. Solving the inverse problem means to find the causes from their effects. One can also say that inverse methods are estimating constituents of models for which direct measurements are not available. The broad utilization of inverse problems in practical applications led to new approaches to inversion and to innovative regularization tools. Furthermore, as part of the need to reconstruct parts of dynamical systems and estimate system states, the importance of *data assimilation* techniques arises.

Inverse problems have universal techniques which can be used in general. However, some are specific and applicable to particular applications. That makes the knowledge of the application and the properties, settings and environment of solutions is very important. Nowadays, there are numerous applications of inverse problems such as in geophysics, meteorology, biomedical engineering, imaging, civil engineering and mechanical engineering. In mathematical neuroscience, instead of studying the dynamics of neural activity as described by equation (1.4) itself, neural field inverse problems are studying the reconstruction of either the connectivity kernels w or both the kernel w and parameters of the activation function f. Also, the inversion can determine the parameter distribution of the temporal parameter τ. For more details we refer to [44, 76, 93, 5, 91, 55].

3.2 Solvability of Delay Neural Field Equation

Here, we study the neural field equation (1.6) on some bounded domain $\Omega \subset \mathbb{R}^m$ in a space with dimension $m = 2$ or $m = 3$. We assume that the transmission delay $D(r, r')$ of neural excitation or inhibition between r' and r is bounded on $\Omega \times \Omega$, i.e. there is a constant c_T such that

$$|D(r, r')| \leq c_T, \quad r, r' \in \Omega. \tag{3.1}$$

At time $t \in \mathbb{R}$, the neural fields $u(r, t)$ at a point $r \in \Omega$ might receive excitations from the past propagation with a maximal delay of c_T. Working on the interval $[0, \rho]$ the fields need to be given on $[-c_T, 0]$ to start some simulation. It is suggested

that this will be the same value of the initial condition. For more details about delays see [34].

The *initial condition* for the delay neural field equation is given by

$$u(r,t) = u_0(r,t), \quad (r,t) \in \Omega \times [-c_T, 0]. \tag{3.2}$$

In this section, we first need to lay the ground for our further inverse and sensitivity analysis by providing a basic discussion to show the solvability and the uniqueness of the solution of equation (1.6) using tools from functional analysis and integral equations.

Our investigation is under *smoothness assumptions* upon the activity function f and the connectivity kernel w. We consider a *continuous* activation function $f(s)$ for $s \in \mathbb{R}$ and an activation threshold η which is introduced in equation (1.2). This function, which can be interpreted as the mass action probability of neuron firing if its membrane potential is over the threshold, comes from a stochastic neuron model, see [17, 52]. Often, see for example [26], f in this case is approximated by the sigmoidal function as in equation 1.3. Here, we will work with general Lipschitz continuous functions f. We assume that the kernel w satisfies

(H1) $w(r,\cdot) \in L^1(\Omega), \quad \forall\, r \in \Omega \subset \mathbb{R}^m$

such that we obtain a *well defined* integral of the form

$$g(r,s) := \int_\Omega w(r,r') f(u(r', s - D(r,r')))dr', \quad r \in \Omega, \ s \in \mathbb{R},$$

The condition

(H2) $\sup_{r\in\Omega} \|w(r,\cdot)\|_{L^1(\Omega)} \leq C_1$

with some constant C_1 leads to g being *bounded* on $\Omega \times \mathbb{R}$. We need $g(r,s)$ to be *continuous* in dependence of r and s, which for continous functions u and D is achieved by the additional condition

(H3) $\|w(r,\cdot) - w(r^*,\cdot)\|_{L^1(\Omega)} \to 0$ for $|r - r^*| \to 0$.

Now, existence is given by the following result.

THEOREM 3.2.1 (Existence). *If the kernel w satisfies* (H1)-(H3)*, and if the delay term D is bounded continuous, i.e., if we have $D \in BC(\Omega \times \Omega, \mathbb{R}^+)$, then for any $T > 0$ and for any initial field u_0 as given by the initial condition (3.2) there exists a unique solution $u \in C^1(\Omega \times [0,T])$ to the delay neural field (1.6) on $[0,T]$.*

Proof. We first need some preparations. We will need to split the function $u(r, s - D(r, r'))$ into the part where the time variable $t = s - D(r, r')$ is in $(0, T]$ and where $t = s - D(r, r')$ is in $[-c_T, 0]$. This is carried out by defining

$$\chi_+(r,t) := \begin{cases} 1, & t > 0, \\ 0, & t \leq 0, \end{cases} \tag{3.3}$$

and $\chi_-(r,t) := 1 - \chi_+(r,t)$. The function χ_- is equal to 1 for negative time arguments and we have $1 = \chi_+ + \chi_-$. For studying the existence of solutions of the delay neural field equation (1.6) we define the operators

$$(A_1 u)(r,t) := \int_0^t -\frac{u(x,s)}{\tau}\, ds, \quad r \in \Omega, t \leq 0, \tag{3.4}$$

and

$$\begin{aligned} &(A_2^{\pm} u)(r,t) := \\ &\frac{1}{\tau}\int_0^t \int_\Omega w(r,r')\chi_{\pm}\left(r, s - D(r,r')\right) f\left[u\left(r', s - D(r,r')\right)\right]\, dr'\, ds, \end{aligned} \tag{3.5}$$

for $r \in \Omega, t \in [0, T]$.

By integration with respect to time, the solution of equation (1.6) can be reformulated as

$$\begin{aligned} &u(r,t) - u(r,0) = \\ &\frac{1}{\tau}\int_0^t \left\{ -u(r,s) + \int_\Omega w(r,r') f\left[u\left(r', s - D(r,r')\right)\right]\, dr' \right\} ds \end{aligned} \tag{3.6}$$

for $r \in \Omega$ and $t \in [0, \rho]$ for some interval $[0, \rho]$ with an auxiliary parameter ρ.

We can now split the operators as follows:

$$\begin{aligned} &\frac{1}{\tau}\int_0^t \int_\Omega w(r,r') f\left[u\left(r', s - D(r,r')\right)\right]\, dr'\, ds \\ &= (A_2^+ u)(r,s) + (A_2^- u)(r,s) \\ &= (A_2^+ u)(r,s) + (A_2^- u_0)(r,s), \end{aligned} \tag{3.7}$$

where the last equality is obtained from

$$
\begin{aligned}
\tau(A_2^- u)(r,t) &= \int_0^t \int_\Omega w(r,r')\chi_-(r,s-D(r,r'))f\left[u\left(r',s-D(r,r')\right)\right]\, dr'\, ds \\
&= \int_\Omega \int_0^t w(r,r')\chi_-(r,s-D(r,r'))f\left[u\left(r',s-D(r,r')\right)\right]\, ds\, dr' \\
&= \int_\Omega \int_0^{D(r,r')} w(r,r')f\left[u\left(r',s-D(r,r')\right)\right]\, ds\, dr' \\
&= \int_\Omega \int_0^{D(r,r')} w(r,r')f\left[u_0\left(r',s-D(r,r')\right)\right]\, ds\, dr' \\
&= \int_\Omega \int_0^t w(r,r')\chi_-(r,s-D(r,r'))f\left[u_0\left(r',s-D(r,r')\right)\right]\, ds\, dr' \\
&= \int_0^t \int_\Omega w(r,r')\chi_-(r,s-D(r,r'))f\left[u_0\left(r',s-D(r,r')\right)\right]\, dr'\, ds \\
&= \tau(A_2^- u_0)(r,t)
\end{aligned}
$$

using that $u(r,t) = u_0(r,t)$ for $t \leq 0$. With $A := A_1 + A_2^+$ the delay neural field equation is equivalent to the *fixed point equation*

$$
u(r,t) = u(r,0) + (A_2^- u_0)(r,t) + (Au)(r,t), \quad r \in \Omega \text{ and } t \in [0,\rho]. \tag{3.8}
$$

Here, the function $u(r,t)$ needs to be considered on $\Omega \times [0,\rho]$ only and we can study the fixed point equation in $BC(\Omega \times [0,\rho])$. Any solution to equation (3.8) will be continuously differentiable with respect to time and satisfy the delay neural field equation (1.6).

We now show that for sufficiently small parameter $\rho > 0$ the operator A is a contraction on the space $BC(\Omega \times [0,\rho])$ equipped with its canonical norm

$$
||v||_\rho := \sup_{r\in\Omega,\, t\in[0,\rho]} |v(r,t)|. \tag{3.9}
$$

We will carry out these arguments in four steps I to IV.

I. For the linear operator A_1 given by equation (3.4), and by the definition (2.1.7), we estimate

$$
||(A_1 u)||_\rho = \sup_{r\in\Omega,\, t\in[0,\rho]} |(A_1 u)(r,t)| \leq \frac{\rho}{\tau} \sup_{r\in\Omega,\, t\in[0,\rho]} |u(r,t)| = \frac{\rho}{\tau}||u||_\rho, \tag{3.10}
$$

i.e. the operator A_1 maps the space $BC(\Omega \times [0,\rho])$ boundedly into itself and by equation (3.10) the operator norm is bounded by ρ/τ.

II. We define

$$Ju(r,t) := \frac{1}{\tau}\int_\Omega w(r,r')\chi_+(r,t-D(r,r'))f\Big(u(r',t-D(r,r'))\Big)\,dr', \tag{3.11}$$

for $x \in \Omega$ and $t \geq 0$, and follow [85], Lemma 2.5, to estimate

$$\begin{aligned}|Ju_1(r,t) - Ju_2(r,t)| \leq \frac{1}{\tau}\int_\Omega |w(r,r')|\ \chi_+(r,t-D(r,r'))\cdot \\ \Big|f\Big[u_1(r',t-D(r,r'))\Big] - f\Big(u_2(r',t-D(r,r'))\Big)\Big|\,dr',\end{aligned} \tag{3.12}$$

for $x \in \Omega$ and $t \in [0,\rho]$. First, using the Lipschitz continuity of the function f with Lipschitz constant $L > 0$, using C_1 given in (H2) we obtain

$$\begin{aligned}&|Ju_1(r,t) - Ju_2(r,t)| \\ &\leq \frac{L}{\tau}\int_\Omega |w(r,r')|\ \chi_+(r,t-D(r,r'))\cdot\Big|u_1(r',t-D(r,r')) - u_2(r',t-D(r,r'))\Big|\,dr' \\ &\leq \frac{LC_1}{\tau}\sup_{r'\in\Omega}\Big\{\chi_+(r,t-D(r,r'))\Big|u_1(r',t-D(r,r')) - u_2(r',t-D(r,r'))\Big|\Big\} \\ &\leq \frac{LC_1}{\tau}\|u_1-u_2\|_{\Omega\times[0,\rho]},\end{aligned} \tag{3.13}$$

for $r \in \Omega$ and $t \in [0,\rho]$.

III. Integration of equation (3.13) with respect to $t \in [0,\rho]$ leads to

$$||A_2^+(u_1) - A_2^+(u_2)||_\rho \leq \frac{\rho LC_1}{\tau}\|u_1-u_2\|_\rho. \tag{3.14}$$

where $||\cdot||_\rho$ as defined in equation (3.9). Now, for the operator A we obtain the estimate

$$\begin{aligned}\|A(u_1) - A(u_2)\|_\rho &= \|A_1(u_1-u_2) + A_2^+(u_1) - A_2^+(u_2)\|_\rho \\ &\leq \tfrac{\rho}{\tau}\|u_1-u_2\|_\rho + \tfrac{\rho LC_1}{\tau}\|u_1-u_2\|_\rho \\ &\leq q\|u_1-u_2\|_\rho\ ,\end{aligned} \tag{3.15}$$

with

$$q := \frac{\rho}{\tau}(1+LC_1)\ . \tag{3.16}$$

in the case where ρ is small enough to guarantee that $q < 1$ by equation (3.15), we have shown that A is a contraction on $BC(\Omega\times[0,\rho], ||\cdot||_\rho)$.

IV. According to the Banach fixed point theorem, there is one and only one fixed point u^* for the fixed-point equation (3.8). We have shown the existence of a unique solution $u(x,t)$ for all $t \in [0,\rho]$. Now, the same argument applied to the interval $[\rho,2\rho]$ and subsequent intervals $[2\rho,3\rho]$ etc. in the same way. This leads to the existence and uniqueness result on the interval $[0,T]$. □

Remark. We note that the proof also works when some bounded continuous forcing term $I(r,t)$, $r \in \Omega$, $t \in [0,T]$, is added to the neural field equation (1.6). It leads to an additional term in equation (3.8), for which all arguments remain valid.

It is well-known [62, 76] that Banach's theorem also provides a *constructive* method to calculate the fixed point by *successive iterations*. Let u_1 be a starting function. Then, the sequence defined by

$$u_{n+1} := u_0 + A_2^-(u_0) + A(u_n), \quad n = 1, 2, 3, ... \tag{3.17}$$

converges to the unique fixed point u^*. An error estimate for this iteration process is, based on equation (3.15), obtained from

$$\begin{aligned} \|u_{n+1} - u^*\| &= \|u_0 + A_2^-(u_0) + A(u_n) - (u_0 + A_2^-(u_0) + A(u^*))\| \\ &= \|A(u_n) - A(u^*)\| \\ &\leq q\|u_n - u^*\|. \end{aligned} \tag{3.18}$$

Induction immediately leads to the full error estimate

$$\|u_{n+1} - u^*\| \leq q^n \|u_1 - u^*\|, \quad n \in \mathbb{N}. \tag{3.19}$$

For our numerical calculations we have, however, instead used Runge-Kutta or Euler methods applied to the differential form of the delay neural field equation.

3.3 The Direct and the Inverse Delay Neural Field Problems

The dynamics of the delay neural field $u(r,t)$ with $r \in \Omega$ and $t \in [0,T]$ satisfies (1.6) and its initial condition. Meanwhile, for our neural kernels w we assume that they are in the class $\mathcal{N}$ of functions on $\Omega \times \Omega$ which are C^1 in the first variable and boundedly integrable in the second variable. The delay functions D on $\Omega \times \Omega$ given by $D(r,r') \simeq \tilde{D}(r,r')/v$, where the case $D = 0$ corresponds to the field equation (1.4). More general, we assume that we have some given delay function D on $\Omega \times \Omega$ which is differentiable with respect to its first variable and bounded measurable with respect to its second variable. We call this class of delays $\mathcal{D}$. Since in this work we do not want to carry out the reconstruction of the delay function, we will assume that this is known throughout our further reconstruction and assimilation steps.

Definition 3.3.1 (Direct Delay Neural Field Problem). *Given an initial state $u_0 \in C^1(\Omega)$, a delay function D in class $\mathcal{D}$ and a neural kernel w in the class $\mathcal{N}$, the direct neural field problem is to calculate $u(r,t)$ for $t \in [0,T]$ with some constant $T > 0$ and $x \in \Omega$ as a solution to the integro-differential equation (1.6).*

As shown in [2, 40] and explained in Section 3.2, the direct neural field problem has a unique solution $u(r,t)$ on $r \in \Omega$ and $t \in [0,T]$.

Let us now come to the inverse problem. Of course, there are many possible settings for the inversion process. Here we first follow the most basic problem as formulated by beim Graben and Potthast [12, 86], where the kernel w for Amari equation 1.4 is to be reconstructed from the knowledge of the full time-dependent neural activity field $u(r,t)$ for $r \in \Omega$ and $t \in [0,T]$. They show that the kernel reconstruction problem is ill-posed and suggest a regularization scheme to deal with ill-posedness.

Definition 3.3.2 (Full Field Delay Neural Inverse Problem). *Given the time-dependent neural field $u(r,t)$ for $x \in \Omega$ and $t \in [0,T]$ the* full field neural inverse problem *is to determine the neural connectivity kernel $w(r,r')$ for $r, r' \in \Omega$ given the knowledge of u, such that u is a solution to the neural field equation (1.6) with kernel w and delay D, where we assume that we know the delay D as a function of r, r'.*

The neural inverse problem without delay has been solved first in [12, 86], see also [26, 82]. Further work on dimensional reduction carried out for example by [84] and with a localization method presented in Chapter 7 of [76]. Here, we follow [12] with a transformation of the inverse problem into a family of linear integral equations based on

$$\varphi(r,r',t) := f(u(r,t) - D(r,r')), \quad r,r' \in \Omega,\ t \in [0,T] \tag{3.20}$$

and

$$\psi(r,t) := \tau \frac{\partial u(r,t)}{\partial t} + u(r,t), \quad r \in \Omega,\ t \in (0,T). \tag{3.21}$$

Then, equation (1.6) is transformed into

$$\psi(r,t) = \int_\Omega \varphi(r,r',t) w(r,r')\, dr', \quad t \in [0,T] \tag{3.22}$$

for $r \in \Omega$. If φ has smoothness properties, such as continuity or piecewise continuity, by standard arguments (c.f. [76]) the integral equation is ill-posed and its solution inhibits sincere instabilities. The singular values of the equation (3.22) are

exponentially decreasing, which has been shown and demonstrated in [76], Chapter 7. Using regularization algorithms, beim Graben et al. stabilize the inversion process by a Tikhonov regularization approach.

We define the operator A by

$$(A[r]v)(t) := \int_\Omega \varphi(r, r', t)v(r')\, dr', \quad t \in [0, T] \tag{3.23}$$

for some integrable function v on Ω. Then, for each $r \in \Omega$ equation (3.22) corresponds to the operator equation

$$A[r]w(r, \cdot) = \psi(r, \cdot) \quad \text{on} \quad [0, T]. \tag{3.24}$$

The regularized solution with regularization parameter $\alpha > 0$ according to Tikhonov regularization as discussed in Section 2.3.2 and (c.f. Chapter 3.1.4 of [76]) is given by

$$w_\alpha(r, \cdot) := \Big(\alpha I + A^*[r]A[r]\Big)^{-1} A^*[r]\psi(r, \cdot), \tag{3.25}$$

for $r \in \Omega$. The inversion based on (3.25) has been tested in [12, 84, 76] and in the case of delay are published in or paper [2]. Here, these results are included in Chapter 6.

3.4 The Inverse Problem of Kernel Reconstruction with delays

We now come to the *kernel reconstruction* from given dynamical neural patterns with delay. We first formulate a *regularized kernel reconstruction* approach based on *integral equations* in the following Sections 3.4.1 and Section3.4.2, then we carry out a *sensitivity analysis* in Section 3.5.

3.4.1 Kernel Reconstruction with Delays

Usually, we will observe the dynamical evolution of some pattern for a system under consideration. For neural system, the neural activation patterns are "the joint firing rate of a population of neurons measured during a brief time window", according to [80]. More generally, observations may start from different initial patterns that lead to different dynamical trajectories in the phase space. If we

have N such trajectories, the task is to find the kernel which will predict these trajectories when the N initial conditions are provided. In more details, this section is investigating the *inverse problem* of *kernel reconstruction* for the delayed neural field equation (1.6). We assume that

- the nonlinear *activation function* $f : \mathbb{R} \to \mathbb{R}^+$ to be known and
- the *delay function* $D : \Omega \times \Omega \to [0, c_T]$ to be given.

The task is to find a kernel $w(r, r')$ for $(r, r') \in \Omega$ given the time-dependent neural activation patterns $u^{(\xi)}(r, t)$ for $(r, t) \in \Omega \times [0, T]$ corresponding to initial conditions $u_0^{(\xi)}(r, t)$ for $(r, t) \in \Omega \times [-c_T, 0]$ according to equation (3.2), where $\xi = 1, ..., N$.

Here, we reformulate the inverse problem into a *family of integral equations* of the *first kind* and study their solution by *regularization methods*. As a first step, we define

$$\phi^{(\xi)}(r, t) := f\left[u^{(\xi)}(r, t)\right], \quad (r, t) \in \Omega \times [-c_T, T] \tag{3.26}$$

and

$$\psi^{(\xi)}(r, t) := \tau \frac{\partial u^{(\xi)}}{\partial t}(r, t) + u^{(\xi)}(r, t), \quad (r, t) \in \Omega \times [0, T] \tag{3.27}$$

for $\xi = 1, 2, ..., N$.

With the integral operator W defined by

$$(W\phi)(r, t) := \int_\Omega w(r, r')\phi(r', t - D(r, r'))\, dr', \quad (r, t) \in \Omega \times [0, T], \tag{3.28}$$

the inverse problem is reformulated into the equation

$$\psi^{(\xi)}(r, t) = (W\phi^{(\xi)})(r, t), \quad (r, t) \in \Omega \times [0, T], \tag{3.29}$$

with $\xi = 1, 2, ..., N$, where the kernel $w(r, r')$, $r, r' \in \Omega$ of the linear operator W is unknown.

The equation (3.29) can be written as

$$\psi = W\phi, \tag{3.30}$$

with $\phi = (\phi^{(1)}, \phi^{(2)}, ..., \phi^{(N)})^T$ and $\psi = (\psi^{(1)}, \psi^{(2)}, ..., \psi^{(N)})^T$, where we search for the unknown operator W. Here, for simplicity we will restrict the further derivation to the case $N = 1$, where only one dynamical field is considered.

An alternative is to rewrite (3.29) as

$$\psi_r(t) = \int_\Omega \phi(r', t - D(r, r'))w_r(r')\, dr', \quad t \in [0, T] \tag{3.31}$$

for every fixed $r \in \Omega$ with

$$\psi_r(t) := \psi(r, t), \quad t \in [0, T] \tag{3.32}$$

and

$$w_r := w(r, r'), \quad r, r' \in \Omega. \tag{3.33}$$

Equation (3.31) is a *family of integral equations* for the unknown kernel $w(r, r')$. For each function $w_r = w(r, \cdot)$ it formulates a different integral equation which has both a different integral kernel and a different left-hand side. It can be rewritten using the integral operator

$$(V_r g)(t) := \int_\Omega K_r(t, r')g(r')\, dr', \quad t \in [0, T], \tag{3.34}$$

with kernel

$$K_r(t, r') := \phi(r', t - D(r, r')), \quad (t, r') \in [0, T] \times \Omega, \tag{3.35}$$

for $r \in \Omega$. For $N > 1$, this kernel is a vector of functions $\phi^{(\xi)}(r,', t - D(r, r'))$, $\xi = 1, ..., N$.

Now, our inverse problem equation (3.31) is given by

$$\psi_r = V_r w_r, \tag{3.36}$$

for $r \in \Omega$. For each $r \in \Omega$, equation (3.36) is a *Fredholm integral equation of the first kind* with continuous kernel ϕ. The operator V_r is a *compact* operator on the spaces $C(\Omega)$, $L^1(\Omega)$ or $L^2(\Omega)$ into $BC([0, T])$. It is well-known (c.f. [37, 59, 62, 76]) that this equation is *ill-posed*, i.e. it does not need to have unique solutions and if it has a solution in general this solution does not depend continuously on the right-hand side.

Ill-posed equations need some *regularization method* (c.f. [48]) in order to obtain a stable solution. A standard approach to regularization is built on the *singular*

system (c.f. [62]) of the operator under consideration. In summary, for a compact linear operator $A : X \to Y$ between Hilbert spaces X,Y, and its adjoint A^* the *singular values* μ_n of the operator A are the non-negative square roots of the eigenvalues of the self-adjoint compact operator $A^*A : X \to X$. This leads to representation of the operator as a multiplication operator of two orthonormal systems $g_n : n \in \mathbb{N}$ in X and $y_n : n \in \mathbb{N}$ in Y. Hence, this correspond to a spectral representation of the operator A in the form

$$Ag = \sum_{n=1}^{\infty} \mu_n \langle g, g_n \rangle y_n, \tag{3.37}$$

for $g \in X$.

For the orthonormal systems g_n and y_n we obtain:

$$Ag_n = \mu_n y_n, \quad A^* y_n = \mu_n g_n. \tag{3.38}$$

Here, in case A is injective, the inverse of A is given by

$$A^{-1} y = \sum_{n=1}^{\infty} \frac{1}{\mu_n} \langle y, y_n \rangle g_n, \tag{3.39}$$

or, if A is not injective, the inverse A^{-1} in equation (3.39) projects onto the orthogonal space

$$N(A)^{\perp} = \{g | \langle g, g^* \rangle = 0, \ \forall g^* \in N(A)\}.$$

Because of the compactness of the operators A, the singular values are a sequence mostly accumulating at zero. So, the behavior of $\left|\frac{1}{\mu_n}\right| \to \infty, n \to \infty$ enlarges small errors causing the instability of applying the inversion. The practical behaviour of the sequence of singular values μ_n provides important insight into the *nature of the instability* of the application at hand the problem is strongly ill-posed for strong smoothness of the function ϕ.

3.4.2 Regularization for Kernel Reconstruction

To deal with this instability, we apply the regularization techniques to minimize the value of the factor $\frac{1}{\mu_n}$ for large n.

We replace it by another factor q_n which is bounded for $n \in \mathbb{N}$, and modify the inverse operator by

$$R_\alpha y = \sum_{n=1}^{\infty} q_n^{(\alpha)} \langle y, y_n \rangle g_n, \tag{3.40}$$

where $\alpha > 0$ is known as *regularization parameter* and the specific choice of damping factors

$$q_n^{(\alpha)} := \frac{\mu_n}{\alpha + \mu_n^2}, \quad n \in \mathbb{N}, \tag{3.41}$$

leads to the famous *Tikhonov regularization* as discussed in Section 2.3.2 (see also [48, 62, 37, 76]).

THEOREM 3.4.1. *Let $u(r,t)$ for $r \in \Omega$ and $t \in [0,T]$ be some neural activity function, which obeys the neural field equation (1.6) with true kernel w^* and some initial conditions $u(r,t) = u_0(r,t)$, for $(r,t) \in \Omega \times [-c_T, 0]$. Then, the application of the Tikhonov regularization (3.40) to the integral equation (3.36) leads to the reconstruction $w_\alpha(r,r')$ of Pw^*, where P is applied to the second argument of $w(r,r')$ as the projection of w_r^* onto $N(V_r)^\perp$, i.e., it is defined as*

$$(Pw)(r,\cdot) = P_r w_r, \quad r \in \Omega. \tag{3.42}$$

Proof. Here, we base our reconstruction on a well-known result (c.f. [76], Theorem 3.1.8) which states that Tikhonov regularization is a *regularization scheme* in the sense of Definition 3.1.4 of [76], and the discussion in Section 2.3.2 i.e. that if

$$f = A(\varphi^*) \in R(A),$$

then $R_\alpha f \to \varphi^*$ for $\alpha \to 0$.

If A is not injective, and by splitting the space into

$$N(A) \text{ and } N(A)^\perp = \overline{A^*(X)},$$

we see by $w_r = Pw_r + (I - P)w_r$ and the definition of A^* that the convergence of $R_\alpha f$ is towards the projection $P\varphi^*$ of φ^* onto $N(A)^\perp$. In our case, the reconstruction calculates an approximation to Pw_r^*. This completes the proof. □

Usually, Tikhonov regularization is carried out by applying an efficient solver[1] to the equation

$$(\alpha I + A^*A)g = A^*y, \tag{3.43}$$

which is equivalent to the spectral version of Equation (3.40). Equation (3.43) is used for our numerical examples of Section 6.1.

[1] For large-scale problems a conjugate-gradient method is used for solving the equation sequentially. For smaller problems matrix inversion by Gauss' method is sufficient.

3.5 Sensitivity Analysis

An important basic question is the influence of noise on the reconstruction. Here, we carry out a *sensitivity analysis*, i.e. we calculate the Fréchet derivative of the reconstructed kernel with respect to the input function u. Differentiability is obtained in a straightforward manner following Section 2.6 of [76], and the discussion in Section 2.1.5.

We start with Equation (3.36), where the operator V_r and the right-hand side ψ_r depend on the input function u. The reconstruction of w is carried out by the regularized version of

$$w_r = (V_r)^{-1}\psi_r, \tag{3.44}$$

which in the case of Tikhonov regularization in Equation (3.43) is

$$\begin{aligned} w_{r,\alpha} &= R_\alpha \psi_r \\ &= (\alpha I + V_r^* V_r)^{-1} V_r^* \psi_r. \end{aligned} \tag{3.45}$$

We differentiate with respect to u on both sides and employ the chain rule and equation (2.6.21) of [76], to derive the *unregularized* form

$$\frac{\partial w_r}{\partial u} = -(V_r)^{-1}\frac{\partial V_r}{\partial u}(V_r)^{-1}\psi_r + (V_r)^{-1}\frac{\partial \psi_r}{\partial u}, \tag{3.46}$$

and the derivative of the *regularized* reconstruction

$$\begin{aligned} \frac{\partial w_{r,\alpha}}{\partial u} &= -Q\frac{\partial (V_r^* V_r)}{\partial u} Q V_r^* \psi_r + Q\frac{\partial V_r^*}{\partial u}\psi_r + Q V_r^* \frac{\partial \psi_r}{\partial u} \\ &= -Q\frac{\partial V_r^*}{\partial u} V_r Q V_r^* \psi_r - Q V_r^* \frac{\partial V_r}{\partial u} Q V_r^* \psi_r + Q\frac{\partial V_r^*}{\partial u}\psi_r + Q V_r^* \frac{\partial \psi_r}{\partial u}, \end{aligned} \tag{3.47}$$

where we use the notation

$$Q_r := (\alpha I + V_r^* V_r)^{-1}. \tag{3.48}$$

The derivatives of V_r and ψ_r with respect to u are calculated as follows, where we restrict our presentation to the case where we are given *one* trajectory only.

The operator V_r in its dependence on u is given by

$$(V_r[u]g)(t) = \int_\Omega f[u(r', t - D(r, r'))]g(r')\, dr', \quad t \in [0, T], \tag{3.49}$$

leading to the Fréchet derivative

$$(\frac{\partial V_r[u]}{\partial u}(\delta u)g)(t) = \\ \int_\Omega f'[u(r', t - D(r, r'))]\delta u(r', t - D(r, r'))g(r')\, dr', \quad t \in [0, T], \tag{3.50}$$

where f' denotes the derivative of the function $f(s)$ with respect to its real argument $s \in \mathbb{R}$. We need to assume that f is differentiable and that the derivative is continuous and bounded. The derivative of the adjoint V_r^* with respect to the L^2 scalar products on Ω and $[0, T]$, which is

$$(V_r^*[u]\eta)(r') = \int_0^T f[u(r', t - D(r, r'))]\eta(t)\, dt, \quad r' \in \Omega, \tag{3.51}$$

is given by

$$(\frac{\partial V_r^*[u]}{\partial u}(\delta u)\eta)(r') = \int_0^T f'[u(r', t - D(r, r'))]\delta u(r', t - D(r, r'))\eta(t)\, dt, \tag{3.52}$$

for $r' \in \Omega$. We note that V_r^* is an operator into $L^1(\Omega)$, which depends bounded continuously on $r \in \Omega$. The Fréchet derivative of the function ψ_r given by (3.27) is readily seen to be given by

$$\frac{\partial \psi_r}{\partial u}(\delta u) = \tau \frac{\partial \delta u}{\partial t}(r, t) + \delta u(r, t), \tag{3.53}$$

for $(r, t) \in \Omega \times [0, T]$.

We summarize the results in the following Theorem.

THEOREM 3.5.1. *Assume that the activation function f is continuously differentiable with derivative f' bounded on $\mathbb{R}$. Then, for each fixed $\alpha > 0$ the regularized reconstruction of the kernel w from input signals u within the framework of the delay neural field equation is continuously Fréchet differentiable with respect to u considered as mapping from $BC(\Omega) \times C^1([0, T])$ into $BC(\Omega) \times L^1(\Omega)$. This implies continuity of the mapping of u onto w. The total derivative of w_r with respect to u is obtained by the combination of (3.48) with (3.50), (3.52) and (3.53).*

Proof. Differentiability follows from the differentiability of all the operators in (3.46) following equations (3.48) to (3.53) of the above arguments. □

3.6 Summary and Further Discussion

The purpose of this chapter is to develop an *integral equation approach* for kernel reconstructions in delayed neural field equations and to study its practical feasibility. We study the activity and evolution of a delayed neural field of Amari-type to develop an effective approach of reconstructing the neural connectivity. As a preparation for the inverse problem, this work includes an explicit study of the solvability of the direct problem of the delayed neural field equation (1.6) and achieve to prove the existence and stability of its solution. We provide an easily accessible functional analytic approach based on an integral equation and Banach's fixed point theorem.

As our main result in this chapter, we apply inverse problems techniques to reconstructing the neural kernel assuming that some measurements of the activity $u(r,t)$ are given. We start by formulating a *family of integral equations* of the *first kind*. Since kernel reconstruction is ill-posed, we need regularization to obtain stable solutions. As stabilization method we employ the *Tikhonov regularization*. A *sensitivity analysis* is carried out, showing that the mapping of the input u to the regularized kernel reconstruction is Fréchet differentiable. The derivative is explicitly calculated based on the integral equation approach.

Furthermore, in this work, we assume that the delay function D to be given, as it would be the case when the delay is approximately proportional to the distance of the nodes under consideration. If D is unknown, w is known and u is measured, we can solve in Equation (1.6) for $u(r', t - D(r, r'))$ for all r, r' and t. This is still ill-posed, since it involves an integral equation of the first kind, but then the determination of D is reduced to the reconstruction of D from the knowledge of $u(r', t-D(r, r'))$, which strongly depends on the form of the signal u and conditions we impose on D. If neither the delay D nor w would be given, the kernel K_r of operator V_r, in Equation (3.34), would be unknown and part of the reconstruction, leading to many open questions of feasibility and observability. In general, the reconstruction of both the kernel w and the delay D is an important non-linear, far reaching and challenging problem of future research.

In summary, we have developed a *stable* and *efficient* approach for the reconstruction of the *connectivity* in neural systems based on *delay neural field equations*. We expect the approach to be extensible to a wide range of field models with delay, and in particular to be highly useful for analysis of experimental data in the domain of computational neuroscience. These methods allow the reconstruction of the underlying "synaptic footprint" of connectivity from available neural activity measurements, thus providing a basis for simulation and prediction of real phenomena in the neuroscience.

It is of high interest to study the applicability of the approach of kernel reconstruction in the framework of other dynamical systems. It is non-trivial that the inversion approach to reconstruct the kernel is an effective tool which is applicable not only to the neural field problem but also many other different dynamical systems. It should be applicable if its dynamics is using a kernel to model connectivity.

Chapter 4

State Estimation for Neural Field Equations

As we mentioned before, it is well known that the community of modeling the activity of neural tissue has grown strongly since the early work of Wilson, Cowan and Amari in the 1970s [102, 103, 54]. This growth leads to improvement to promising tools in theory and applications of mathematical neuroscience. For details, we refer to different books that discuss the history of neural imaging, neural modeling and their applications [70, 4, 58, 26]. More discussion of further studies in applying different techniques for neural fields can be found in recent books such as [76, 60]. The challenge in this field of study is related to the complex structure of both neural networks and neurons themselves, where simplifying the system is not easy at all. We note that the neural tissue presents different properties in space and time, as well as, including a large number of neurons in human nerve system counted as 100 billion all connected to each other in complicated way which makes studying its full complexity a nearly impossible task, for more details of complexity of neurons, neural fields see [70, 58, 4, 26, 17, 14].

Let us describe the challenges, which researchers face on several layers when dealing with dynamical models in neuroscience, in more detail. We note that, even though neural models have gained significantly in complexity and realism (see e.g. recent work [1, 3, 77, 69, 36]), they do not fully reflect the complexity of nature comparing to psychological, physiological and experimental results. Usually, models employ some level of abstraction, such as field models which replace the micro-structure of neurons by their average activity patterns and connectivity [26, 95]. We note that our choice of Amari equation has an advantage of being simple while capturing neural dynamics and meeting the results of the EEG. Although it is relatively simple, there are some challenges facing researchers in dealing with them. Here, we need to carry out two basic tasks:

- *first*, to move from basic pattern studies to realistic simulations we need to determine the *connectivity structure*, which is called the *inverse kernel problem* of neuroscience [85, 12, 82].

- *second*, to carry out simulations we need to determine *initial states* from measurements. This is known as the *data assimilation problem.* Applying data assimilation techniques in neuroscience is mentioned in some recent work, see for example [77, 74, 73].

After studying the kernel reconstruction of the delayed neural field in Chapter 3, we are now ready to provide a study of the state estimation of the neural field. This is a further step towards our plan described in Section 1.2 to develop an *iterative* algorithm to solve the coupled problem of inversion and data assimilation for neural field equation by iteratively carrying out a data assimilation first, based on a (potentially rough) approximation to the dynamical system, which is then used to calculate better structural information by solving the inverse problem. This process can be repeated and, under suitable conditions, converges to the full solution of the coupled problem, as will be shown in Chapter 5.

Firstly, despite the level of simplicity we employ for our current simulation, the task to determine (1) the structural information and (2) the initial states are indispensable. The arguments of this chapter apply to both the non-delay as well as the delay case, i.e. it applies to the Amari and Cowan-Wilson neural field model (1.4), which is obtained by considering neurons which are continuously distributed over the space Ω in $\mathbb{R}^2$ or $\mathbb{R}^3$

$$\tau \frac{\partial u}{\partial t}(r,t) = -u(r,t) + \int_\Omega w(r,r') f(u(r',t))\, dr', \quad r \in \Omega. \tag{4.1}$$

From Section 1.1.2 we recall that $u(r,t)$ interprets a neural field representing the activity of the population of neurons at position r and time t.

Taking into account a delay of the propagation from the source of some excitation to its evaluation point, we studied the inverse problem for the more general form of the neural field equation in Chapter 3, where the equation takes the form

$$\tau \frac{\partial u}{\partial t}(r,t) = -u(r,t) + \int_\Omega w(r,r') f(u(r',t - D(r,r')))\, dr'. \tag{4.2}$$

We assume that $D(r,r') \simeq \tilde{D}(r,r')/v$, where $\tilde{D}(r,r')$ is the length of the fiber between r and r', and v is the finite propagation speed of signals. For simplify, in some of our examples we will work with $\tilde{D}(r,r') = \|r - r'\|$ and $v = 1$. In general,

the delay D is not constant and we assume that D is continuous. The existence of solutions to the neural field equation (4.2) has been investigated in Chapter 3 and is published in [2, 85, 40].

In Chapter 3, for kernel reconstruction problem of 4.2 we have assumed that a full state u is known to carry out an inversion process. The same assumption has been made in the literature cited above. In reality, to gain knowledge of the field u one needs the estimation of the neural state, given some measurements or experimental results. The estimation techniques within a dynamical framework are usually known under the name *data assimilation*.

In this chapter, we employ this technique for neural field theory for Amary equation 4.1. We will first introduce the concept of state estimation by data assimilation and its applications in Section 4.1. In Section 4.2, we describe a variational approach to data assimilation in a neural field setup, which is basically a three-dimensional variational assimilation 3D-VAR equipped with a Gaussian covariance matrix B.

4.1 Introduction to Data Assimilation.

Data assimilation is a mathematical and statistical discipline combining theory of modeling with observations. The goals of data assimilation might differ between determining either (1) the best state estimate of a system, (2) initial conditions for a model, (3) interpolate observation data or (4) train model parameters based on observed data. These different goals lead to a variety of methods. However, the approach is usually based on an evolution of the dynamic state from earlier points in time to the time of some measurement, known as first guess. Now, data assimilation is calculating a best estimate based on the observation and the first guess, in this sense merging observations and the state propagated by the dynamical model [16, 76].

Data Assimilation is distinguished from other forms of machine learnign, image analysis and statistical methods in that it utilizes a dynamical model of the system being analyzed. It has a long history in atmospheric and oceanic sciences and in hydrology since 1950s when it gained attention due to developing of computers. However, the first attempt was in 1922 by Richardson according to [13]. Many books consider mathematical aspect of data assimilation problem such as [66, 42], while others concern on shaping the knowledge of data assimilation as a unique discipline [31, 56]. There are also plenty of research to bridge between the mathematical knowledge and data assimilation theory such as [63, 89, 13, 69, 3, 20, 39]. More researchers study data assimilation in neural dynamics framework [71, 77, 60].

Historically, data assimilation was strongly related to geoscience and meteorology and its famous application in the framework of prediction [31, 56, 1, 42]. Then, since the 90's, it proved to be applicable in a variety of applications of different scientific, medical, technological and industrial fields. By using data assimilation algorithms, simulations can be synchronized with reality and dynamical forecasting becomes possible.

Aiming to estimate the state of a dynamical system, data assimilation uses a combination of observations and a numerical model. This combination is an efficient tool to deal with

- missing information and
- noisy observations.

Our goal is to employ data assimilation as an effective approach to achieve the reconstruction of the activity fields u given particular measurements y of this field modeled by some linear *observation operator* $y = Hu$. If we have any measurements at any time, data assimilation can improve our knowledge of the connectivity, and thus contribute to reconstructing the model and get information or improvement for all times.

We assume that at different time steps $t_1, t_2, ..., t_N$, there are given measurements

$$y_1, y_2, ..., y_N, \tag{4.3}$$

in some measurement space Y with a measurement operator H.

The state can be written as

$$u_t := M(t, 0)u_0, \ t \geq 0,$$

where M is the model operator. The goal is to estimate the state of the dynamical system

$$u_1^{(a)}, u_2^{(a)}, ..., u_N^{(a)}, \tag{4.4}$$

at times $t_1, t_2, ..., t_N$, where the letter a indicates the *analysis* or the results of the estimation procedure. Now to build the estimation cycle, we need a *background* to carry out the state estimation. Here, we calculate the background at the times $t_1, ..., t_n$ by

$$u_0^{(b)} = u_0, \ u_1^{(b)} = M_{t_1}u_0, \ ..., \ u_n^{(b)} = M_{t_n}u_0, \tag{4.5}$$

where M_{t_n} is the model propagation from time $t = 0$ to time $t = t_n$ and the letter b indicates the *background*. More details will be provided later.

4.1.1 Errors and Uncertainty in Data assimilation

Assuming that the background field $u^{(b)}$ is known, we already have a background error which is the difference between the background and the true states:

$$\epsilon^{(b)} = u^{(b)} - u^{(t)}$$

, where t indicates the *truth*. It is really difficult to control this error after applying the analysis many times. That leads to the need of calculate statistics of $\epsilon^{(b)}$.

When applying data assimilation processes, we expect three different kinds of errors:

- background error: this error has an average $\overline{\epsilon^{(b)}}$ and coveriances

$$B = \overline{(\epsilon^{(b)} - \overline{\epsilon^{(b)}})(\epsilon^{(b)} - \overline{\epsilon^{(b)}})^T}$$

  , where T indecate the transpose,

- observation error: $\epsilon^{(o)} = y - Hu^{(t)}$, with average $\overline{\epsilon^{(o)}}$ and covariances

$$R = \overline{(\epsilon^{(o)} - \overline{\epsilon^{(o)}})(\epsilon^{(o)} - \overline{\epsilon^{(o)}})^T}$$

  ,

- analysis error: $\epsilon^{(a)} = u^{(a)} - u^{(t)}$ with average $\overline{\epsilon^{(a)}}$.

4.1.2 Methods of Data Assimilation

There are different mathematical methods used to carry out data assimilation. A first class are so-called *variational methods*, in particular

(1) 3D Variational Data Assimilation,

(2) 4D Variational Data Assimilation,

which originally came from control theory. These methods search a *best fit* to both the observations and the first guess (background) states.

Three-dimensional variational assimilation (3D-VAR) carries out some minimization at a given point in time t_k, calculating the so-called *analysis*. Then, it propagates this analysis to the next point in time where measurements are available. The state is called *first guess* or *background* and provides the basis for the next minimization step.

Four-dimensional variational assimilation carries out minimization over a temporal interval $[t_k, t_k + T]$, taking into account all observations between points t_k and $t_k + T$ in time.

Secondly, there is *ensemble data assimilation* based on stochastic ideas, in particular

(3) Ensemble Kalman Filters and Smoothers

(4) Particle Filters.

Their origin can be found in estimation theory. They are based on an ensemble of states estimating the underlying uncertainty distribution, and they calculate an ensemble of states to estimate the analysis uncertainty distribution. For more details about data assimilation, its methods and applications, we refer to [31, 16, 56, 41, 66, 76, 13, 42, 10, 20].

Here, we will focus on three-dimensional variational data assimilation (3D-VAR) applied to the neural field equations. Let we start with an overview of 3D-Var in next section.

4.1.3 Overview of 3D-Var

The analysis in 3D-Var is considering the approximation solution to a minimization problem defined by a cost function J as several evaluations of it,

$$J(u) = (u - u^{(b)})^T B^{-1}(u - u^{(b)}) + (y - H[u])^T R^{-1}(y - H[u]), \tag{4.6}$$

and of its gradient

$$\nabla J(u) = 2B^{-1}(u - u^{(b)}) - 2H^T R^{-1}(y - H[u]). \tag{4.7}$$

The minimization approach is an iteration to find the analysis state u that minimizes $J(u)$. This solution uses a priori data, represented as the background $u^{(b)}$ and observations y, to produce a pasteriori maximum likelihood estimate of the truth. The method includes the estimates of the background B and observation R error covariance matrices to weigh the data in the analysis. To reach the goal of optimization, this method uses a stopping procedure by limiting either the number of iterations or the norm of the gradient $||\nabla J(u)||$. For the latter procedure, a specific level of decreasing of the gradient is required for the stopping point where the analysis is expected to be closer to the optimum. We note that, in addition to the simplicity of the method, efficient calculations of the gradient of the cost function can be obtained by using adjoint operations. In fact, that gives the

method one of its advantages as only the operators and adjoints of their tangent linear are needed despite how the observation operators are complex. There is always one and only one analysis if J is strictly convex. For more details see [16, 11]

4.2 Data Assimilation of Neural Fields

The goal of this section is to develop a data assimilation approach for the estimation of states in neural field theory following three-dimensional variational data assimilation as described in [76]. Data assimilation algorithms already have a long history in geophysical applications and become a promising technique in various of dynamic modeling, compare [42, 44, 56].

Here, for the sake of simplicity we will describe the data assimilation problem in the discretized space, i.e. we assume that we employ a discretization of the domain Ω, i.e. we define

$$v_j := v(r_j), \quad j = 1, ..., n \tag{4.8}$$

where v_j's are observational points, $r_j \in \Omega$ are discretization or *collocation points*. Our neural activity field $u(\cdot, t)$ in $L^2(\Omega)$ is thus approximated by its values $u_j(t) := u(r_j, t)$, such that $u(t) \in \mathbb{R}^n$ for $t \in [0, T]$.

We assume that we have *measurements* of the neural field which are given by electrodes. Usually electrodes collect the potential of the underlying tissue such that the measurement process corresponds to an integration of the original fields. This is given by the measurement or *observation operator*

$$(H_\ell u(\cdot, t)) := \int_\Omega v_\ell(r') u(r', t)\, dr' \tag{4.9}$$

for $\ell = 1, ..., m$. Here, the functions v_ℓ for $\ell = 1, ..., m$ represent the electrode geometry and functionality in the sense that it measures a weighted average of the activity potential u over some area $U \subset \Omega$ and neglects the activity in the complement U^c of U in Ω. Choosing v_ℓ as a delta function $v_\ell(r') := \delta(r' - r_{j_\ell})$ located at some node r_{j_ℓ} for $\ell = 1, ..., m$, the measurements correspond to individual activity measurements at a selection of m nodes of the discretized tissue areas. In discretized form, equation (4.9) takes the form of a sum

$$H_\ell u(\cdot, t) = v_\ell \cdot u(\cdot, t) = v_\ell^T\, u(\cdot, t) \tag{4.10}$$

to be read either as a scalar product of the vectors $v_\xi \in \mathbb{R}^n$ and $u(\cdot, t) \in \mathbb{R}^n$ or as a matrix-vector product of the row vector v_ℓ^T with the column vector $u(\cdot, t)$. We

collect all components H_ℓ for $\ell = 1, ..., n$ into one matrix operation which we call H.

Data assimilation searches for an optimal state $u^{(a)}$ with respect to both the measurements

$$y = Hu(\cdot, t_k) + \epsilon \quad \in \mathbb{R}^m \tag{4.11}$$

at some given point t_k in time with some vector of errors $\epsilon \in \mathbb{R}^m$, and some á-priori state $u^{(b)}(\cdot, t_k)$, which is usually calculated from earlier estimates. If measurements do not pick up the complete state, we need to exploit dependencies between different state variables. For example, if we only measure $u_1(t_k)$ at time t_k, we have no further information about the values of u at other nodes r_j, $j = 2, ..., n$. But if we have statistical information about the correlation of u_2 to u_1, then information about u_1 will also tell us much about u_2 and potentially the other variables.

Data assimilation usually employs the covariance matrix $B \in \mathbb{R}^{n\times n}$ to incorporate the current stochastic relationship between different system variables. The minimization of the cost function (4.6) leads to the estimate $u^{(a)}$ of the neural state given measurements according to (4.9) given by

$$u^{(a)} := u^{(b)} + BH^T(R + HBH^T)^{-1}(y - Hu^{(b)}), \tag{4.12}$$

where $R \in \mathbb{R}^{m\times m}$ denotes the covariance matrix of the data error $\epsilon \in \mathbb{R}^m$ and B is the covariance of the errors of the model state u, compare also equation (5.2.14) of Chapter 5 of [76] (with $\alpha = 1$). Here, we choose R and B as prescribed matrices in $\mathbb{R}^m$ and $\mathbb{R}^n$, respectively. For linear observation operators H the solution $u^{(a)}$ is the unique minimizer of the functional (4.6).

Here, we choose our matrix B to be Gaussian with respect to the distance of coordinates $r_i, r_j \in \Omega$, i.e. we have

$$B_{i,j} := ce^{-\sigma|r_i - r_j|^2}, \quad r_i, r_j \in \Omega, \tag{4.13}$$

where $\sigma > 0$ is a decay parameter for the statistical dependencies and c is a norming constant. The covariance matrix R is considered to be diagonal, which corresponds to statistically independent measurements with the different electrodes H_ℓ for $\ell = 1, ..., m$.

In the case where we do not know any reasonable system dynamics yet, we can employ the zero dynamics for the first propagation, i.e. we work with $u^{(b)} \equiv 0$ on Ω for $t = 0$ to $t = t_n$ and carry out estimates based on the zero background field. For the second iteration, we have now estimated some kernel and u^b will be non-zero for any further iteration step. If some neural kernel is given, we can use

the neural field equation to calculate a first guess $u^{(b)}(\cdot, t_k)$ at time t_k from earlier estimates $u^{(a)}(\cdot, t_{k-1})$. We will denote the model propagation of the state at time t_{k-1} to time t_k by $M_{k-1,k}$.

Definition 4.2.1 (3D-VAR for Neural State Estimation).
The three-dimensional variational method for neural state estimation from electrode measurements employs measurements

$$y_k = Hu^{(true)}(\cdot, t_k) + \epsilon \tag{4.14}$$

with error $\epsilon \in \mathbb{R}^m$ *and a* first guess *or* background

$$u^{(b)}(\cdot, t_k) := M_{k-1,k} u^{(a)}(\cdot, t_{k-1}) \tag{4.15}$$

to calculate an estimate $u^{(a)}(\cdot, t_k)$ *of the neural activity at time* t_k *according to the equation (4.12).*

$$u^{(a)} := u^{(b)} + BH^T(R + HBH^T)^{-1}(y - Hu^{(b)})$$

For 3D-VAR, the background state covariance matrix B can be calculated from statistical evaluations of neural fields, or it can be chosen by generic arguments. Here, we will start with a Gaussian matrix as given by (4.13).

In the case of high-dimensional state spaces X and a large number of measurements, the solution of the equation

$$(R + HBH^T)z = y - Hu^{(b)}, \tag{4.16}$$

which is needed to evaluate equation (4.12), is usually carried out based on the *conjugate gradient* method. Then, the analysis increment $u^{(a)} - u^{(b)}$ is calculated by $BH^T z$ in a subsequent step. When the dimension n of the state space X allows explicit inversion of the equation (4.12), we can employ standard implementations of Gauss' method.

We can see an example of this idea how we expect its convergence in Figure 4.1 where the original state u is shown in red, and the measurements appear in blue. Using the observation operator $y = Hu$ and covariance matrix B, the estimation by 3D-Var based on the reconstruction is shown in magenta. Here, we have used an observation operator H which selects states at each 5th point on a unit circle. We have chosen a B matrix as defined in (4.13), an observation error covariance matrix $R = rI$ to be the identity matrix I multiplied by a scalar r. Reconstructions are

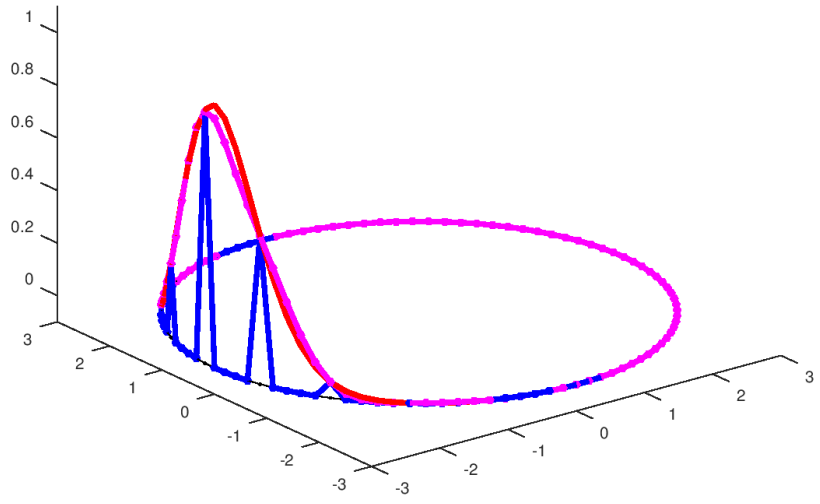


Figure 4.1: In this figure, the original state in red looks close to the updated one in magenta where reconstructed kernel appears in blue

then obtained on all points by (4.16). The code of this example is presented in Section A.2.

Complete numerical examples of applying 3D-Var to estimate the neural state are described in Section 6.3. After applying the estimation of the state, we now ready to the next step. We put together the both methods of kernel reconstruction and state estimation in one iterative approach and study its stability in next Chapter 5.

4.3 Summary

Following the application of *inverse problem* technique to reconstruct the kernel of NFE, this part is concerned on estimate the neural state using *data assimilation* or precisely 3D-VAR method. Although the effectiveness of kernel reconstruction approach has been shown before in different studies and discussed in details for delay case in Chapter 3, the argument was always built on assumption of having knowledge of the excitation field. In reality, we have measurements of the field and we need to use one of data assimilation technique to fill this gap.

In this chapter, we provide a study of the application of 3D-VAR method in neural field equation. That could improve by updating the kernel using inverse problem. This is leading to the next step of introducing the iteration of inversion and data assimilation in next chapter.

Finally, in next chapter, we show the opportunity to build a complete scheme based on applying both procedures of *data assimilation* and *inverse problems* techniques in turns.

Chapter 5

Iteration of Data Assimilation and Inversion

Let us briefly review the developments worked out in Chapter 3 on inverse neural delay and Chapter 4 on data assimilation for neural fields.

- We have developed an *inverse problems* technique to reconstruct the connectivity kernel once the full neural activity $u(x,t)$ is known for all x in space and for time t in some interval $[0,T]$ of times.
- We have developed a *data assimilation* technique to estimate the full neural activity $u(x,t)$ from measurements y_k at times t_k, $k = 1,2,3,...$ for $t_k \in [0,T]$.

For the kernel reconstruction, the full excitation field u is the critical input. It is determining the kernel w as a result of the inversion or reconstruction process. For the estimation step by data assimilation, the *first guess* or *background* $u^{(b)}$ is very important, since the quality of the analysis $u^{(a)}$ is strongly depending on the *prior* knowledge of the dynamics and, thus, on the kernel w which is our critical input to the estimation process.

We summarize the dependencies as follows:

- The quality of the kernel reconstruction is strongly depending on the quality of the input u for its reconstruction.
- The quality of the state estimation of u is strongly depending on the quality of the input, which is both the measurements y_k and the background fields $u^{(b)}(x,t)$. The background fields $u^{(b)}$ depend on the knowledge of the kernel w, since they are simulated based on some initial field $u^{(b)}(x,0)$ and the neural dynamics (4.1) or (4.2), which includes the kernel w.

A simple idea to resolve the mutual dependencies of inversion and state estimation is *iteration*.

(S1) Based on some initial knowledge of our kernel we simulate background fields $u^{(b)}(x,t)$ for $x \in \Omega$ and $t \in [0,T]$. We call this the *transport map*.

(S2) Then, we employ our measurements y_k at t_k and the background fields to estimate the full neural fields $u^{(a)}(x,t)$ for $x \in \Omega$ and $t \in [0,T]$ using the techniques of Chapter 4. We call this the *estimation step*.

(S3) We now employ the full neural excitation field $u^{(a)}(x,t)$ to carry out an *inversion step* following Chapter 3 and reconstruct the neural connectivity kernel w.

(S4) With the kernel w we have gained better knowledge of the neural dynamics, such that we can go back to step 1 and restart the whole process of transport, estimation and inversion. We speak about *iterating estimation and inversion*.

As an important part of the full development of our methods, we need to investigate the convergence and the stability of our iterative approach of taking inversion and data assimilation in turns to improve kernel reconstruction and forecasting. Although the basic idea of an iterative approach of data assimilation and inversion for neural fields is easily explained, the convergence and stability of the iterative method turns out to be a challenging task due to the ill-posedness of the kernel reconstruction and the strong role of the regularisation scheme, which needs to be balanced with the errors of the state estimation during the iteration process.

There is a well developed theory and convergence analysis for the usefulness and effectiveness of both the *inverse problems technique* and each *data assimilation* step individually. This work considers the iterative procedure of *kernel reconstruction* and *state estimation*, which calculates an approximation both for the neural field u and for the kernel w. We note that the use of the same observations each time step in (S2) introduces correlations between the observation and background errors after the first iteration. This could be treated by some regularization method.

The question at this point is about the convergence of the iterative procedure to the best estimate for both u and w. In our first Section 5.1 we develop the combination of inversion and data assimilation into one *iterative* approach. This iteration of data assimilation and inversion is described and analyzed in Section 5.2 by decomposing it into three operators and studying their properties. We formulate the iterative approach as a sequential application of these three operators

in Section 5.3. Then we work out the convergence proof, for our iterative method in Section 5.4. Numerical examples will be given in the subsequent Chapter 6.

For our mathematical derivation, we will limit our presentation to the case of non-delay equations to keep the approach readible. In principle, as we have transfered the existence proof to the delay case, we believe that this should be possible for the arguments presented as well. But it is beyond the possiblities of this thesis and has to be left to future work.

5.1 On The Combination of State Estimation and Inversion

In this section we introduce an iterative method to solve the combined inverse neural field problem when measurements are given based on some basic electrode forward model. The main idea is schematically sketched in Figure 5.1. We treat the case where our measurements provide knowledge about the underlying neural activation function u, but these states are not directly accessible. Here, different types of measurements are possible, modeled by the *observation operator* H introduced in (4.9). Different approaches to state estimation can be seamlessly integrated into our iteration procedure.

The main idea of the iterative scheme is summarized through the following points. In each step ℓ for $\ell = 1, 2, 3, ...$:

(S1) We first take some initial kernel $w^{(0)}$ and calculate initial fields $u^{(b,1)}(x,t)$ for $x \in \Omega$ and $t \in \{t_1, t_2, ..., t_K\}$ based on some initial field $u(x, t_0)$, $x \in \Omega$ solving equation (4.1) or (4.2). This is called the *transport step* and we use the notation

$$u^{(b,\ell)} = T(w^{(\ell-1)}) \tag{5.1}$$

with the *transport map* T. Note that if no knowledge about the kernel is given, we can take $w^{(0)} \equiv 0$ and then also $u^{(b,\ell)}(x,t) \equiv 0$ for $x \in \Omega$ and $t \in \{t_1, ..., t_N\}$.

(S2) We now calculate an approximation $u^{(a,\ell)}(x,t)$ for $x \in \Omega$ and $t \in \{t_1, t_2, ..., t_K\}$ to the full neural activity field from measurements $y_1, ..., y_K$ at times $t_1, ..., t_K$. The estimation is carried out based on the background $u^{(b,\ell)}$ calculated in step (S1). We use the notation

$$\begin{aligned} u^{(a,\ell)} &= E(u^{(b,\ell)}) \\ &= E(T(w^{(\ell-1)})) \end{aligned} \tag{5.2}$$

with the *estimation operator* E.

(S3) As third step, we calculate a reconstruction $w^{(\ell)}$ of the neural connectivity kernel from the estimation $u^{(a,\ell)}$ of the full time-dependent neural field. We write this in the form

$$w^{(\ell)} = K(u^{(a,\ell)}) \tag{5.3}$$

with the *kernel reconstruction operator* K.

(S4) We now iterate the above steps (S1) - (S3), until a reconstruction of the kernel w of sufficient precision is achieved. We might have to stop the iterations after some steps, since error in the measurements may otherwise spoil the reconstruction. This will be discussed in more detail in the upcoming sections. Here, we now integrate our iteration steps into the formula

$$w^{(\ell)} = K(E(T(w^{(\ell-1)}))), \quad \ell = 1, 2, 3, ... \tag{5.4}$$

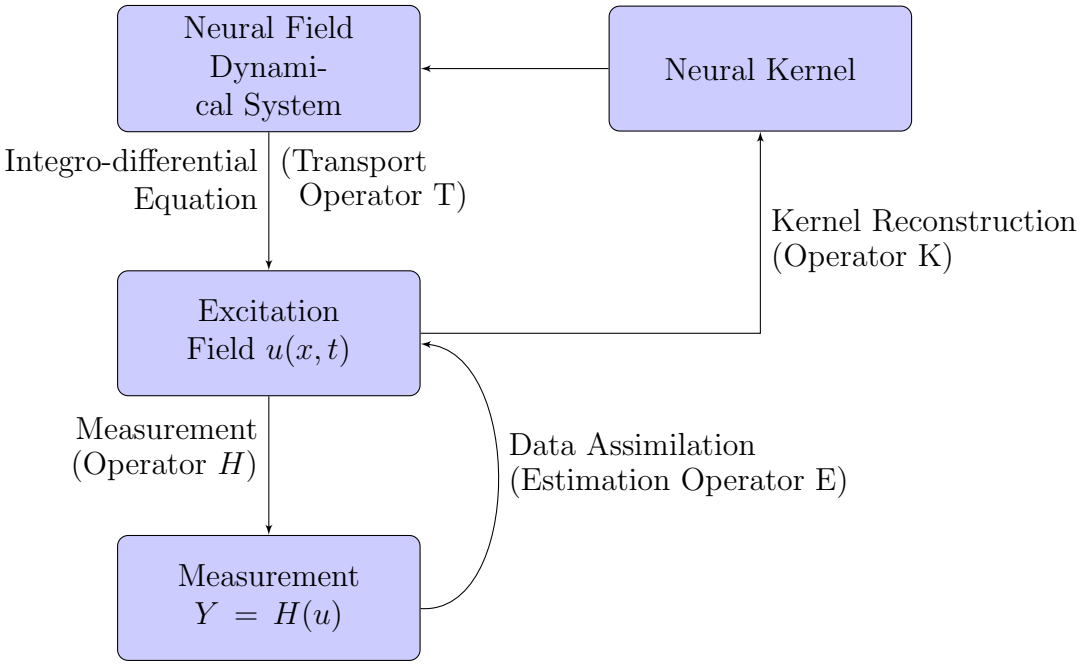


Figure 5.1: The main idea and components of iterative approach based on inversion and data assimilation to simulate neural system dynamics. First, applying data assimilation we obtain the neural field u. Then, using the kernel reconstruction approach we approximate the kernel w.

Here, we have the special situation that both the mapping K as well as the mapping E are operators solving ill-posed problems. Both of them include regularization with some regularization parameter α_K and α_E, respectively. There are two important points to take into account here:

- In general, we know that for small parameters α the norm of regularization operators becomes large, i.e. they are bounded, but still lead to very ill-conditioned operations. As a consequence, in general we cannot expect the norm of KET to be bounded in any reasonable space. This means that in operator norm we do not expect the iteration (5.4) to converge, meaning that for each u_0 and u^{true} we have

$$||(KET)^n(u_0) - u^{(true)}|| \to 0, \quad n \to \infty, \tag{5.5}$$

  but not uniformly in u_0 or $u^{(true)}$. We will provide a more detailed convergence analysis in Sections 5.3 and 5.4.

- As for iterative methods to solve inverse problems, see for example Chapter 3 of [76] or [57], it might be possible to achieve convergence when an appropriate stopping rule is employed. In this case, we have an improvement of the kernel for its lower modes for several or even many iteration steps, until the influence of noise amplified by the ill-posed inverse takes over and destroys the achieved reconstructions.

At the end of this section, let us address an important question. Usually, in estimation theory where the error covariance matrix R and the background error covariance matrix B is employed, the errors are assumed to be statistically independent. When you calculate an analysis estimate of the form $u^a = BH^T(R + HBH^T)(y - Hu^b)$ for u from observations y, you introduce correlations between u^a and y. A second estimate, where now u^b is replaced by u^a would now violate the assumption of stochastical independence of the background uncertainty and the observation error distribution.

Do we fall into this problem with the iteration approach described above? We note that for our approach we try to solve the non-linear equation

$$(KE_yT)w = w, \tag{5.6}$$

where y is one choice of observations parametrizing the operator E_y, which maps Tw onto an excitation field $u = E_yTw$. The data y in (5.6) are parametrizing the operator $A := KE_yT$, which is set-up once. After this, we are merely carrying out the standard iteration $A^n(w_0)$, which in the case where A is a contraction will converge to the unique solution of $A(w) = w$.

So the main point here is to investigate the stochastical properties of the formulation of the full non-linear inverse kernel reconstruction problem as equation (5.6). We realize that the arguments are not completely settled here, and that stochastical analysis of the iterations should be studied in more detail in future research.

5.2 Analysis of the Algorithmic Components

The task of this section is to analyse the different parts of the iterative algorithms. Firstly, we study the neural transport map T in Section 5.2.1, considered as a mapping $w \mapsto T(w)$ of the neural kernel onto the solution of the neural field equation. Secondly, the state estimation or data assimilation map E is investigated in Section 5.2.2, which when you consider it as a mapping $u^{(b)} \mapsto u^{(a)}$ of the background $u^{(b)}$ onto the analysis $u^{(a)}$ has some contraction properties. The third component, the *kernel reconstruction* K, is analysed in Section 5.2.3, studied as a mapping $u^{(a)} \mapsto w$.

5.2.1 The Transport Map: Dependence on the Kernel

We assume that the neural activity function u obeys the Amari neural field equation without delay

$$\tau \frac{\partial u}{\partial t}(x,t) = -u(x,t) + \int_\Omega w(x,y) f(u(y,t))\, dy, \quad x \in \Omega, t \in [0,T], \tag{5.7}$$

with initial condition $u(x,0) = u_0$. We note that when reconstructing w and u, we do neither know w nor do we know u_0 or u when we start our reconstruction. The task is to reconstruct all these. However, for our analysis we will assume that we are given u_0 and finally argue what can be done if u_0 is unknown. It has been shown in [85] that the equation can be solved by transforming it into an integral equation. Here, we study the dependence of the solution u on the neural kernel w. We first present our main result and then carry out the proof in several steps.

THEOREM 5.2.1. *We assume that f is a Lipschitz continuous function $\mathbb{R} \to [0,1]$ and $\tau > 0$ is a constant. Let u_1, u_2 be two solutions of the neural field equation (5.7) with kernels w_1 and w_2, respectively, and with initial condition $u_j(\cdot,0) = u_0^{(j)}$, for $j = 1,2$. Then, there is a constant C such that*

$$||u_1(\cdot,t) - u_2(\cdot,t)||_\infty \leq C\Big(||u_0^{(1)} - u_0^{(2)}||_\infty + ||w_1 - w_2||_\infty\Big) \tag{5.8}$$

for $t \in [0,T]$.

Proof. We proceed in several steps.

I. We formulate an integral equation for the difference $\delta u := u_1 - u_2$ between the two activity functions u_1 and u_2. Each of these functions satisfies the equation

$$\begin{aligned} \tau(u_j(x,t) - u_j(x,0)) &= -\int_{s=0}^{t} u_j(x,s)\, ds \\ &\quad + \int_{s=0}^{t}\int_{\Omega} w_j(x,y) f(u_j(y,s))\, dy\, ds \end{aligned} \tag{5.9}$$

for $j = 1, 2$, $x \in \Omega$ and $t \in [0, T]$. Here, we consider the slightly more general situation where initial values of $\delta u(x,t)$ are given by the fixed function $\gamma(x) := \delta u(x,0)$, $x \in \Omega$. Then, for δu we derive

$$\begin{aligned} \tau\delta u(x,t) &= \gamma(x) - \int_{s=0}^{t} \delta u(x,s)\, ds \\ &\quad + \int_{s=0}^{t}\int_{\Omega} w_1(x,y) f(u_1(y,s))\, dy\, ds \\ &\quad - \int_{s=0}^{t}\int_{\Omega} w_2(x,y) f(u_2(y,s))\, dy\, ds \\ &= -\int_{s=0}^{t} \delta u(x,s)\, ds \\ &\quad + \int_{s=0}^{t}\int_{\Omega} w_1(x,y)\Big(f(u_1(y,s)) - f(u_2(y,s))\Big)\, dy\, ds \\ &\quad + \gamma(x) + \int_{s=0}^{t}\int_{\Omega} \Big(w_1(x,y) - w_2(x,y)\Big) f(u_2(y,s)\, dy\, ds. \end{aligned} \tag{5.10}$$

For the function $f : \mathbb{R} \to \mathbb{R}^+$ we note that

$$f(s) \subset [0,1], \quad s \in \mathbb{R}. \tag{5.11}$$

Here, we will work with general Lipschitz continuous functions f satisfying this condition. We assume that the kernel w satisfies

- (H1) $w(x,\cdot) \in L^1(\Omega), \quad \forall\, x \in \Omega \subset \mathbb{R}^m$,

such that we obtain a *well defined* integral of the form

$$g(x,s) := \int_{\Omega} w(x,y) f(u(y,s)) dy, \quad x \in \Omega,\ s \in \mathbb{R},$$

The condition

- (H2) $\sup_{x\in\Omega} \|w(x,\cdot)\|_{L^1(\Omega)} \le C_1$.

with some constant C_1 leads to g being *bounded* on $\Omega \times \mathbb{R}$. We need $g(x,s)$ to be *continuous* in dependence of x and s, which for continuous functions u is achieved by the additional condition

- (H3) $\|w(x,\cdot) - w(x^*,\cdot)\|_{L^1(\Omega)} \to 0$ for $|x - x^*| \to 0$.

Then, from the existence Theorem 2.7 of [85] we know that there is a unique solution to the integral equations (5.9) for $j = 1, 2$ under the assumptions on f and w as stated in 5.11 and the conditions H1-3 earlier in this section.

We consider u_2 as fixed reference and write $u_1 = u_2 + \delta u$. Then, from (5.10) we conclude that δu satisfies the non-linear equation

$$\delta u = A(\delta u) + F \quad \text{on } \Omega \times [0,T] \tag{5.12}$$

with

$$A(\delta u) := \int_{s=0}^{t} \int_{\Omega} w_1(x,y) \Big(f(u_2(y,s) + \delta u(y,s)) - f(u_2(y,s)) \Big)\, dy\, ds \tag{5.13}$$

and the forcing term F defined by

$$F(x,t) := \gamma(x) + \int_{s=0}^{t} \int_{\Omega} \Big(w_1(x,y) - w_2(x,y) \Big) f(u_2(y,s)\, dy\, ds \tag{5.14}$$

for $x \in \Omega$ and $t \in [0,T]$.

II. Referring to the definition 2.1.7, with the arguments of Lemma 2.5 of [85], we conclude that A is a contraction on the space $BC(\Omega \times [0,\rho])$ for all w_1 with $|w_1| \le c$ with some constant c and with some sufficiently small constant $\rho > 0$ depending on the Lipschitz constant L of f and on c. Since F is a constant function not depending on δu, this contraction property also holds for the operator $A(\cdot)+F$. By the Banach fixed-point theorem, which is introduced in Section 2.1.4, there is a unique solution to the equation and the solution is the limit of the sequence

$$u_0 := 0, \quad \delta u_k := A(\delta u_{k-1}) + F, \quad k = 1, 2, 3, ... \tag{5.15}$$

We take the norm in (5.15) and estimate

$$\begin{aligned} ||\delta u_k|| &\le ||A(\delta u_{k-1})|| + ||F|| \\ &\le c_A ||\delta u_{k-1}|| + \sigma \end{aligned} \tag{5.16}$$

with the contraction constant $0 < c_A < 1$ of A and the constant $\sigma := ||F||$. Then, a bound α_k for $||\delta u_k||$ is defined by

$$\alpha_0 := 0, \quad \alpha_k := c_A \alpha_{k-1} + \sigma, \quad k = 1, 2, 3, ... \tag{5.17}$$

which by induction has the explicit representation

$$\alpha_k = \sigma \sum_{j=0}^{k-1} c_A^j + (c_A)^k \alpha_0 = \sigma \frac{1-c_A^k}{1-c_A} + (c_A)^k \alpha_0, \quad k = 1, 2, 3, ..., \tag{5.18}$$

where we kept the α_0 term to clarify that the following estimate does not depend on the choice of the initial field u_0. This leads to the estimate

$$||\delta u|| \leq \frac{1}{1-c_A} \sigma. \tag{5.19}$$

Finally, we estimate σ based on

$$||F(y,s)|| \leq ||\gamma|| + ||w_1 - w_2||_\infty \rho |\Omega| \tag{5.20}$$

to be given by $\sigma = \tilde{c}||w_1 - w_2||_\infty$ with the constant $\tilde{c} = \rho|\Omega|$.

Now, for the interval $[0, \rho]$ we employ $\gamma = u_0^{(1)} - u_0^{(2)}$ and the estimate (5.19) obtains the form

$$||\delta u|| \leq c\Big(||u_0^{(1)} - u_0^{(2)}||_\infty + ||w_1 - w_2||_\infty\Big) \tag{5.21}$$

in $BC(\Omega \times [0, \rho])$ with constant $c := (1 + \rho|\Omega|)/(1 - \rho)$.

III. The arguments from above are now repeated for the interval $[\rho, 2\rho]$. The full framework from above can be used, with the only difference that now $\gamma(x) := \delta u(x, \rho)$ needs to be employed. In this case, from the estimates on the first interval $[0, \rho]$ we have $||\gamma|| \leq c(||u_0^{(1)} - u_0^{(2)}|| + ||w_1 - w_2||_\infty)$, such that the estimate (5.21) holds also on $BC(\Omega \times [0, 2\rho])$, with a modified constant c. After $[T/\rho]$ steps (denoting the smallest integer larger than T/ρ) the estimate is achieved for the full interval $[0, T]$ with some constant C. This ends the proof for (5.8). □

As a special case of the above Theorem 5.2.1 for $u_0^{(1)} = u_0^{(2)}$, we have shown that the nonlinear mapping of the kernel w onto the solution u of the neural field equation (1.4) is both bounded and continuous from $BC(\Omega \times \Omega)$ into $BC(\Omega \times [0, T])$. The derivative of u is estimated as follows.

THEOREM 5.2.2. *The mapping of w onto the solution u of the neural field equation satisfies*

$$||u_1(x,t) - u_2(x,t)||_{BC(\Omega) \times BC^1([0,T])} \leq C\Big(||u_0^{(1)} - u_0^{(2)}||_\infty + ||w_1 - w_2||_\infty\Big) \tag{5.22}$$

with some constant C.

Proof. We employ equations (5.9) and the estimate (5.8). Differentiation of (5.9) with respect to t leads to

$$\left|\frac{\partial u_1}{\partial t}(x,t) - \frac{\partial u_2}{\partial t}(x,t)\right| \leq \left|\frac{\partial}{\partial t}\Big(A(u_1-u_2)(x,t) + F(x,t)\Big)\right| \tag{5.23}$$

with A given by (5.13) and F defined in (5.14). Again using the Lipschitz continuity of f estimating the derivative of A yields

$$\left|\frac{\partial}{\partial t}A(u_1-u_2)(x,t)\right| \leq \tilde{c}L||u_1-u_2||_\infty \tag{5.24}$$

with some constant $\tilde{c}$. Similarly, for F we estimate

$$\left|\frac{\partial}{\partial t}F(x,t)\right| \leq c||w_1-w_2||_\infty \tag{5.25}$$

with some constant c. Finally, with the help of the estimate (5.8) we now obtain (5.22) including the temporal derivative of the excitation function u. □

5.2.2 Assimilation in finite and infinite dimensional Setup

One of the main components of kernel reconstruction is the estimation of the state of the neural activity from given measurements. We call it either the *state estimation task* or the *data assimilation component* of the full problem. Here, we will focus on the three-dimensional variational approach to data assimilation (c.f. [76], Section 5.1).

In the following paragraphs we summarize the data assimilation setup either in the classical finite dimensional framework or in a Hilbert space environment. In the finite-dimensional case, the state space X is given by $X = \mathbb{R}^n$, $n \in \mathbb{N}$, and the observation space is $Y = \mathbb{R}^m$, $m \in \mathbb{N}$. In this case we study the setup where B is an invertible matrix in $\mathbb{R}^{n\times n}$, R is an invertible matrix in $\mathbb{R}^{m\times m}$, $y \in Y$ is the observation and $x \in X$ is the state.

In the case of the *infinite dimensional setup* we consider X and Y being the weighted L^2-type Hilbert spaces with scalar products $\langle x, \tilde{x}\rangle$ on X and $\langle y, \tilde{y}\rangle$ on Y. We define weighted scalar products by

$$\langle x, \tilde{x}\rangle_{B^{-1}} = \langle x, B^{-1}\tilde{x}\rangle \tag{5.26}$$

$$:= \int_\Omega x B^{-1}\tilde{x}\, d\mu, \quad x, \tilde{x} \in X \tag{5.27}$$

with some set Ω, a measure μ on Ω, measurable functions $x, \tilde{x}$ on Ω and a bounded self-adjoint and boundedly invertible operator $B : X \to X$ and an analogous setup on Y, i.e.

$$\begin{aligned} \langle y, \tilde{y} \rangle_{R^{-1}} &= \langle y, R^{-1}\tilde{y} \rangle && (5.28) \\ &:= \int_{\tilde{\Omega}} y R^{-1} \tilde{y} \, d\sigma, \quad y, \tilde{y} \in Y. && (5.29) \end{aligned}$$

We note that the notation (5.26) and (5.28) can be used both in the finite-dimensional and infinite-dimensional case and we will carry out our arguments in a uniform way for both cases.

The assimilation step based on a three-dimensional variational algorithm (3D-VAR) can be understood as the minimization of the functional

$$J(x) := ||x - x^{(b)}||^2_{B^{-1}} + ||y - Hx||^2_{R^{-1}}, \tag{5.30}$$

over $x \in X$. By standard functional analytic tools, here the calculation of the normal equations (see equation (5.2.14) of [76]), the minimizer of (5.30) is given by

$$\begin{aligned} x^{(a)} &= x^{(b)} + H^*(I + HH^*)^{-1}(y - Hx^{(b)}) && (5.31) \\ &= x^{(b)} + BH'(R + HBH^{-1})^{-1}(y - Hx^{(b)}), && (5.32) \end{aligned}$$

with the adjoint H' of H with respect to the scalar products $\langle x, \tilde{x} \rangle$ and $\langle y, \tilde{y} \rangle$ in X and Y, respectively, without weights.

The two equations (5.31) and (5.32) are transformed into each other based on $H^* = BH'R^{-1}$ (compare equation (5.2.11) of [76]). To study the dependence on R, we employ the adjoint H' with respect to the scalar products $\langle x, \tilde{x} \rangle$ and $\langle y, \tilde{y} \rangle$ (without weights B or R).

We note that from

$$\begin{aligned} \langle y, Hx \rangle &= \langle H'y, x \rangle \\ &= \langle H'y, BB^{-1}x \rangle \\ &= \langle BH'y, B^{-1}x \rangle \\ &= \langle BH'y, x \rangle_{B^{-1}} && (5.33) \end{aligned}$$

we see that $H^\dagger := BH'$ is the adjoint of the operator H with respect to the scalar products $\langle y, \tilde{y} \rangle$ on Y and $\langle x, \tilde{x} \rangle_{B^{-1}}$ on X.

Similarly, we obtain

$$H^* = BH'R^{-1} = H^\dagger R^{-1} \tag{5.34}$$

for the adjoint between the weighted scalar products $\langle x, \tilde{x}\rangle_{B^{-1}}$ and $\langle y, \tilde{y}\rangle_{R^{-1}}$, compare equation (5.2.11) of [76]. Using equation (5.2.12) of [76]

$$H^*(I + HH^*)^{-1} = (I + H^*H)^{-1}H^* \tag{5.35}$$

we can also write the assimilation step as

$$\begin{aligned} x^{(a)} &= x^{(b)} + (I + H^*H)^{-1}H^*(y - Hx^{(b)}) \\ &= x^{(b)} + (I + H^\dagger R^{-1}H)^{-1}H^\dagger R^{-1}(y - Hx^{(b)}). \end{aligned} \tag{5.36}$$

If $y = Hx^{(true)}$ is a measurement of the true state $x^{(true)}$, we obtain

$$\begin{aligned} x^{(a)} - x^{(true)} &= x^{(b)} - x^{(true)} \\ &\quad +(I + H^\dagger R^{-1}H)^{-1}H^\dagger R^{-1}H(x^{(true)} - x^{(b)}) \\ &= \Big[I - (I + H^\dagger R^{-1}H)^{-1}H^\dagger R^{-1}H\Big](x^{(b)} - x^{(true)}) \\ &= (I + H^\dagger R^{-1}H)^{-1}\Big[(I + H^\dagger R^{-1}H) - H^\dagger R^{-1}H\Big](x^{(b)} - x^{(true)}) \\ &= (I + H^\dagger R^{-1}H)^{-1}(x^{(b)} - x^{(true)}). \end{aligned} \tag{5.37}$$

Equation (5.37) can be used to study bounds for the analysis error. We formulate this result in a particular form as a lemma.

LEMMA 5.2.3. *Let $R := \alpha R_0$ with $\alpha \in \mathbb{R}^+$ and R_0 be a self-adjoint boundedly invertible operator on Y. Then, under the assumption of perfect observations, we have the estimate*

$$\begin{aligned} ||x^{(a)} - x^{(true)}||_{B^{-1}} &= ||(I + H^\dagger(\alpha R_0)^{-1}H)^{-1}(x^{(b)} - x^{(true)})||_{B^{-1}} \\ &\leq ||\alpha(\alpha I + H^\dagger R_0^{-1}H)^{-1}(x^{(b)} - x^{(true)})||_{B^{-1}} \end{aligned} \tag{5.38}$$

and the *pointwise* convergence

$$||\alpha(\alpha I + H^\dagger R_0^{-1}H)^{-1}x)||_{B^{-1}} \to 0, \quad \alpha \to 0 \tag{5.39}$$

for each fixed $x \in X$.

We note that in the above lemma the choice of $y = Hx^{(true)}$ corresponds to convergence in the case where y has no measurement error. The assumption is needed to obtain a convergence to a true solution when measurement error tends to zero or is zero.

Proof. The inequality (5.38) is an immediate consequence of (5.37) and the choice $R = \alpha R_0$. We first note that $H^\dagger R_0^{-1} H$ is self-adjoint with respect to the scalar product $\langle x, \tilde{x} \rangle_{B^{-1}}$, since we have

$$\begin{aligned} \left\langle x, H^\dagger R_0^{-1} H \tilde{x} \right\rangle_{B^{-1}} &= \left\langle Hx, R_0^{-1} H \tilde{x} \right\rangle \\ &= \left\langle R_0^{-1} Hx, H\tilde{x} \right\rangle \\ &= \left\langle H^\dagger R_0^{-1} Hx, \tilde{x} \right\rangle_{B^{-1}}. \end{aligned} \tag{5.40}$$

This means there is an orthonormal system $\{\varphi_n, n \in \mathbb{N}\}$ in $(X, \langle , \rangle_{B^{-1}})$ and a sequence $\{\lambda_n, n \in \mathbb{N}\}$ of real positive eigenvalues ordered according to its size $\lambda_1 \geq \lambda_2 \geq ... \geq 0$, such that

$$H^\dagger R_0^{-1} Hx = \sum_{n=1}^{\infty} \lambda_n \langle \varphi_n, x \rangle \varphi_n, \tag{5.41}$$

where the sum is finite in the finite-dimensional case. Thus, the operator $\alpha(\alpha + H^\dagger R_0^{-1} H)^{-1}$ has the spectral representation

$$\alpha(\alpha I + H^\dagger R_0^{-1} H)^{-1} x = \sum_{n=1}^{\infty} \frac{\alpha}{\alpha + \lambda_n} \langle \varphi_n, x \rangle_{B^{-1}} \varphi_n. \tag{5.42}$$

Now, for $x \in X$ fixed given $\epsilon > 0$ we can first choose N such that

$$\sum_{n=N+1}^{\infty} | \langle \varphi_n, x \rangle_{B^{-1}} |^2 \leq \frac{\epsilon}{2}. \tag{5.43}$$

Then, since $\alpha/(\alpha + \lambda_N) \to 0$ for $\alpha \to 0$ we can choose $\alpha > 0$ small enough such that

$$(\frac{\alpha}{\alpha + \lambda_N})^2 \sum_{n=1}^{N} | \langle \varphi_n, x \rangle_{B^{-1}} |^2 \leq \frac{\epsilon}{2}. \tag{5.44}$$

This yields

$$\begin{aligned} \sum_{n=1}^{\infty} \left(\frac{\alpha}{\alpha + \lambda_n} \right)^2 | \langle \varphi_n, x \rangle_{B^{-1}} |^2 &\leq \left(\frac{\alpha}{\alpha + \lambda_N} \right)^2 \sum_{n=1}^{N} | \langle \varphi_n, x \rangle_{B^{-1}} |^2 \\ &\quad + \sum_{n=N+1}^{\infty} | \langle \varphi_n, x \rangle_{B^{-1}} |^2 \\ &\leq \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon. \end{aligned} \tag{5.45}$$

This shows the pointwise convergence (5.39), and the proof is complete. □

The convergence of Lemma 5.2.3 is carried out with respect to the norm $||\cdot||_{B^{-1}}$ on X. We note that since B and B^{-1} are bounded linear operators on X, we have for $x \in X$ the estimate

$$||x||^2_{B^{-1}} = \left\langle x, B^{-1}x \right\rangle \leq ||x|| \cdot ||B^{-1}x|| \leq C||x||^2 \tag{5.46}$$

with the constant $C := ||B^{-1}||$, leading to

$$||x||_{B^{-1}} \leq \sqrt{||B^{-1}||} \cdot ||x||. \tag{5.47}$$

We also note that we have

$$\begin{aligned} ||x||^2 &= \langle x, x \rangle = \left\langle x, BB^{-1}x \right\rangle = \left\langle Bx, B^{-1}x \right\rangle \\ &= \langle Bx, x \rangle_{B^{-1}} \leq ||Bx||_{B^{-1}} \cdot ||x||_{B^{-1}} \end{aligned} \tag{5.48}$$

and

$$||Bx||^2_{B^{-1}} = \langle Bx, Bx \rangle_{B^{-1}} = \langle Bx, x \rangle \leq ||B|| \cdot ||x||^2, \tag{5.49}$$

such that

$$||Bx||_{B^{-1}} \leq \sqrt{||B||} \cdot ||x|| \tag{5.50}$$

and from (5.48)

$$||x||^2 \leq \sqrt{||B||} \cdot ||x|| \cdot ||x||_{B^{-1}} \tag{5.51}$$

which dividing by $||x||$ finally yields

$$||x|| \leq \sqrt{||B||} \cdot ||x||_{B^{-1}}. \tag{5.52}$$

COROLLARY 5.2.4. *Under the assumptions of Lemma 5.2.3 the* pointwise *convergence (5.39) also holds in* $||\cdot||$ *of* X*, i.e. we have*

$$||\alpha(\alpha I + H^\dagger R_0^{-1} H)^{-1} x)|| \to 0, \quad \alpha \to 0 \tag{5.53}$$

for each fixed $x \in X$*.*

For later use, we finally need to discuss the estimation of the derivative $\frac{\partial x^{(a)}}{\partial t}$ of the estimated activity function $u = x^{(a)}$ and its dependence on the first guess. Clearly, the temporal change $\frac{\partial u}{\partial t}$ of the activity $u = x^{(a)}$ depends on the temporal change of the true data $y^{(true)}(t) = H(x^{(true)}(t))$, $t \in [0, T]$, and the temporal structure of the measurement noise. The true excitation is $BC^1([0, T])$ since it

obeys the Amari equation. If H depends on space only, then the true data $y^{(true)}$ is also an element of $BC^1([0,T])$. We note that we can differentiate the term

$$z := \alpha(\alpha I + H^\dagger R_0^{-1} H)^{-1} x \tag{5.54}$$

in (5.53) with respect to time, to obtain

$$\frac{\partial z}{\partial t} := \alpha(\alpha I + H^\dagger R_0^{-1} H)^{-1} \frac{\partial x}{\partial t}. \tag{5.55}$$

This means that the temporal derivative $\frac{\partial u}{\partial t}$ of the reconstructed u shows the same convergence behaviour (5.53) as the function u itself.

5.2.3 Kernel Reconstruction: Dependence on the Analysis

For the final step we analyse the full-field kernel reconstruction method. We have introduced the problem in Definition 3.3.2 and discussed thoroughly in Section 3. In this section we consider infinite speed $v = \infty$ as it has been introduced and discussed in a series of papers [85, 12, 84, 87]. In this section we develop norm estimates for the dependence of the kernel w on the field u. These could be derived as a consequence of the sensitivity analysis of Chapter 3.5 or by elementary norm estimates. Here, we choose the elementary approach, which will also provide detailed insight into the evolution of the constants.

Here, we follow Equations 3.27 and 3.26, which is based on the functions

$$\psi(r,t) := \tau \frac{\partial u(r,t)}{\partial t} + u(r,t), \quad r \in \Omega, \ t \in [0,T] \tag{5.56}$$

and

$$\varphi(r,t) := f(u(r,t)), \quad r \in \Omega, \ t \in [0,T]. \tag{5.57}$$

The inverse problem is then given by the *family of integral equations*

$$\psi(r,t) = \int_\Omega \varphi(r',t) w(r,r') \, dr', \quad t \in [0,T] \tag{5.58}$$

which, since the function $\varphi(r',t)$ is continuous in $r' \in \Omega$ and $t \in [0,T]$, for each point $r \in \Omega$ is an *integral equation of the first kind*.

It is well known (e.g. [76], Chapter 3) that the inversion of an integral equation of the first kind is ill-posed, i.e. the inverse operator is not bounded. However, using regularization techniques it is possible to approximate the unbounded inverse

by a bounded inverse. Here, we employ classical Tikhonov regularization for the inversion of (5.58). We define the integral operator

$$(V\eta)(t) := \int_\Omega \varphi(r',t)\eta(r')\,dr', \quad t \in [0,T] \tag{5.59}$$

as an operator from $L^2(\Omega)$ into $L^2([0,T])$. Then, its adjoint $V^* : L^2([0,T]) \to L^2(\Omega)$ is given by

$$\begin{aligned} \langle V\eta, \sigma\rangle &= \int_0^T \int_\Omega \varphi(r',s)\eta(r')\,\sigma(s)\,dr'\,ds \\ &= \int_\Omega \eta(r) \int_0^T \varphi(r,s)\sigma(s)\,ds\,dr \\ &= \langle \eta, V^*\sigma\rangle\,, \end{aligned} \tag{5.60}$$

such that

$$(V^*\sigma)(r) = \int_0^T \varphi(r,s)\sigma(s)\,ds, \quad r \in \Omega. \tag{5.61}$$

The family (5.58) of integral equations can be written in the form

$$Vw(r,\cdot) = \psi(r,\cdot), \quad r \in \Omega. \tag{5.62}$$

The regularized solution of (5.58) is given by

$$\begin{aligned} w_\alpha(r,t) &:= \Big((\alpha I + V^*V)^{-1}V^*\psi(r,\cdot)\Big)(t) \\ &= \Big(V^*(\alpha I + VV^*)^{-1}\psi(r,\cdot)\Big)(t), \quad r \in \Omega,\ t \in [0,T], \end{aligned} \tag{5.63}$$

where we used Equation 3.45.

We note some useful properties of this regularized inverse.

LEMMA 5.2.5. *For each fixed $\alpha > 0$ the operator $(\alpha I + VV^*)^{-1}$ maps $L^2([0,T])$ boundedly into itself. Further, the operator V^* maps $L^2([0,T])$ boundedly into $BC(\Omega)$, and thus the same is true for $V^*(\alpha I + VV^*)^{-1}$.*

Proof. The kernel $\varphi(r',s)$ is continuous for $r' \in \Omega$ and $s \in [0,T]$, where Ω is a bounded set. Thus, the operators V and V^* define compact bounded linear operators $V : L^2(\Omega) \to L^2([0,T])$ and $V^* : L^2([0,T]) \to L^2(\Omega)$. The boundedness of the inverse of $\alpha I + VV^*$ on $L^2([0,T])$ as well as the boundedness of the inverse of $\alpha I + V^*V$ on $L^2(\Omega)$ is a consequence of the structure of the self-adjoint

operators and the compactness of V and V^*. As a final step, consider $V^*\sigma$ with $\sigma \in L^2([0,T])$. The function is bounded on Ω by use of the Cauchy-Schwartz inequality

$$\begin{aligned} \Big|\Big|(V^*\sigma)(r)\Big|\Big|_\infty &= \sup_{r\in\Omega}\Big|\int_0^T \varphi(r,s)\sigma(s)\,ds\Big|_\infty \\ &\leq \Big(\int_0^T |\varphi(r,s)|^2\,ds\Big)^{\frac{1}{2}}\Big(\int_0^T |\sigma(s)|^2\,ds\Big)^{\frac{1}{2}} \\ &\leq T^{\frac{1}{2}}||\varphi||_\infty||\sigma||_{L^2([0,T])}. \end{aligned} \tag{5.64}$$

We also note that, similarly, $(V^*\sigma)(r)$ depends continuously on $r \in \Omega$, such that the second statement of the lemma is obtained and the proof is complete. □

We can now summarize the boundedness and convergence for the regularized solution of the family (5.58) of integral equations in the following theorem. Since ψ is the sum of the derivative of u with respect to time t (times a constant) and u itself, we need u to be in $H^1([0,T])$, the closure of the space of functions $u \in C^1([0,T])$ whose temporal derivatives are in $L^2([0,T])$. Clearly, the mapping $u \mapsto \psi = \tau\frac{\partial u}{\partial t} + u$ is bounded from $H^1([0,T]) \to L^2([0,T])$.

THEOREM 5.2.6. *For each fixed $\alpha > 0$ the regularized reconstruction of w from u by (5.63) is a bounded mapping from $BC(\Omega)\times H^1([0,T])$ into $BC(\Omega)\times L^2([0,T])$ and $BC(\Omega\times[0,T])$ into $BC(\Omega\times[0,T])$. If u is a true solution of some Amari type equation (1.4) with kernel $w^{(true)}$, then for $\alpha \to 0$ we have convergence*

$$V^*(\alpha I + VV^*)^{-1}\psi(r,\cdot) \to w^*(r,\cdot) \tag{5.65}$$

in $L^2(\Omega)$, where w^* is the projection of $w^{(true)}$ onto $N(V)^\perp$.

Proof. We note that by continuity of the kernel V is a compact linear operator $L^2(\Omega) \to L^2([0,T])$. It is injective on $N(V)^\perp$ since for $\sigma \in N(V)^\perp$ from $V\sigma = 0$ we conclude $\sigma \in N(V)\cap N(V)^\perp = \{0\}$, thus $\sigma = 0$. Clearly, we have $Vw = Vw^*$ since $w - w^* \in N(V)$ by definition. Now, we apply classical convergence theory for Tikhonov regularization in Hilbert spaces, as stated in Theorem 2.3.4 , to conclude the convergence (5.65). □

As a second step of our analysis we need to investigate the dependence of the operator V on the activity function u. We note that the kernel $\varphi = f(u)$ is a non-linear function of the activity u. In the case where f is Lipschitz continuous, we obtain the following result.

THEOREM 5.2.7. *The operator V defined by (5.59) depends continuously on $u \in BC(\Omega \times [0,T])$ and it is bounded in the sense that the mapping*

$$V : BC(\Omega \times [0,T]) \to BL\Big(BC(\Omega), BC([0,T])\Big), \quad u \mapsto V[u] \tag{5.66}$$

satisfies

$$||V[u] - V[\tilde{u}]||_\infty \leq C||u - \tilde{u}||_\infty \tag{5.67}$$

with some constant C, where the norm of $V[u]$ is the operator norm in the spaces of bounded continuous functions. An analogous estimate is true for V^* defined in (5.61).

Proof. We employ the Lipschitz continuity of the kernel to estimate

$$\begin{aligned} ||V[u]\eta - V[\tilde{u}]\eta||_\infty &= \Big| \int_\Omega \Big(f(u(r',t)) - f(\tilde{u}(r',t)) \Big) \eta(r')\, dr' \Big| \\ &\leq \int_\Omega |f(u(r',t)) - f(\tilde{u}(r',t))||\eta r'|\, dr' \\ &\leq L \int_\Omega |u(r',t) - \tilde{u}(r',t)|\, dr' \cdot ||\eta||_\infty \\ &\leq L|\Omega| \cdot ||u - \tilde{u}||_\infty \cdot ||\eta||_\infty, \end{aligned} \tag{5.68}$$

where L is a Lipschitz constant. With the operator norm $||V[u]||$ defined as

$$||V[u]||_\infty := \sup_{||\eta||_\infty > 0} \frac{||V[u]\eta||_\infty}{||\eta||_\infty}, \tag{5.69}$$

this implies (5.67). The derivation for V^* works the same way. □

The dependence of $(\alpha I + V^*V)^{-1}V^*$ on u is now estimated in a canonical way, based on general estimates for operators $A, \tilde{A}$ and $B, \tilde{B}$

$$||BA - \tilde{B}\tilde{A}|| \leq ||B - \tilde{B}|| \cdot ||A|| + ||\tilde{B}|| \cdot ||A - \tilde{A}||. \tag{5.70}$$

and

$$\begin{aligned} ||A^{-1} - \tilde{A}^{-1}|| &\leq ||A^{-1}A(A^{-1} - \tilde{A}^{-1})\tilde{A}\tilde{A}^{-1}|| \\ &\leq ||A^{-1}|| \cdot ||\tilde{A} - A|| \cdot ||\tilde{A}^{-1}||, \end{aligned} \tag{5.71}$$

where we first apply (5.70) to $A = V[u]$, $B = V^*[u]$, $\tilde{A} = V[\tilde{u}]$ and $\tilde{B} = V^*[\tilde{u}]$, then the estimate (5.71) to $A = \alpha + V^*[u]V[u]$ and $\tilde{A} = \alpha + V^*[\tilde{u}]V[\tilde{u}]$ and finally the estimate (5.70) to $A = (\alpha + V^*[u]V[u])^{-1}$, $B = V^*[u]$, $\tilde{A} = (\alpha + V^*[\tilde{u}]V[\tilde{u}])^{-1}$ and $\tilde{B} = V^*[\tilde{u}]$.

We abbreviate

$$W_\alpha[u]\eta := (\alpha I + V^*[u]V[u])^{-1}V^*[u]\eta. \tag{5.72}$$

From the arguments above we obtain

$$||W_\alpha[u]\eta - W_\alpha[\tilde{u}]\eta|| \leq c_\alpha ||u - \tilde{u}||_{BC(\Omega\times[0,T])}||\eta|| \tag{5.73}$$

with some constant c_α depending on $\alpha > 0$.

We now finally estimate the total dependence of the regularized solution (5.63) of (5.58) on u and formulate the result as a theorem.

THEOREM 5.2.8. *For each $\alpha > 0$, the mapping $u \mapsto w_\alpha = W_\alpha[u]\psi[u]$ defined by (5.72), (5.63), (5.57) and (5.56) satisfies the estimate*

$$||W_\alpha[u]\psi[u] - W_\alpha[\tilde{u}]\psi[\tilde{u}]|| \leq C_\alpha ||u - \tilde{u}||, \quad \tilde{u} \in BC(\Omega) \times BC^1([0,T]) \tag{5.74}$$

with some constant C_α depending on α and u.

Proof. Based on the estimates (5.70) and (5.71) applied to the components of $W_\alpha[u]$ and $W_\alpha[\tilde{u}]$ defined in (5.72) we only need to carry out the final step to estimate $||W_\alpha[u]\psi[u] - W_\alpha[\tilde{u}]\psi[\tilde{u}]||$ based on the estimates for $W_\alpha[u]$ and $W_\alpha[\tilde{u}]$.

As stated in Theorem 5.66 for ψ defined by (5.56) we have

$$||\psi[u] - \psi[\tilde{u}]||_{L^2(\Omega)\times L^2([0,T])} \leq (1+\tau)||u - \tilde{u}||_{L^2(\Omega)\times H^1([0,T])}, \tag{5.75}$$

where the space $H^1([0,T])$ is needed since ψ involves the temporal derivative of the excitation function u. We now employ the standard decomposition

$$\begin{aligned} ||W_\alpha[u]\psi[u] - W_\alpha[\tilde{u}]\psi[\tilde{u}]|| &\leq ||W_\alpha[u]\psi[u] - W_\alpha[\tilde{u}]\psi[u]|| \\ &\quad + ||W_\alpha[\tilde{u}](\psi[u] - \psi[\tilde{u}])|| \\ &\leq C_\alpha ||u - \tilde{u}||_{L^2(\Omega)\times H^1([0,T])}, \end{aligned} \tag{5.76}$$

with some constant C_α dependent on α based on (5.73) for the first term and (5.75) for the second. This completes the proof. □

5.3 The iterative kernel and state reconstruction approach

We here use the notation x for the state to introduce the data assimilation discussion. Let Z denote the space of kernels w under consideration. The excitation functions u live on Ω for all times $t \in [0,T]$. We assume that we have measurements at times t_k for $k = 1, ..., n_k$. Then, we use the letter $x_k^{(b)}$ to denote the excitation state at time t_k and

$$x^{(b)} = \left\{ x_k^{(b)}, \quad k = 1, ..., n_k \right\}. \tag{5.77}$$

We denote this state space by $\mathcal{X} = X \times \cdots \times X$, where X is the state space of excitations for one given point in time. We can work with $X = L^2(\Omega)$ or $X = C(\Omega)$.

As introduced in (5.1), we write the simulation of the neural excitation as a mapping of the kernel w onto the background state $x^{(b)}$, i.e.

$$x^{(b)} = T(w), \quad w \in Z. \tag{5.78}$$

The letter T is chosen due to the *transport* activity of this operator, which transports the initial state through time based on the kernel w.

As introduced in (5.2), the estimation of the neural states from measured data y is written in the form

$$x^{(a)} = E(x^{(b)}), \tag{5.79}$$

with the *estimation operator* E, where E depends on the data

$$y = \{y_k, \quad k = 1, ..., n_k\}, \tag{5.80}$$

with $y_k \in Y$ being the measurements at time t_k, living in the observation space Y. Since for the iteration process, which we are studying, this data y is constant, we will not explicitly include it into our notation.

As introduced in (5.3), the regularized solution of the inverse kernel problem is denoted by K_α, i.e.

$$w^{(a)} = K(x^{(a)}). \tag{5.81}$$

The letter K is chosen since this is the *kernel estimation* step. The operator K is a nonlinear operator in the excitation states $u = x^a$ defined in (5.77), defined by

$$\psi(r,t) = \tau \frac{\partial u}{\partial t}(r,t) + u(r,t), \quad r \in \Omega, t \in [0,T], \tag{5.82}$$

$$\varphi(r,t) := f(u(r,t)), \quad r \in \Omega, \quad t \in [0,T] \tag{5.83}$$

and

$$\psi(r,t) = \int_\Omega \varphi(r',t) w(r,r')\, dr', \quad r \in \Omega, \quad t \in [0,T], \tag{5.84}$$

where we note that u is another notation for the state vector x^a in the neural field framework, x^a is the standard data assimilation notation for the result of the state estimation problem.

As introduced in (5.4), we summarize the three steps into the notation

$$S = K(E(T(\cdot))), \tag{5.85}$$

where S is a non-linear operator on the kernel space Z and the letter S is chosen since this is one *iteration or estimation step* of the *iterative data assimilation kernel estimation* approach.

5.4 Convergence Proof

In this section, we provide a proof of the convergence of the iteration method we have introduced in this chapter. The proof is achieved under assumption that the observation operator is invertible and in the case where perfect observations are given.

In the case where you have imperfect observations, clearly convergence cannot be expected. Depending on the size of the observation error, you need a stopping rule to stop the iterations. Here we will focus on showing the potential of the method to provide a full kernel reconstruction when observations have no error.

We first formulate three lemmas on which our convergence analysis is based.

LEMMA 5.4.1. *For the non-linear operator T we have*

$$||T(w) - T(\tilde{w})||_{BC(\Omega)\times BC^1([0,T])} \leq c||w - \tilde{w}||_\infty \tag{5.86}$$

with some constant c and with the infinity norm $||w||_\infty = ||w||_{BC(\Omega\times\Omega)}$.

Proof. The result is a consequence of the continuous dependence of the neural field equation on the kernel with respect to the norm $L^2(\Omega \times \Omega)$ or $C(\Omega \times \Omega)$ shown in Theorem 5.2.2. □

Here, $T_k : w \mapsto u_k^{(b)}$ is considered as a mapping from $BC(\Omega \times \Omega)$ into $BC(\Omega)$ We also note that

$$||u^{(b)}||_{L^2(\Omega)} \leq C||u^{(b)}||_{BC(\Omega)} \tag{5.87}$$

with the constant $C = |\Omega|^{\frac{1}{2}}$. We will consider our estimation step in the spaces of square-integrable functions $L^2(\Omega)$ or $L^2_{B^{-1}}(\Omega)$. We note that since the estimation takes place with respect to the spatial variables only, estimates for the temporal derivatives of the functions are obtained by simple differentiation, which can be pulled through the operator mapping of the estimation step. Numerically we apply a finite difference approximation, which is a convergent regularization scheme for calculating the derivative of u with respect to time.

LEMMA 5.4.2. *For the operator E between the spaces $L^2(\Omega)$ and $L^2_{B^{-1}}(\Omega)$ we have*

$$||E(x) - E(\tilde{x})|| \leq \rho ||x - \tilde{x}|| \tag{5.88}$$

on the state space X, where the constant ρ depends on the covariance operator R. For a covariance matrix $R = \alpha R_0$ in the case where H is invertible, $||B^{-1}|| \leq c$ and $||H^{-1}|| \leq C$ with constants $c, C > 0$ we obtain the asymptotics

$$\rho \to 0 \quad \textit{for} \quad \alpha \to 0, \tag{5.89}$$

i.e. for α sufficiently small E is a contraction and the contraction constant ρ tends to zero when the observation error $R = \alpha R_0$ tends to zero.

Proof. The operator E is given by n_k equations

$$x_k^{(a)} = x_k^{(b)} + BH^T(R + HBH^T)^{-1}(y_k - Hx_k^{(b)}), \quad k = 1, ..., n_k. \tag{5.90}$$

For $k = 1, ..., n_k$, we calculate

$$x_k^{(a)} = \Big(I - BH^T(R + HBH^T)^{-1}H\Big)x_k^{(b)} + BH^T(R + HBH^T)^{-1}y_k \tag{5.91}$$

Thus, we obtain

$$E_k(x) - E_k(\tilde{x}) = (I - BH^T(R + HBH^T)^{-1}H)(x_k - \tilde{x}_k) \tag{5.92}$$

We note that using equation (5.34) ,we have

$$\begin{aligned} BH^T(\alpha R_0 + HBH^T)^{-1} &= BH^T R_0^{-1}(\alpha I + HBH^T R_0^{-1})^{-1} \\ &= H^*(\alpha I + HH^*)^{-1}, \\ &= (\alpha I + H^*H)^{-1}H^* \end{aligned} \tag{5.93}$$

by using equation (5.35) such that we can write (5.92) in the form

$$\begin{aligned} E_k(x) - E_k(\tilde{x}) &= \Big(I - (\alpha I + H^*H)^{-1}H^*H\Big)(x_k - \tilde{x}_k) \\ &= \Big((\alpha I + H^*H)^{-1}(\alpha I + H^*H) - H^*H)\Big)(x_k - \tilde{x}_k) \\ &= \alpha(\alpha I + H^*H)^{-1}(x_k - \tilde{x}_k) \end{aligned} \tag{5.94}$$

for $k = 1, ..., n_k$. The singular value representation of (5.94) following (3.38) where μ_n are the singular values of the operator H is given by

$$E_k(x) - E_k(\tilde{x}) = \sum_{n=1}^{\infty} \frac{\alpha}{\alpha + \mu_n^2} \langle g_n, x_k - \tilde{x}_k \rangle \, g_n. \tag{5.95}$$

This proves (5.88) with $\rho = 1$ in $L^2_{B^{-1}}(\Omega)$ with weighted norm (5.26), which by (5.52) is carried over to $L^2(\Omega)$ with some different constant ρ. In the case where H is invertible with a bound on the inverse, we have $\mu_n \geq c > 0$ with some constant c. This proves the second statement (5.89). □

Further, differentiating both sides of (5.94) with respect to t yields the estimate in the space $L^2(\Omega) \times BC^1([0,T])$. At this point, we need to point out an important ingredient needed for convergence. The kernel reconstruction relies on the knowledge of the field u and its temporal derivative $\frac{\partial u}{\partial t}$. The derivative can be reconstructed

- either by measuring the temporal derivative of the data y_k and then using the estimation as above to reconstruct the temporal derivative of the field u,
- or by measuring y_k as above and after reconstruction of u_k using a method like finite differences to estimate $\frac{\partial u_k}{\partial t}$ at t_k.

Clearly, the reconstruction of the derivative from function values is ill-posed itself. Convergence relies on the convergence of the measured data y_k to the true measurements in combination with an appropriate regularization scheme. Finite differences with a temporal grid h_t which is chosen to tend to zero carefully when the error in y_k tends to zero can achieve this. Classical regularization theory applies, for further discussion about regularization we refer to [37, 76].

We note the boundedness of the kernel reconstruction operator $K : u^{(a)} \mapsto w$ defined in (5.3) and (5.63) in the following lemma.

LEMMA 5.4.3. *For each fixed $\alpha > 0$ there is a constant C_α such that*

$$||K_\alpha(u^{(a)}) - K_\alpha(\tilde{u}^{(a)})||_{BC(\Omega\times\Omega)} \leq C_\alpha ||u^{(a)} - \tilde{u}^{(a)}||_{L^2(\Omega)\times H^1([0,T])} \tag{5.96}$$

for all $u^{(a)}, \tilde{u}^{(a)} \in \mathcal{X}$.

Proof. The theorem is just a reformulation of Theorem 5.2.8, where K defined in (5.3) can be seen to be identical to $K(u) = W_\alpha \psi[u]$ with W_α given by (5.72) and ψ given in (5.56). □

We are now prepared to formulate our main result on the operator S defined in (5.85). For the following Lemma we employ the notation E_ρ to refer to the relationship between the estimation operator E and the parameter ρ defined in Lemma 5.4.2.

THEOREM 5.4.4. *For ρ sufficiently small, the operator $S_\rho = KE_\rho T$ given in (5.85) is a contraction on the space Z of kernels equipped with the norm $L^2(\Omega \times \Omega)$ or $C(\Omega \times \Omega)$.*

Proof. From Lemma 5.4.1, 5.4.2 and 5.4.3 we obtain

$$\begin{aligned} ||S_\rho(w) - S_\rho(\tilde{w})|| &\leq ||K(E_\rho(T(w))) - K(E_\rho(T(\tilde{w})))|| \\ &\leq C_\alpha \rho c ||w - \tilde{w}||, \end{aligned} \tag{5.97}$$

where C_α and c are given constants and under particular conditions ρ can be made arbitrarily small. We can achieve

$$\tau := C_\alpha \cdot \rho \cdot c \tag{5.98}$$

to be smaller than 1, such that S_ρ is a contraction. This completes the proof. □

As a consequence of the above lemma, we know that in the case where we have sufficient observations (in the sense that the observation operator H is injective) and the observation error R is sufficiently small, the iteration process is a contraction. Thus, by the Banach fixed point theorem, the iteration will converge to a unique fixed point. In the case of true observations and when the temporal grid spacing is tending to zero, this limit of the sequence of fixed points will be the projection of the true kernel onto the space $N(V)^\perp$.

5.5 Summary

In this chapter we have developed and analysed an iterative method based on kernel reconstruction by inversion and data assimilation for the neural field equation. We started by introducing the idea of iterating the two processes of kernel reconstruction and state estimation for neural field equation as a combined method. Then we have analysed the algorithmic components using three different mappings:

- the transport map,
- the estimation operator,

- the kernel reconstruction operator.

We have studied the boundedness of the transport map and the boundedness and convergence of both the state estimation and kernel reconstruction. Then, we have analysed the complete process of iteration formulating a proof of convergence. A numerical example of these results will be shown in Section 6.4.

Chapter 6

Numerical Examples

The goal of this chapter is to demonstrate the arguments of this work in a visual way. We provide numerical examples to show the feasibility and effectiveness of our methods in each of their different components. The chapter is divided into four main sections. Each section is related to one part of our previous discussions. In first two Sections 6.1 and 6.2 we show the feasibility of inversion (solving the kernel reconstruction problem) and its sensitive analysis. The numerical examples are related to Chapter 3. Then, in Section 6.3 we present a feasibility study of 3D-VAR described in Chapter 4. Finally, in Section 6.4, we show examples for the iterative method and its convergence from the discussion of Chapter 5.

6.1 Feasibility of Kernel Reconstructions

Here we demonstrate the feasibility of the inverse method for the reconstruction of spatial kernels which model excitations of the spatio-temporal neural field activity. We study a) feasibility of the reconstruction in Section 6.1 and b) sensitivity with respect to variations in the input function u in Section 6.2. First, we consider a one-dimensional manifold embedded in a two-dimensional space, illustrating the method for a case with 10,000 degrees of freedom. Then, an example involving a two-dimensional spatial domain evaluates the method for an inverse problem with more than 200,000 degrees of freedom for the kernel estimation.

We first need to consider the role of boundaries in the neural field model equation (1.6) and its examples. For any distribution of neurons in space, some activity $u(r,t)$ depending on time t can be defined. Mutual influence in space is given by the integral in equation (1.4). In contrast to models based on partial differential equations, there is no direct boundary effect in these equations. However,

- if one uses a local kernel $w(r,r')$ with strong connectivity only in a neighbor-

hood of r, boundary effects for neurons close to the boundary of the domain will appear, since less neurons are included in a neighborhood there; whereas

- if the activity of neurons close to the boundary is close to zero, usually such boundary effects remain negligible.

We will study a setup which avoids boundary effects by the choice of an embedding of a one-dimensional manifold into two-dimensional space in our first example, where there are always the same number of neurons in a neighborhood of any neuron on the whole manifold. The second example instead limits boundary effects by using only small excitations close to the boundary in a two-dimensional neural patch.

Example 1. We start with a simple one-dimensional closed curve or manifold, respectively, embedded in a two-dimensional space. In particular, we study the dynamics of the activity field $u(r,t)$ on the boundary $\partial B_R \subset \mathbb{R}^2$ of a disk with radius R, as displayed in Figure 6.1. Here we consider that $v = 1$, and use a simple and smooth delay function for $r = (x, y)$ and $r' = (x', y')$ with $r, r' \in \partial B_R$ based on the embedding into $\mathbb{R}^2$ which defined by

$$D(r,r') := \tilde{D}(r,r') = |r - r'| = \sqrt{(x-x')^2 + (y-y')^2}, \quad r, r' \in \partial B_R. \tag{6.1}$$

This simple sandbox for testing our method hence can be considered as neurons growing on the boundary of a disk, but connecting directly through its interior. This is reminiscent of the thin exterior layer of gray matter containing neurons connecting through an interior bulk of white matter containing axons in the brain. However, we point out that this is a different setup from previous studies that superficially appear similar, where the spatial domain instead is a ring with periodic boundary conditions [51, 101].

We implemented the delay neural field equation in MATLAB® based on an Euler method for the time-evolution of the system with zero-order or first-order quadrature (rectangular rule or trapezoidal rule) for the integral parts of the integro-differential equation. For the purposes of studying the kernel reconstruction on a rather short temporal window this simple approach is completely sufficient and does not show any deficiencies compared to higher-order methods for the forward problem, as employed for example in [87, 25, 76].

We first solve the direct problem, i.e., calculate the time-evolution of the excitation field $u(r,t)$. As initial condition, we choose an exponential function

$$u(r,0) = e^{-\sigma |r - r_0|^2}, \quad r \in \partial B_R. \tag{6.2}$$

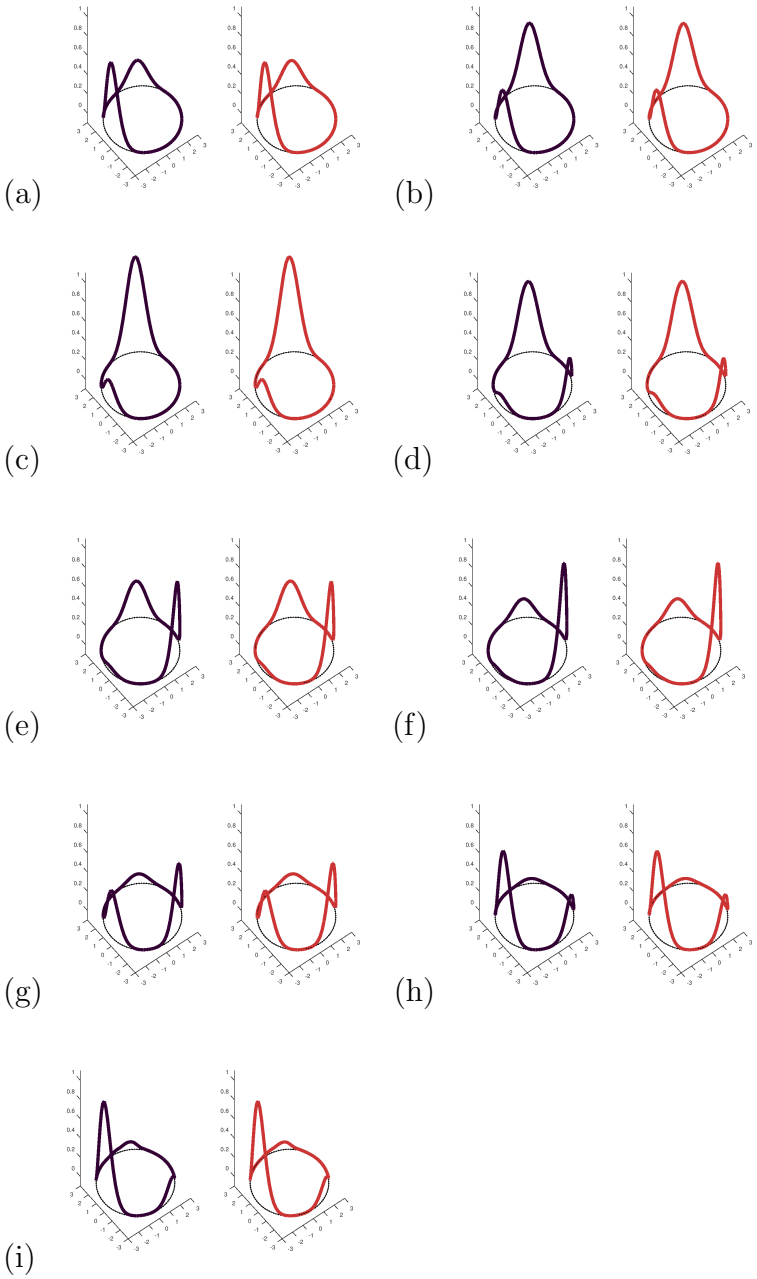
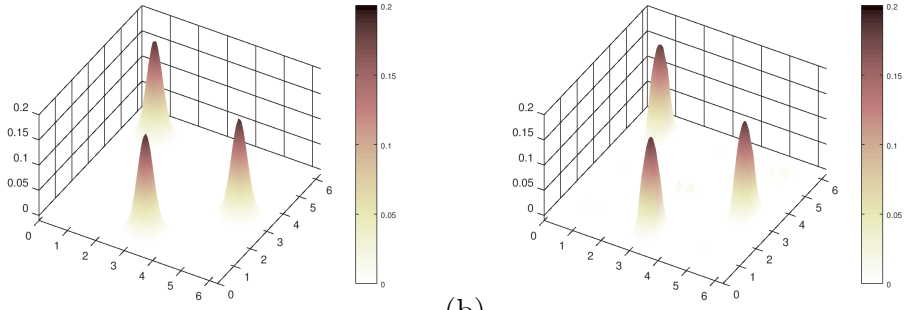


Figure 6.1: Time sequence of excitation of the one-dimensional delay neural field. The original field is shown in black, in red the dynamics based on the delay kernel reconstruction. One cycle of the oscillation is shown at time steps $1, 3, 6, 10, 13, 16, 19, 22, 25$, with a step size of $\Delta t = 0.2$, in panels (a) to (i).



(a) (b)

Figure 6.2: For the one-dimensional example the kernel can be visualized as a two-dimensional scalar function $w(r, r')$. We display (a) the original and (b) the reconstructed kernel of the one-dimensional delay dynamics shown in Figure 6.1.

We prescribe a neural kernel of the form

$$w(r, r') = c\left(e^{-\tau|r-r_1|^2}e^{-\tau|r'-r_0|^2} + e^{-\tau|r-r_2|^2}e^{-\tau|r'-r_1|^2} + e^{-\tau|r-r_0|^2}e^{-\tau|r'-r_2|^2}\right) \quad (6.3)$$

for $r, r' \in \partial B_R \subset \mathbb{R}^2$ with constants $c > 0$ and $\tau > 0$.

The full set of values used for our simulations are given in Table 6.1. This leads to delayed excitation of areas around three points r_0, r_1 and r_2 equally distributed on a circle, where, with some delay, the excitation field around r_0 will excite the field around r_1, the field around r_1 will excite the field around r_2 and the field around r_2 will excite the field around r_0 again. The function f is chosen to be sigmoidal as in Equation (1.3). We have generated a classical oscillator, as can be seen in the snapshots in Figure 6.1 (black curves). Its kernel is visualized in Figure 6.2(a).

r_0	$(\cos(\pi), \sin(\pi))$	σ	1.0
r_1	$(\cos(\pi/3), \sin(\pi/3))$	τ	1.0
r_2	$(\cos(-\pi/3), \sin(-\pi/3))$	c	3.0

Table 6.1: Parameter values for example 1. Simulations have been carried out with $N = 101$ equally distributed nodes on the circle, $N_t = 50$ time steps, and a time step size $\Delta t = 0.2$ for the inverse problem.

Next we reconstruct the kernel by the inverse problem technique from the so obtained temporal evolution of the excitation field $u(r, t)$ for some time window $t \in [0, T]$ according to equations (3.31) and (3.40). Given a discretized version of

$u(r,t)$ on nodes

$$r_\ell := \left(\cos\left(\frac{2\pi\cdot\ell}{N}\right), \sin\left(\frac{2\pi\cdot\ell}{N}\right)\right), \quad t_k = k\cdot\Delta t, \tag{6.4}$$

for $\ell = 0, ..., N$ and $k = 0, ..., N_t - 1$, we calculate ϕ and ψ according to equations (3.26) and (3.27) and then employ the regularization (3.40) via equation (3.43) to solve equation (3.36) for $r \in \partial B_R$. In Figure 6.2, we compare the original with the reconstructed kernel in the case where no additional noise is added, carried out with $\alpha = 0.01$ and find a very good agreement between both.

As a test, we employ the reconstructed kernel with the same initial condition to calculate a reconstructed neural field $u_{\text{rec}}(r,t)$ on $(r,t) \in \partial B_R \times [0,T]$. The original dynamics is shown in black in Figure 6.1, based on the kernel Eq. (6.3) visualized in Figure 6.2(a).

The reconstructed dynamics is shown in red in Figure 6.1, based on the reconstructed kernel visualized in Figure 6.2(b). A very good agreement between original and reconstructed solution is observed.

Example 2. As a second example, we study oscillating two-dimensional neural field activity. Here, the dimension of the state space is higher with $N = 21 \times 22 = 462$ spatial elements as shown in Figures. 6.3 and 6.4. Our approach is analogous to the one-dimensional example, but now with 213,444 degrees of freedom for the possible connectivity values. We first simulate the neural field dynamics based on equation (1.6) on a neural patch described by $\Omega := [a_1, b_1] \times [a_2, b_2] = [0, 6] \times [0, 6]$. Time slices of this dynamical evolution are displayed in Figure 6.3. The kernel has been chosen to be of a form similar to equation (6.3), but now with points r_0, r_1 and r_2 in the two-dimensional neural patch. This leads to an oscillating field in an area around these points r_j with $j = 0, 1, 2$. The activation function f is chosen to be sigmoidal again. The initial condition is a Gaussian excitation around the point r_0. For our simple tests, we again employ zero or first order quadrature and Euler's method to carry out the simulation.

The kernel $w(r, r')$ with $r, r' \in \Omega$ now lives on a subset $U := \Omega \times \Omega$ of a four-dimensional space, since Ω is a subset of a two-dimensional patch. Visualization of $w(r, r')$ can be carried out by either fixing r' and showing a two-dimensional surface plot, or by re-ordering r and r' into one-dimensional vectors, so that $w(r, r')$ can be displayed in full as a two-dimensional surface. The first approach is chosen in Figure 6.4(c), where the white star indicates r'. The second approach is shown in Figure 6.4(a). Next, we solve the *inverse delay neural problem* and reconstruct the kernel based on Equation (3.31) regularized as indicated by Equations. (3.40)

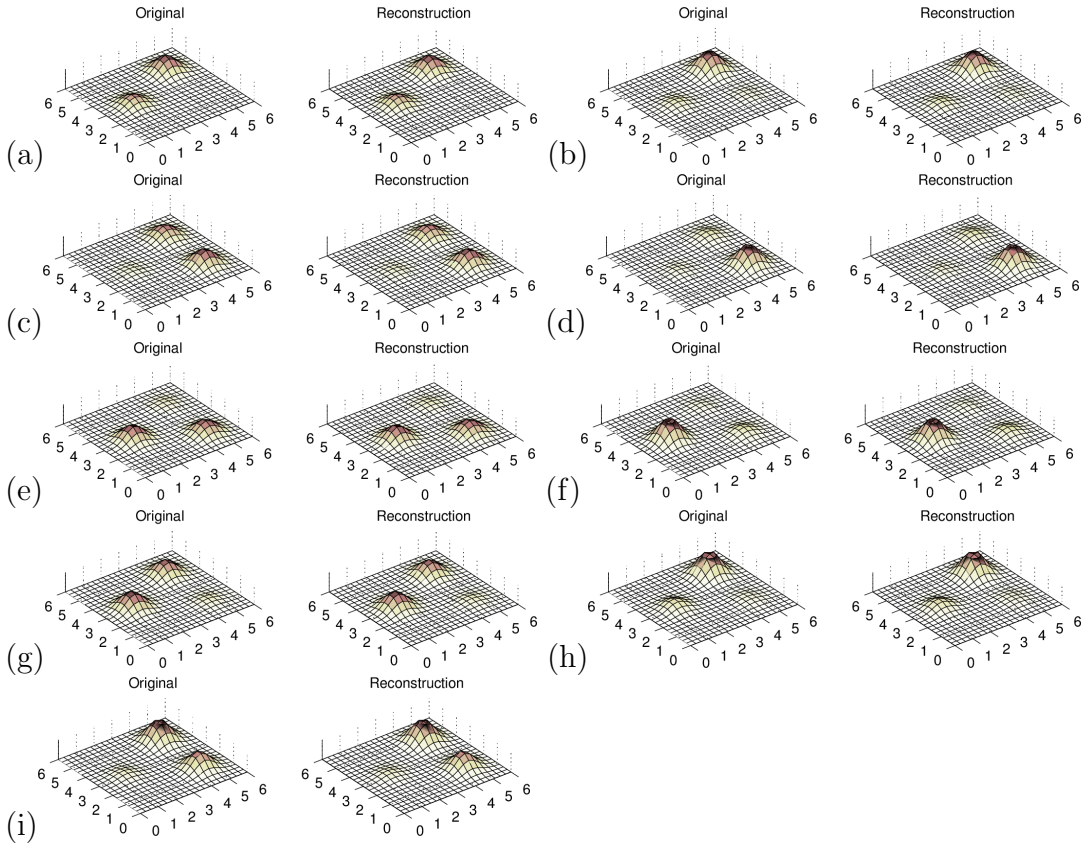


Figure 6.3: Selection of time slices for the two-dimensional delay neural field. We display time steps $3, 6, 9, 12, 15, 18, 21, 24, 27$ with $\Delta t = 0.2$ to show one and a half cycles of the oscillation in panels (a) to (i). Each panel shows the original on the left and simulation with the reconstructed kernel on the right.

r_0	$(1.5, 3.0)$	σ	2.0
r_1	$(4.5, 4.5)$	τ	1.0
r_2	$(4.5, 1.5)$	c	2.1

Table 6.2: Parameter values for Example 2. Simulations have been carried out with $N = 21 \times 22 = 462$ nodes, $N_t = 30$ time steps with time step size $\Delta t = 0.2$ for the inverse problem. The kernel estimation problem has 213,444 degrees of freedom.

and (3.43). Again, this is carried out by calculation of ϕ and ψ first according to Equations (3.26) and (3.27), then solving Equation (3.36) by regularization via Equation (3.40) with the regularization parameter chosen as $\alpha = 0.1$. This choice leads to a reasonable stability of the reconstructions combined with high reconstruction quality, and has been chosen by trial and error. That choice is made to reach a sufficient results. There are different well-known methods to choose the regularization parameter, such as Morozov's discrepancy or the L-curve approach, see e.g. section 3.1.6 of [76]. Since this is not the focus of this work, we have decided to stay with a simple approach.

Figure 6.4(c) and (d) displays the *original* and the *reconstructed* kernel column, which represents the impact of the location at the black star to all other spatial locations of the neural patch. The result as displayed in (b) shows that the regularized reconstruction of the delay neural kernel is not perfect. However, it is working well if the field activity reaches specific parts of the neural environment. Otherwise the reconstruction is just zero due to missing input for the reconstruction equations and the regularization chosen here. The regularization penalizes the distance to the zero kernel function. Therefore, the results clearly demonstrate the feasibility of the method. To summarize our results, we see that those examples provide a great evidence of the effectiveness and applicability of the kernel reconstruction method.

6.2 Sensitivity with Respect to Functional Input

In this section we will carry out a numerical sensitivity study of our first example to explore the dependence of the kernel reconstructions on the input function u. It complements our sensitivity analysis of Section 3.5.

We study the stability of the reconstruction when we add some random error to the measured signal $u(r, t)$ displayed in Figure 6.1. We remark that we need measurements of our signal which are differentiable with respect to time, since

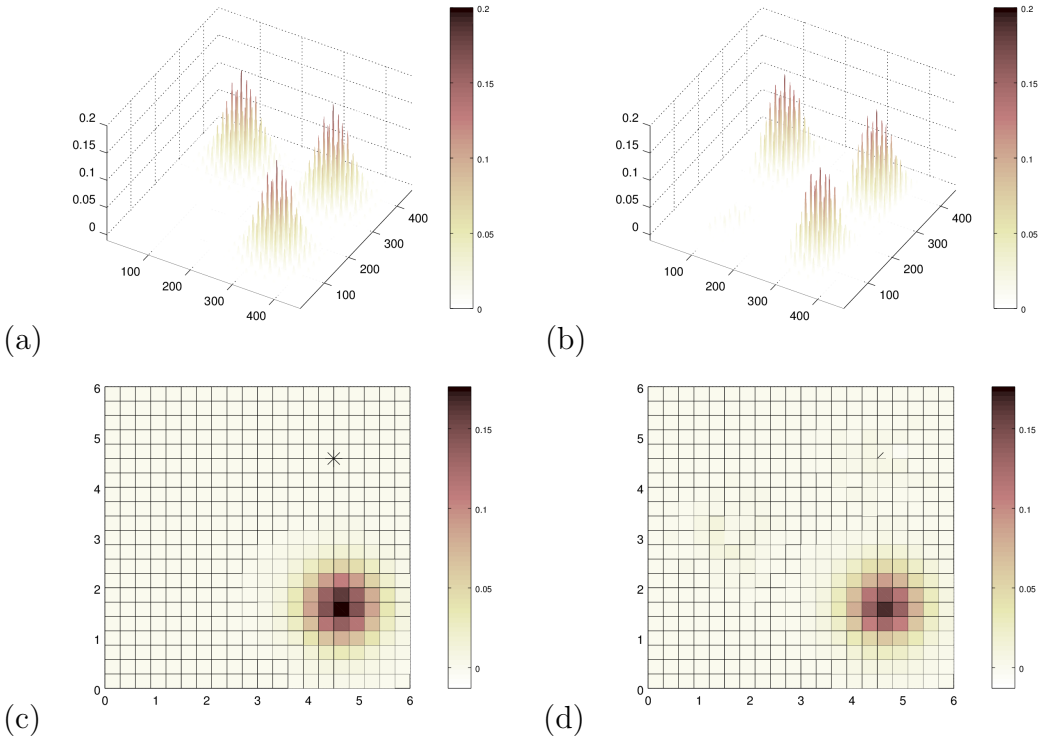


Figure 6.4: We display (a) the original and (b) the reconstructed kernel of the two-dimensional neural delay dynamics shown in Figure 6.3. The images (c) and (d) show a column of the original and reconstructed kernel, visualizing the connection from the point indicated by the black star to the rest of the neural patch.

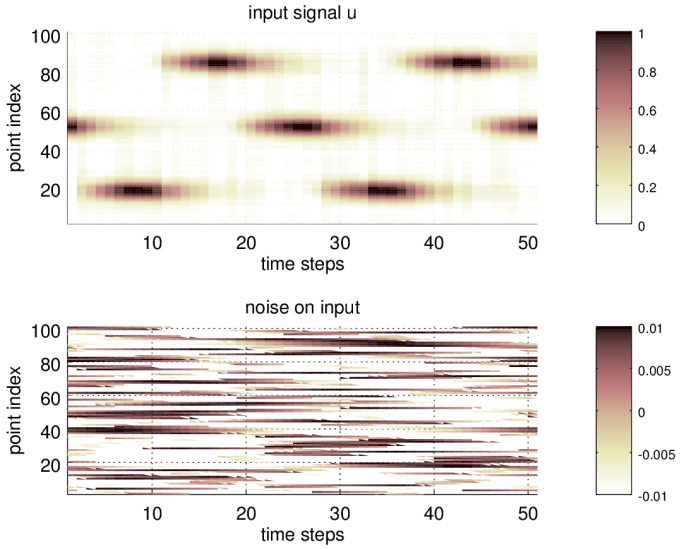
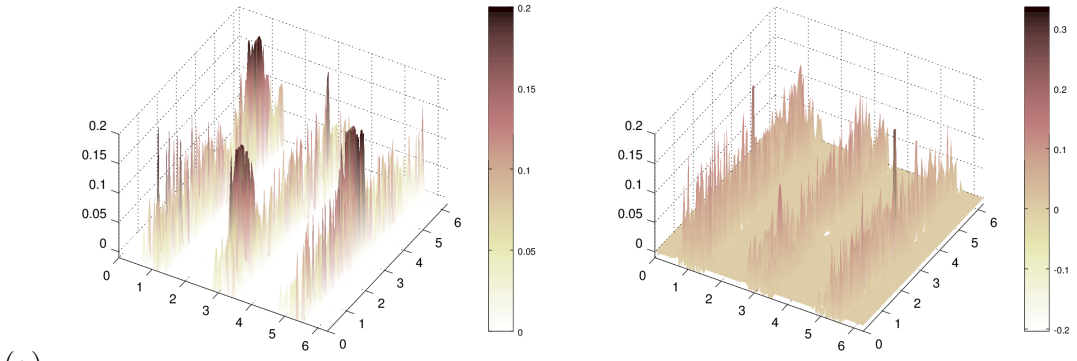


Figure 6.5: In the upper image we display the input signal $u(r,t)$ independence of the point index of the discretized vector r and the temporal evolution $t \in [0,T]$. The lower image shows the measurement error which has been added to the signal before a reconstruction has been carried out.

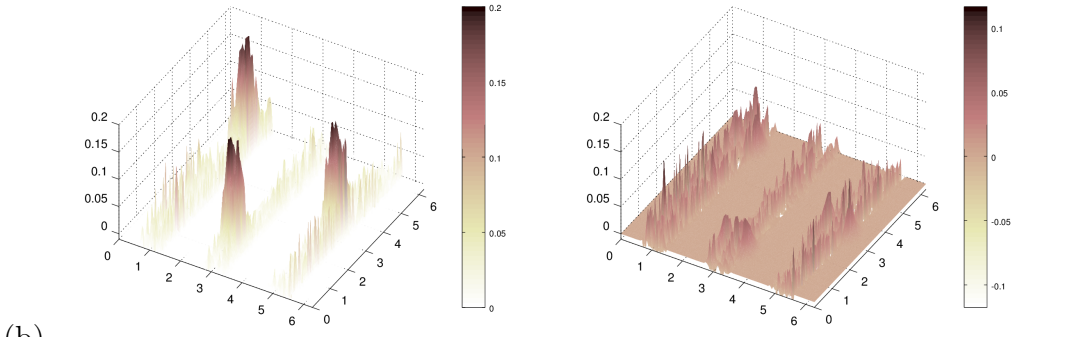
the calculation of ψ in Equation (3.27) includes the temporal derivative of the signal. In practical situations, this would be achieved by a temporal smoothing of the signal. Here, for testing the sensitivity we have added a random shift of a temporally smooth signal in each of the analysis points. The amplitude of the signal is given by $\epsilon = 0.01$, which corresponds to noise of 1% added to the measured temporal signal, compare Figure 6.5.

Now, we study reconstructions with different regularization parameters α, where larger α means we regularize in a stronger way, damping the error which comes from the measurement error. Figure 6.6 displays three different choices of α, where $\alpha = 1$ leads to reasonable reconstructions, $\alpha = 0.1$ shows kernel reconstruction still disturbed by noise, and $\alpha = 0.01$ does not lead to satisfactory reconstructions at all.

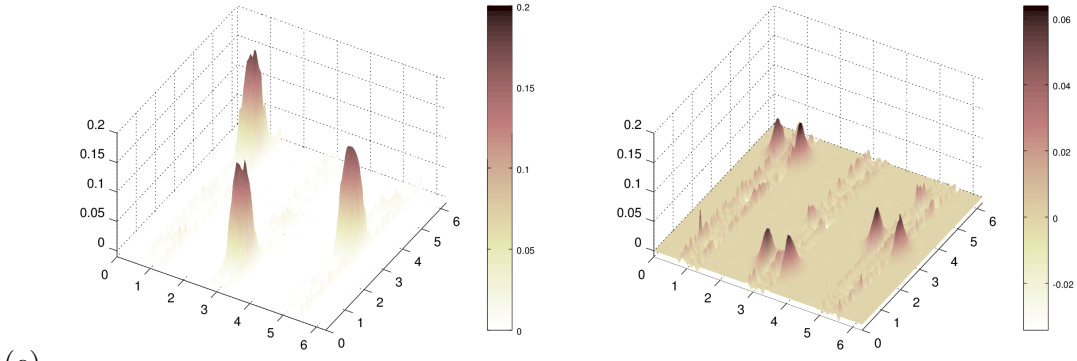
According to Theorem 3.5.1 we have continuity of $u \mapsto w$, such that if we lower the error ϵ for fixed α, we need to have convergence to the reconstructed kernel in the case of no data error. Indeed, we obtain a figure similar to Figure 6.6 when we lower the error parameter ϵ from $\epsilon = 0.01$ to $\epsilon = 0.005$ and $\epsilon = 0.001$, leading to the reconstruction displayed in Figure 6.2(b) for $\epsilon = 0$.



(a)



(b)



(c)

Figure 6.6: We show reconstruction kernels and the reconstruction error for 1% noise shown in Figure 6.5 with regularization parameters $\alpha = 0.01$ in (a), $\alpha = 0.1$ in (b) and $\alpha = 1$ in (c). A sufficient reconstruction quality is achieved with $\alpha = 1$.

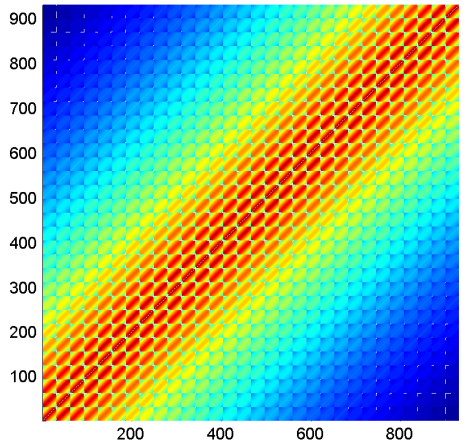


Figure 6.7: A visualisation of the background error covariance matrix

6.3 3D-Var for Neural Field Equations

We now come to our second task, i.e. to apply *data assimilation*. In this section, we provide a numerical example for the state estimation of a neural field following our theory of Section 4.2.

We first start with introducing the *background error covariance matrix* in Figure 6.7. Here, we have chosen it to be a *Gaussian* distribution in space in the sense that each fields at a point r has a covariance with respect to the field at a point r' defined by

$$B(r, r') := ce^{-\frac{1}{2}\frac{|r-r'|^2}{\sigma^2}}, \quad r, r' \in \mathbb{R}^2. \tag{6.5}$$

In the two-dimensional case, when we employ a regular grid with a total of n points over some area $[a_1, b_1] \times [a_2, b_2]$ and sort the points line by line into an array, the visual display of the corresponding correlation matrix $B \in \mathbb{R}^{n\times n}$ is shown in Figure 6.7.

Then, we provide a visualization of the state estimation processes. We simulate data based on some traveling pulse. The figure shows in four images the components of *data assimilation* in Figure 6.8.

6.4 Iteration with convergence Examples

The goal of this section is visualizing the results of our iterative method. We provide the numerical example to demonstrate the iterative method of inversion

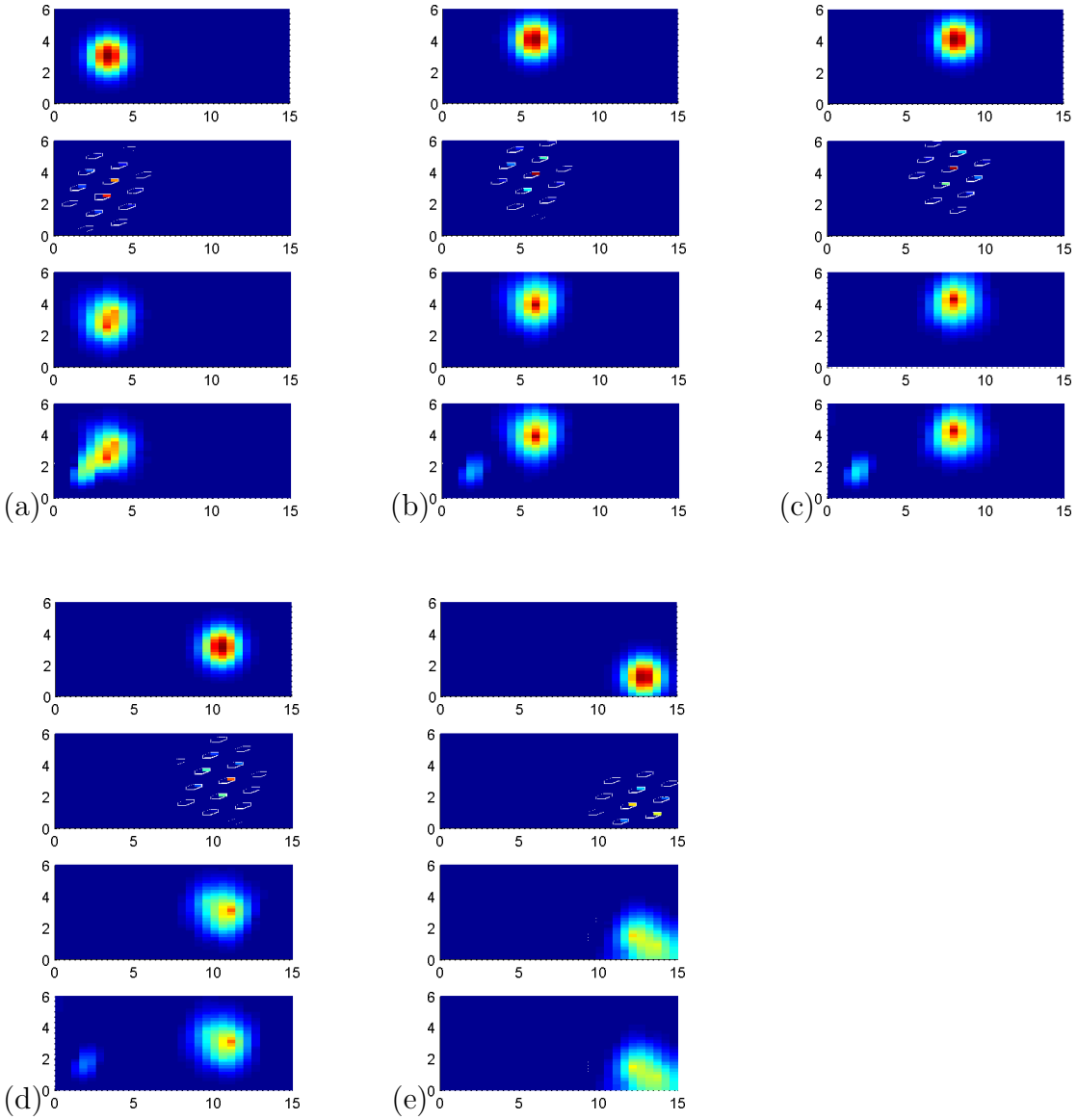


Figure 6.8: An illustration of the components of data assimilation of neural field at five different time steps. The figure consists of 4 images. They are from top to bottom the background $u^{(b)}$, the observations y, the 3D-VAR estimation $u^{(a)}$ and an estimate without background.

and data assimilation for neural field equation and its convergence.

In Chapter 5 we have shown that under suitable conditions the transport map is bounded, the estimator is convergent and the reconstruction is convergent and bounded. As a consequence, the numerical examples for the iteration should show a convergent behavior in the sense that

- the distance between the true kernel and true excitation and the iterates for each of these should become smaller during an initial set of iterations.
- When using noisy data, we expect the error to start to be dominant after a set of initial iterations, such that
- in the case of noisy data we should stop iterating when the error starts to dominate.

First we consider a one-dimensional case, embedded in a two-dimensional space. We apply the iterative method carrying out five iterations and visualize the results of third iteration in Figure 6.9. This choice depends on the results where we got convergence. It can be seen as a stopping role in the case of incomplete observations.

6.5 Summary

We have worked out numerical examples on one- and two-dimensional spatial domains for

- *kernel reconstruction* as a typical inverse problem for delayed neural field equations,
- *variational data assimilation* applied to neural activity functions, when measurements of neural activity are given at some points of the neural tissue or integrated values as they are typically measured based on standard medical electrodes,
- the *iterative method* of iterating state estimation and kernel reconstruction to carry out kernel reconstruction in a realistic environment.

These examples show that the regularized reconstruction of the delay neural kernel is practically feasible. We study the numerical sensitivity, by adding random noise of size ϵ (testing 1%, 0.1% and 0.01%) and studying the regularized reconstruction with different regularization parameters.

Then, we provide an illustration of applying the 3D-VAR method for neural dynamics based on traveling pulses and considering a Gaussian distribution of the background error covariance. Our example shows four components of data assimilation process.

Finally, we demonstrate the iteration of inversion and state estimation approach and its convergence. This example is comparing the estimated field resulted from applying the iteration with the original.

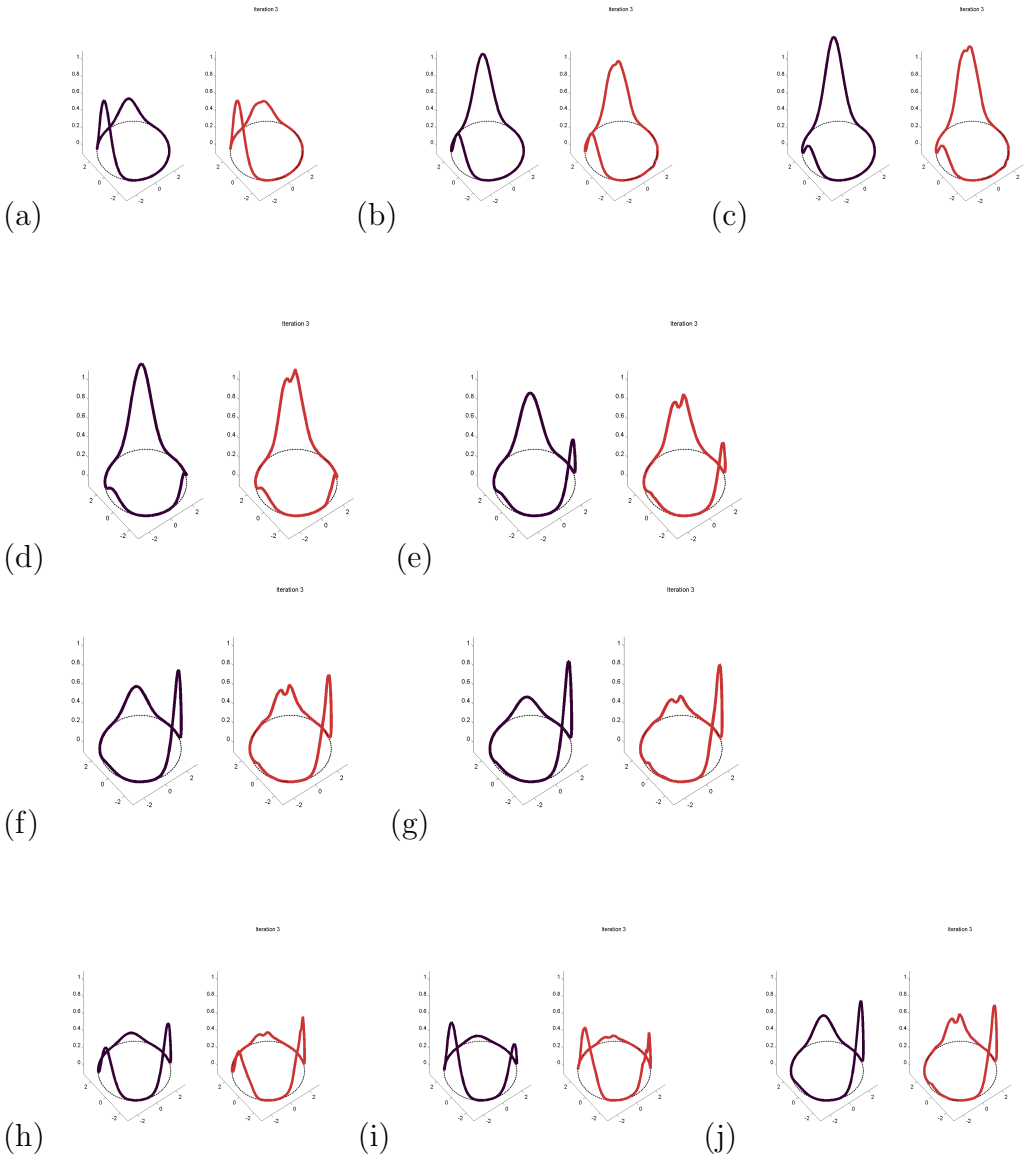


Figure 6.9: A comparison of the results of third iteration of different time steps when applying the iterative method on the one-dimensional delayed neural field. One cycle of the oscillation is shown at the third iteration in sequence of different time steps $1, 4, 6, 9, 11, 14, 16, 19, 21, 24$, with a step size of $\Delta t = 0.2$, in panels (a) to (j). The choice of this iteration is due to when the convergence is obtained.

Chapter 7

Conclusions and Outlook

The iteration of inversion and data assimilation is a combined work aims to merge the effectiveness of the both famous techniques. Data assimilation is the method of obtaining best estimations of the system state depending on knowledge of the model dynamics, observations and error statistics. On the other hand, inversion is the method of finding the cause of the activity.

The objective of this thesis is to build a theory from three different aspects. First, we discuss the delayed neural field equation and its inversion. Second, we introduce the application of data assimilation in neural field equation. Finally, we study an iterative approach for the neural field equation (1.4). This research is motivated by the challenge of complexity of neural systems and several promising recent studies on this topic.

In the two following sections we summarize the aim and results of this work and the what future work with high potential could be built on our investigation.

7.1 Conclusions

The idea is divided to various tasks starting with studying the solvability of the delayed neural field equation, where we provide an easy to follow and non-trivial proof using tools of functional analysis. Then, we start our study by applying inverse problems techniques to reconstruct its kernel in Chapter 3. This work is following the theory that has been developed in [85, 82] for non-delay case. Due to the ill-posedness of the inverse problem, we need to approximate the solution using a regularization method. We chose to use Tikhonov regularization as a simple and very popular approach.

The kernel reconstruction is built on a sequence of known system states u, although, in reality, this knowledge is not completely available. We actually have

some measurements or experimental results. This leads to the need of applying data assimilation. In Chapter 4, an introduction of the theory of data assimilation is provided and the application of 3D-Var is studied to estimate the state.

The final task is the combination of both techniques in one iterative approach and studying its convergence. We formulate the method using three sequential operators and study the theory of its convergence. This theory has been discussed in detail in Chapter 5.

In Chapter 6 we support our discussion by numerical examples. We show the feasibility of the inversion, the sensitivity analysis, application of 3D-Var and the iteration approach.

We conclude that we have made a significant step forward to develop an approach for practical kernel reconstruction by combination of data assimilation and inverse problems techniques. Further, we have suggested an innovative iterative approach and studied its convergence. For all three parts of the thesis, i.e. kernel reconstruction, state estimation and iteration of the approaches, we have shown feasibility by numerical examples.

7.2 Future Investigation

The result of this work is promising for future research from many different points of view. First, the kernel reconstruction approach is opening the doors to develop the theory in other disciplines such as *machine learning*. The idea of reconstructing the kernel is not only applicable to neural field equations, but also to different dynamical systems where connections are employed. That means this method could work, for instance, to reconstruct the kernel for machine learning problems or radiative transfer equation with scattering.

Reconstructing of the firing rate function is another problem could be considered. However, it is more difficult and challenging due to the non- linearity of its inverse problem, since the neural field equation contains the product of the kernel w with $f(u(\cdot))$, such that any reconstruction of both w and f will be a non-linear inverse problem.

The construction of the kernel could be also considered in more biologically realistic distributions of velocities, which is more challenging as well, since the distribution of propagation speed would be an unknown parameter field on top of other unknown parts of the system.

Our theory could also be applied to neural field models that separate out excitation and inhibition such as the activity-based Wilson and Cowan model as well as for spiking neural networks. This could, for example, be realized by different kernel functions for exciting and inhibiting influence. The resulting kernel matrix $(w_+, -w_-)^T$ for exciting and inhibiting neural signalling could, e.g. be reconstructed using similar techniques as applied in this work, under the condition that the sensitivity functions f_+ and f_- in both cases lead to linear independent reconstruction matrices, when solving the kernel reconstruction problem for

$$\dot{u}\,(x) + u(x) = \int_\Omega w_+(x,y) f_+(u(y))\,dy - \int_\Omega w_-(x,y) f_-(u(y))\,dy, \quad x \in \Omega.$$

We have used Tikhonov regularization for inverting the kernel equations. There are, of course, many further *regularization methods*, which are not explored yet in the area of inverse neural field theory.

Secondly, we provide a theory for using 3D-Var for a neural field equation. This theory should be improved in future work for (delayed) neural field equation. We note that with 3D-VAR we apply a particular data assimilation technique to estimate the neural state, although there are many other data assimilation methods which could be used for state estimation. 3D-VAR employs a fixed covariance matrix. The Ensemble Kalman Filter could be used which employs an ensemble of forcasted states to calculate a dynamic covariance matrix. 3D-Var and the Kalman Filter are both based on second-order statistics, using a mean and covariance matrix with an exact update for Gaussian distributions. For higher order statistics particle filters might be useful. It is important to note that the field is young and not explored widely yet.

Finally, our theoretical study of the iterative estimation and inversion method contains many interesting elements which need further investigation. What will happen in the case of incomplete data? What is the influence of further knowledge about properties of the neural tissue on the iteration process? What would be an appropriate stopping rule for the iterations?

Clearly, we have studied very simple systems to show the feasibility of the methods developed in this work. It is an open question how the techniques could be improved to overcome any additional exotic dynamics. We remark, however, that these challenges are common to most chaotic dynamical systems and data assimilation methodology applied to such systems. The community is intensely working on, for example, particle filters, which treat the strongly non-Gaussian distributions which arise from chaotic dynamics.

Bibliography

[1] H. D. I. Abarbanel. *Predicting the Future: Completing Models of Observed Complex Systems*. Springer, 2013.

[2] J. Alswaihli, R. Potthast, I. Bojak, D. Saddy, and A. Hutt. Kernel reconstruction for delayed neural field equations. *Journal of Mathematical Neuroscience*, 8:3, 2018.

[3] Z. An, D. Rey, J. Ye, and H. D. I. Abarbanel. Estimating the state of a geophysical system with sparse observations: time delay methods to achieve accurate initial states for prediction. *Nonlin. Processes Geophys.*, 24(1):9–22, 2017.

[4] J. A. Anderson and E. Rosenfeld. *Neurocomputing, Foundations of Research*. Massachusetts Institute of Technology, England, 1988.

[5] P. Argoul. Overview of inverse problems. *DEA. Parameter Identification in Civil Engineering, Ecole Nationale des Ponts et Chaussées*, page 13, 2012.

[6] F. M. Atay and A. Hutt. Stability and bifurcations in neural fields with finite propagation speed and general connectivity. *SIAM J. APPL. MATH.*, 65, No. 2:644–666, 2005.

[7] G. E. Backus and J. F. Gilbert. Numerical applications of a formalism for geophysical inverse problems. *Geophysical Journal of the Royal Astronomical Society*, 13(1-3):247–276, 1967.

[8] G. E. Backus and J. F. Gilbert. The resolving power of gross earth data. *Geophysical Journal International*, 16(2):169–205, 1968.

[9] S. Baillet. *Forward and Inverse Problems of MEG/EEG*, pages 1–8. Springer New York, 2014.

[10] R. N. Bannister. A review of operational methods of variational and ensemble-variational data assimilation. *Q. J. R. Meteorol. Soc.*, 2017.

[11] D. M. Barker, W. Huang, Y.-R. Guo, and A. Bourgeois. A three-dimensional variational (3dvar) data assimilation system for use with mm5. *NCAR Tech. Note. NCAR/TN-453 + STR, 68 pp*, 2003.

[12] P. beim Graben and R. Potthast. Inverse problems in dynamic cognitive modeling. *Chaos - An Interdisciplinary Journal of nonlinear Science*, 2009.

[13] E. Blayo, E. Cosme, M. Nodet, and A. Vidard. Introduction to data assimilation, 2011.

[14] C. G. Boeree. The neuron. *General Psychology, 1-6*, 2008.

[15] I. Bojak and D. T. Liley. Axonal velocity distributions in neural field equations. *PLoS Comut. Biol.*, 6(1): e1000653, 2010.

[16] F. Bouttier and P. Courtier. Data assimilation concepts and methods. *Training Course notes of ECMWF*, 1999.

[17] P. C. Bressloff and S. Coombes. Physics of the extended neuron. *International Journal of Modern Physics B*, 11:2343–2392, 1997.

[18] P. C. Bressloff and Z. P. Kilpatrick. Two-dimensional bumps in piecewise smooth neural fields with synaptic depression. *SIAM J. APPL. MATH*, 71:379–408, 2011.

[19] S. A. Campbell. *Time Delays in Neural Systems*, pages 65–90. Springer., 2007.

[20] A. Carrassi, M. Bocquet, L. Bertino, and G. Evensen. Data assimilation in the geosciences: An overview of methods, issues and perspectives. *Wiley interdisciplinary reviews*, 2018.

[21] S. Cavallari, S. Panzeri, and A. Mazzoni. Comparison of the dynamics of neural interactions between current-based and conductance-based integrate-and-fire recurrent netwworks. *Frontiers in Neural Circuits.*, 8, 2014.

[22] W. Cheney. *Analysis for Applied Mathematics*. Springer-Verlag New York, 2001.

[23] S. Coombes. Waves, bumps, and patterns in neural field theories. *Biological Cybernetics*, 93:91–108, 2005.

[24] S. Coombes. Large-scale neural dynamics: Simple and complex. *NeuroImage*, 52:731–739, 2010.

[25] S. Coombes, P. beim Graben, and R. Potthast. *Tutorial on neural field theory*, pages 1–43. Springer, 2014.

[26] S. Coombes, P. beim Graben, R. Potthast, and J. Wright. *Neural Fields: Theory and Applications*. Springer, 2014.

[27] S. Coombes and C. Laing. Delays in activity-based neural networks. *Phil. Trans. R. Soc.*, 367:1117–1129, 2009.

[28] S. Coombes and H. Schmidt. Neural fields with sigmoidal firing rates: Approximate solutions. *Discrete and Continuous Dynamical Systems -A*, 28(4):1369–1379, 2010.

[29] S. Coombes and J. R. Terry. The dynamics of neurological disease: Integrating computational, experimental and clinical neuroscience. *European Journal of Neuroscience*, 36:2118–2120, 2012.

[30] S. Coombes, N. Venkov, L. Shiau, I. Bojak, D. Liley, and C. Laing. Modeling electrocortical activity through improved local approximations of integral neural field equations. *Phys. Rev. E*, 76:051901-8, 2007.

[31] R. Daley. *Atmospheric Data Analysis*. Cambridge University Press, 1991.

[32] G. Deco and V. K. Jirsa. Ongoing cortical activity at rest: Critically, multistability, and ghost attractors. *Journal of Neuroscience*, 32(10):3366–3375, 2012.

[33] G. Deco, V. K. Jirsa, P. A. Robinson, M. Breakspear, and K. Friston. The dynamic brain: From spiking neurons to neural masses and cortical fields. *PLOS Computayional Biology*, 4(8), 2008.

[34] O. Diekmann. *Delay Equations: Functional-, Complex-, and Nonlinear Analysis.* Springer- Verlag Ne York, Inc., 1995.

[35] K. Dijkstra, S. A. van Gils, and S. G. Janssens. Pitchfork-hopf bifurcations in 1d neural field models with transmission delays. *Physica D: Nonlinear Phenomena*, 297:88–101, 2015.

[36] G. T. Einevoll. Mathematical modeling of neural activity. *Dynamics of Complex Interconnected Systems: Networks and Bioprocesses*, pages 127–145, 2006.

[37] H. W. Engl, M. Hankle, and A. Neubauer. *Regularization of Inverse Problems*. Mathematics and Its Applications. Springer Netherlands, 2000.

[38] B. Ermentrout. Neural networks as spatio-temporal pattern-forming systems. *Reports on Progress in Physics*, 61:353–430, 1998.

[39] Z. Fang, A. S. Wong, K. Hao, A. J. A. Ty, and H. D. I. Abarbanel. Precision annealing monte carlo methods for statistical data assimilation and machine learning. *Nonlin. Processes Geophys.*, 24(1):9–22, 2017.

[40] G. Faye and O. Faugeras. Some theoretical and numerical results for delayed neural field equations. *Physica D*, 239:561–578, 2010.

[41] M. Fisher. Background error covariance modelling, 2003.

[42] S. J. Fletcher. *Data Assimilation for Geoscience: From Theory to Application.* Elsevier Inc., USA, 2017.

[43] D. R. Freestone, P. J. Karoly, D. Nešić, P. Aram, M. J. Cook, and D. B. Grayden. Estimation of effective connectivity via data-driven neural modeling. *Frontiers in Neuroscience*, 8, 2014.

[44] M. A. Freitag and R. W. E. Potthast. Synergy of inverse problems and data assimilation techniques. *Radon Series on Computational and applied Mathematics*, pages 1–54, 2013.

[45] A. Ghosh, Y. Rho, A. R. McIntosh, R. K$'$otter, and V. K. Jirsa. Cortical network dynamics with time delays reveals functional connictivity in the resting brain. *Cogn. Neurodyn*, 2(2):115–120, 2008.

[46] A. Ghosh, Y. Rho, A. R. McIntosh, R. K$'$otter, and V. K. Jirsa. Noise during rest enables the exploration of the brain's dynamic repertoire. *PLOS Comput. Biol.*, 4(10), 2008.

[47] S. Gils, S. G. Janssens, Y. Kuznetsov, and S. Visser. On local bifurcations in neural field models with transmission delays. *Journal of Mathematical Biology*, 66:837–887, 2013.

[48] C. W. Groetsch. *Inverse problems in the mathematical sciences.* Theory and Practice of Applied Geophysics Series. Vieweg, 1993.

[49] A. Hutt. Generalization of the reaction-diffusion, swift-hohenberg, and kuramoto-sivashinsky equations and effects of finite propagation speeds. *Phys. Rev. E*, 75:026214, 2007.

[50] A. Hutt and F. M. Atay. Analysis of nonlocal neural fields for both general and gamma-distributed connectivities. *Physica D*, 203:30–54, 2005.

[51] A. Hutt, M. Bestehorn, and T. Wennekers. Pattern formation in intracortical neural fields. *Network: Comput. Neural Syst.*, 14:351-368, 2003.

[52] A. Hutt and L. Buhry. Study of gabaergic extra-synaptic tonic inhibition in single neurons and neural populations by traversing neural scales: Application to propofol-induced anaesthesia. *J. Comput. Neurosci.*, 37(3):417-437, 2014.

[53] S. ichi Amari. Homogeneous nets of neuron-like elements. *Biolobical Cybernetics*, 17:211–220, 1975.

[54] S. ichi Amari. Dynamics of patterns formation in lateral-inhibition type neural fields. *Biolobical Cybernetics*, 27:77–87, 1977.

[55] S. I. Kabanikhin. Definitions and examples of inverse and ill-posed problems. *J. Inv. Ill-posed Problems*, 16:317–357, 2008.

[56] E. Kalnay. *Atmospheric Modeling, Data Assimilation and Predictability.* Cambridge University Press, 2003.

[57] B. Kaltenbacher, A. Neubauer, and O. Scherzer. *Iterative Regularization Methods for Nonlinear Ill-Posed Problems.* Radon Series on Computational and Applied Mathematics. De Gruyter, 2008.

[58] E. R. Kandel, J. H. Schwartz, and T. M. Jessell. *Principles of Neural Science.* Appleton & Lange, USA, 1991.

[59] A. Kirsch. *An Introduction to The Mathematical Theory of Inverse Problems.* Applied Mathematical Sciences. Springer-Verlag New York, 2011.

[60] A. Kolossa. *Computational Modeling of Neural Activities for Statistical Inference.* Springer, Switzerland, 2016.

[61] R. Kress. *Numerical Analysis.* Springer-Verlag New York, Inc., 1998.

[62] R. Kress. *Linear Integral Equations*, volume 82 of *Applied Mathematical Sciences.* Springer New York, 1999.

[63] W. A. Lahoz and P. Schneider. Data assimilation: making sense of earth observation. *Frontiers in Environmental Science*, 2:16, 2014.

[64] C. R. Laing and S. Coombes. The importance of different timings of excitatory and inhibitory pathways in neural field models. *Computation in Neural Systems*, 17:151–172, 2006.

[65] L. Lapidus and G. Pinder. *Numerical Solution of Partial Differential Equations in Science and Engineering*. Wiely-Interscience Publication, 1999.

[66] K. Law, A. Steuart, and K. Zygalakis. *Data Assimilation: A Mathematical Introduction*. Springer, 2015.

[67] P. Lima and E. Buckwar. Numerical investigation of the two-dimentional neural field equation with delay. In *2015 Second International Conference on Mathematics and Computers in Science and in Industry (MCSI)*, pages 131–137, Sliema, 2015.

[68] P. M. Lima and E. Buckwar. Numerical solution of the neural field equation in the two-dimentional case. *SIAM J Sci Comput*, 37:963-979), 2015.

[69] N. Lingala, N. S. Namachchivaya, and H. C. Yeong. Data assimilation in multiscale complex systems. *2015 8th International Workshop on the Analysis of Multitemporal Remote Sensing Images (Multi-Temp)*, pages 1–4, 2015.

[70] J. T. Manter, A. J. Gatz, and R. G. Clark. *Manter and Gatz's Essentials of Clinical Neuroanatomy and Neurophysiology*. F. A. Davis Company, USA, original from the University of California, 1975.

[71] C. D. Meliza, M. Kostuk, H. Huang, A. Nogaret, D. Margoliash, and H. D. I. Abarbanel. Estimating parameters and predicting membrane voltages with conductance-based neuron models. *Biological Cybernetics*, 2014.

[72] C. M. Michel and D. Brunet. Eeg source imaging: A practical review of the analysis step. *Frontiers in Neurology*, 2019.

[73] A. Miller, D. Li, J. Platt, A. Daou, D. Margoliash, and H. D. I. Abarbanel. Statistical data assimilation: Formulation and examples from neurobiology. *Frontiers in Applied Mathematics and Statistics*, 26, 2018.

[74] M. J. Moye and C. O. Diekman. Data assimilation methods for neuronal state and parameter estimation. *The Journal of Mathematical Neuroscience*, 8, 2018.

[75] T. T. Nakagawa, M. Woolrich, H. Luckhoo, M. Joensson, H. Mohseni, M. L. Kringelbach, V. Jirsa, and G. Deco. How delays matter in an oscillatory whole-brain spiking-neuron network model for meg alpha-rhythms at rest. *NeuroImage*, 87:383–349, 2014.

[76] G. Nakamura and R. Potthast. *Inverse Modeling: An Introduction to The Theory and Methods of Inverse Problems and Data Assimilation*. IOP Institute of Physics, Bristol, UK, 2015.

[77] A. Nogaret, C. D. Meliza, D. Margoliash, and H. D. I. Abarbanel. Automatic construction of predictive neuron models through large scale assimilation of electrophysiological data. *Scientific Reports 6, www.nature.com/scientificreport*, 2016.

[78] P. L. Nunez. The brain wave equation: A model for the eeg. *Mathematical Biosciences 21, 279-297*, 1974.

[79] P. L. Nunez. *Neocortical Dynamics and Human EEG Rhythms*. Oxford University Press, 1995.

[80] E. R. Oby, M. D. Golub, J. A. Hennig, A. D. Degenhart, E. C. Tyler-Kabara, B. M. Yu, S. M. Chase, and A. P. Batista. New neural activity patterns emerge with long-term learning. *PNAS*, 116 (30):15210–15215, 2019.

[81] A. Oleynik, A. Ponosov, and J. Wyller. Iterative schemes for bump solutions in a neural field model. *Differ. Equ. Dyn. Syst.*, 23:79–98, 2015.

[82] R. Potthast. Inverse problems in neural population models. *Encyclopedia of Computational Neuroscience*, pages 1453–1456, 2015.

[83] R. Potthast. Introduction to functional analysis and inverse problems. *Lecture notes*, Spring-term 2007.

[84] R. Potthast and P. beim Graben. Dimensional reduction for the inverse problem of neural field theory. *Frontiers in Computational Neuroscience*, 3,17, 2009.

[85] R. Potthast and P. beim Graben. Existence and properties of solutions for neural field equations. *Mathematical Methods in the Applied Science*, 33(8):935–949, 2009.

[86] R. Potthast and P. beim Graben. Inverse problems in neural field theory. *SIAM Journal on Applied Dynamical Systems*, 8:1405 –1433, 2009.

[87] R. Potthast and P. Graben. Inverse problems in neural field theory. *SIAM Journal on Applied Dynamical Systems*, 8(4):1405–1433, 2009.

[88] J. Rankin, D. Avitabil, J. Baladron, G. Faye, and D. J. Lloyd. Continuation of localised coherent structures in nonlocal neural field equations. *SIAM Journal on Scientific Computing*, 36:1:70–93, 2014.

[89] D. Rey, M. Eldridge, M. Kostuk, H. D. I. Abarbanel, J. Schumann-Bischoff, and U. Parlitz. Accurate state and parameter estimation in nonlinear systems with sparse observations. *Physics Letters A*, 2014.

[90] M. E. Rule, D. Schnoerr, M. H. Henning, and G. Sanguinetti. Neural field models for latent state inference: Application to large-scale neuronal recording. *PLOS Computayional Biology*, 15(11), 2019.

[91] P. C. Sabatier. Past and future of inverse problems. *Journal of Mathematical Physics*, 41(6), 2000.

[92] L. Spek, Y. A. Kuznetsov, and S. A. V. Gils. Neural field models with transmission delays and diffusion. *arXiv 1912.09762[math.DS]*, 2019.

[93] A. Tarantola. *Inverse Problems Theory and Methods for Model Parameter Estimation*. Society for Industerial and Applied Mathematics, 2005.

[94] A. Tarantola and B. Valette. Inverse problems = quest for information. *Journal of Geophysics*, 50(1):159–170, 1982.

[95] W. van Drongelen. Modeling neural activity. *ISRN Biomathematics*, 2013:37 pages, 2013.

[96] R. Veltz and O. Faugeras. Stability of the stationary solutions of neural field equations with propagation delays. *Journal of Mathematical Neuroscience*, 1:1–25, 2011.

[97] R. Veltz and O. Faugeras. A center manifold result for delayed neural fields equations. *SIAM J. Math. Anal., 45(3), 1527-1562*, 2013.

[98] N. A. Venkov. Dynamics of neural field models. *Ph.D. Thesis*, 2008.

[99] N. A. Venkov, S. Coombes, and P. C. Matthews. Dynamic instabilities in scalar neural field equations with space-dependent delays. *Physica D: Nonlinear Phenomena*, 232(1):1–15, 2007.

[100] P. D. Wasserman. *Neural Computing, Theory and Practice*. Van Nostrand Reinhold, 1989.

[101] T. Wennekers. Orientation tuning properties of simple cells in area v1 derived from an approximate analysis of nonlinear neural field models. *Neural Comput.*, 13(8):1721-1747, 2001.

[102] H. R. Wilson and J. D. Cowan. Excitatory and inhibitory interactions in localized populations of model neurons. *Biophys. J.*, 12(1):1–24, 1972.

[103] H. R. Wilson and J. D. Cowan. A mathematical theory of the functional dynamics of cortical and thelamic nervous tissue. *Kybernatik*, 13(2):55–80, 1973.

[104] F. Yaman, V. G. Yakhno, and R. Potthast. A survey on inverse problems for applied sciences. *Mathematical Problems in Engineering*, 2013.

Appendix A

Appendix

Here, we collect the codes which have been used to generate the images of this thesis. We have been using MATLAB or its free clone OCTAVE to carry out all calculations and simulations.

A.1 Firing function and control code

We first define the sigmoidal firing rate function f, as given by (1.3).

```
function s_out = f(s_in,eta)
s_out = 1./(1+exp(-20*(s_in-eta*ones(size(s_in)))));
```

We also set our variables and parameters as shown in the following code, where a simple Euler scheme is used to simulate the neural field equation (1.4).

```
clear all; close all;
a = 2*pi;
N = 101;                    % number of grid points
h = a/(N-1);                % grid spacing
p = 0:h:2*pi;               % grid on circle
R = 3;                      % radius for visualization
px = R*cos(p);
py = R*sin(p);
sigma = 1;                  % decay parameter for Gaussians
r1 = R*sqrt( (cos(p)+1).^2 + (sin(p)).^2);
for j=1:N
  rmat(j,:) = R*sqrt( (cos(p)-cos(p(j))).^2 +(sin(p)-sin(p(j))).^2);
end
uv(:,1) = exp(-sigma*r1.^2);
```

```
r2 = R*sqrt( (cos(p)-cos(pi/3)).^2 + (sin(p)-sin(pi/3)).^2);
r3 = R*sqrt( (cos(p)-cos(-pi/3)).^2 + (sin(p)-sin(-pi/3)).^2);
Wmat = 10*(0.3*repmat(exp(-sigma*r2.'.^2),1,N).*   % construct kernel
  repmat(exp(-sigma*r1.^2),N,1)+                   % ~
  0.3*repmat(exp(-sigma*r3.'.^2),1,N).*            % ~
  repmat(exp(-sigma*r2.^2),N,1)+                   % ~
  0.3*repmat(exp(-sigma*r1.'.^2),1,N).*            % ~
  repmat(exp(-sigma*r3.^2),N,1))*h;                % ~

Nt = 50;        % number of time steps
eta = 0.5;      % threshold
ht = 0.2;       % time stepping
tau = 1;        % time constant

fo = figure;    % start figure generating dynamics
for k=1:Nt
   disp(['k=' num2str(k)]);
   %fflush(stdout);
   for j=1:N;
      kv      = round( rmat(j,:) );
      for xi=1:N
          ec(xi,1) = f(uv(xi,max(1,k-kv(xi))),eta);
      end
  du(j,1) = ht/tau*(-uv(j,k) + Wmat(j,:)*ec)\;
 end
 uv(:,k+1) = uv(:,k)+ du;

   % visualization of current excitation function u
   if( 1==1 )
      plot3(px,py,zeros(size(px)),'k.-');
      hold on;
      plot3(px,py,uv(:,k+1),'r.-','LineWidth',3);
      axis([-R,R,-R,R,-0.1, 1.1]);
      hold off;
drawnow;
  end
end
```

A.2 Example of Chapter 4

In this MATLAB script, we show the code we use to draw Figure 4.1. The script assumes that uv has been simulated before.

```
ND=5;
M = floor(N/ND);
H = zeros(M,N);
for j=1:M
  H(j,ND*j) = 1;          % Setup Observation Operator
end
u = uv(:,1);
y = H*u;                  % Observation Simulation
alpha = 0.01;
ualpha1 = inv(alpha*eye(N,N)+H'*H)*H'*y;     % 3D-VAR with B=I
sigma = 2;
B = exp(-sigma*rmat.^2);  % Gaussian B Matrix
ualpha2 = B*H'*inv(alpha*eye(M,M)+H*B*H')*y; % 3D-VAR with B
%
fo = figure;
plot3(px,py,zeros(size(px)),'k.-');
hold on;
plot3(px,py,u,'r.-','LineWidth',3);
plot3(px,py,ualpha1,'m.-','LineWidth',3,'MarkerSize',15);
plot3(px,py,ualpha2,'b.-','LineWidth',3,'MarkerSize',15);
axis([-R,R,-R,R,-0.1, 1.1]);
hold off;
drawnow;
saveas(fo,'estimated1.png','png');
```

A.3 Examples of Chapter 6

A.3.1 Examples of Section 6.1

To show the dynamics of the activity field $u(r,t)$ on the boundary $\partial B_R \subset \mathbb{R}^2$ of a disk with radius R, as displayed in Figure 6.1 of *Example 1.*,

The script for the dynamics is:

```
uv_b(:,1) = uv(:,1);
fo = figure;
```

```
for k=1:Nt                          % loop over time steps
    disp(['k=' num2str(k)]);        % time index
    for j=1:N                       % loop over grid points
       kv = round( rmat(j,:) );     % distance to index
       for xi=1:N                   % collecting delayed input
          ec(xi,1) = f(uv_b(xi,max(1,k-kv(xi))),eta);
       end
       du(j,1) = ht/tau*(-uv_b(j,k) + Wmat_alpha(j,:)*ec);
    end
    uv_b(:,k+1) = uv_b(:,k) + du; % Euler step next

    subplot(1,2,1)                  % Visualization
    plot3(px,py,zeros(size(px)),'k.-');
    hold on;                        % of original and
    plot3(px,py,uv(:,k+1),'b.-','LineWidth',5);
    axis([-R,R,-R,R,-0.1, 1.1]);
    hold off;
    drawnow;
    subplot(1,2,2)
    plot3(px,py,zeros(size(px)),'k.-');
    hold on;                        % reconstructed dynamics
    plot3(px,py,uv_b(:,k+1),'r.-','LineWidth',5);
    axis([-R,R,-R,R,-0.1, 1.1]);
    hold off;
    drawnow;
    % save current figure with time index k
    saveas(fo,['dynamics_'num2str(k,3)'.png'],'png')
end
```

Then, for the reconstruction and visualization of the kernel as shown in Figure 6.2, we used the code:

```
for j=1:N
  kv = round(rmat(j,:));    % distance to index
  for k=1:(Nt-1)            % calculate functions psi and phi
      psi(k) = tau*(uv(j,k+1)-uv(j,k))/ht + uv(j,k);
      for xi=1:N
        phi(xi,k) = f(uv(xi,max(1,k-kv(xi))),eta);
      end
  end
%
```

```
  A = phi';                 % setup matrix and
  rhs = psi';               % right-hand side
  alpha = 0.1;              % regularization parameter
  nn = size(A,2);           % dimension of A
  % kernel reconstruction by Tikhonov Regularization
  Wmat_alpha(j,:) = (inv( alpha*eye(nn,nn)+A'*A )*A'*rhs)';
end

ca = 0.2;                   % parameter for axis control
fo = figure;                % original kernel
surf(p,p,Wmat); shading interp; caxis([0 ca]); colorbar;
axis tight;  view(30,50);  axis([0  a  0  a  -0.1 ca])
set(gca,'FontSize',14)
saveas(fo,'kernel_original.png','png')

fo = figure;                % reconstructed kernel
surf(p,p,Wmat_alpha); shading interp; caxis([0 \; ca]); colorbar;
axis tight; view(30,50); axis([0  a  0  a  -0.1  ca])
set(gca,'FontSize',14)
saveas(fo,'kernel_reconstruction.png','png')

fo = figure;                % kernel reconstruction error
surf(p,p,Wmat-Wmat_alpha); shading interp; colorbar;
axis tight; view(30,50); axis([0  a  0  a  -0.1 ca])
set(gca,'FontSize',14)
saveas(fo,'kernel_differences.png','png')
```

We then set up for example 2 of 2D neural field that is shown in Figures 6.3 and 6.4 by this code is divided in next three scripts. The code `control_delay_2d` is given first.

```
clear all; close all;
a1 = 0;                   % parameters for 2d domain
b1 = 6;                   % [a1 b1]x[a2 b2]
a = b1;
a2 = 0;
b2 = 6;
N1 = 21;                  % number of grid points
N2 = 22;                  % N1 x N2
N = N1*N2;                % total number of nodes
h1 = (b1-a1)/(N1-1);
```

```
h2 = (b2-a2)/(N2-1);
h = h1*h2;                %
x1 = a1:h1:b1;            % setting up grid
x2 = a2:h2:b2;            %
x1mat = repmat(x1,N2,1);
x2mat = repmat(x2',1,N1);
x1v = reshape(x1mat,N,1);
x2v = reshape(x2mat,N,1);

sigma = 2;                % decay parameter for Exponentials
r1 = sqrt(  (x1v-1.5).^2 + (x2v-3).^2 );
uv(:,1) = exp(-sigma*r1.^2);
r2 = sqrt(  (x1v-4.5).^2 + (x2v-4.5).^2 );
r3 = sqrt(  (x1v-4.5).^2 + (x2v-1.5).^2 );
for j=1:N                 % matrix of distances
  rmat(j,:) = sqrt( (x1v-x1v(j)).^2 + (x2v-x2v(j)).^2 );
end

if( 1 == 0 )
  fo=figure(1);           % visualization of initial field
  uvmat = reshape(uv(:,1),N2,N1);
  surf(x1,x2,uvmat)       % surf plot
  set(gca,'FontSize',14) % axes control
  view(3);                % ~
  colorbar                % ~
end

% definition of neural kernel on 2d domain
Wmat = 7*(0.3*repmat(exp(-sigma*r2.^2),1,N).*...
  repmat(exp(-sigma*r1.'.^2),N,1)+...
  0.3*repmat(exp(-sigma*r3.^2),1,N).*...
  repmat(exp(-sigma*r2.'.^2),N,1)+...
  0.3*repmat(exp(-sigma*r1.^2),1,N).*...
  repmat(exp(-sigma*r3.'.^2),N,1))*h;


Nt = 30;              % number of time steps
eta = 0.5;            % threshold for neural processing
ht = 0.2;             %
tau = 1;              % temporal parameter
```

```
fo = figure;          % visualize dynamical evolution
for k=1:Nt            % temporal loop
    disp(['k=' num2str(k)]);

    for j=1:N;        % loop over grid points
        kv      = 1.0*round( rmat(j,:) );
        for xi=1:N    % collect temporal delay
           ec(xi,1) = f(uv(xi,max(1,k-kv(xi))),eta);
        end           %
        du(j,1) = ht/tau*(-uv(j,k) + Wmat(j,:)*ec);
    end               % ODE Euler step
    uv(:,k+1) = uv(:,k)+ du;

    if( 1==0 )
      uvmat = reshape(uv(:,k+1),N2,N1);
      fo=figure(1);
      surf(x1,x2,uvmat)
      set(gca,'FontSize',14)
      view(2);
      axis([a1 b1 a2 b2 0 1]);
      hold off;
      drawnow;
      if( mod(k,3) == 0 )
        filename=['dyn_' sprintf('%02d',k)];
        saveas(fo,filename,'png')
      end
    end
end
```

Then, the code `inverse_delay_2d` which is:

```
for j=1:N                % loop over grid points
  disp(['j=' num2str(j)]);
  fflush(stdout);
  kv = round(rmat(j,:)); % distance to index
  for k=1:(Nt-1)         % define functions psi and phi
      psi(k) = tau*(uv(j,k+1)-uv(j,k))/ht + uv(j,k);
      for xi=1:N         % collect past influence
        phi(xi,k) = f(uv(xi,max(1,k-kv(xi))),eta);
      end
```

```
  end
  A = phi';              % setup reconstruction matrix
  rhs = psi';            % and right-hand side
  alpha = 0.1;           % regularization parameter
  nn = size(A,2);        % dimension
  % kernel reconstruction by Tikhonov Regularization
  Wmat_alpha(j,:) = (inv( alpha*eye(nn,nn)+A'*A )*A'*rhs)';
end

% display the kernel and its reconstruction
ca = 0.2;                % display parameter
fo = figure;             % show original kernel
colormap(pink);          % setup colormap for visualization
colormap(flipud(colormap))
surf(Wmat); shading interp; caxis([0 ca]); colorbar;
axis tight; view(30,50); axis([1 N 1 N -0.01 ca])
set(gca,'FontSize',14)
saveas(fo,'kernel_original.png','png')

fo = figure;             % show reconstructed kernel
colormap(pink)           %
colormap(flipud(colormap))
surf(Wmat_alpha); shading interp; caxis([0 ca]); colorbar;
axis tight; view(30,50); axis([1 N 1 N -0.01 ca])
set(gca,'FontSize',14)
saveas(fo,'kernel_reconstruction.png','png')

fo = figure;             % show kernel reconstruction error
colormap(pink)           %
colormap(flipud(colormap))
surf(Wmat-Wmat_alpha); shading interp; colorbar;
axis tight; view(30,50); axis([1 N 1 N -0.01 ca])
set(gca,'FontSize',14)
saveas(fo,'kernel_differences.png','png')
```

Then, the code `show_inverse_result_2d` calculates the dynamical behaviour with reconstructed kernel.

```
colormap(bone)           % setup colormap
colormap(flipud(colormap))
uv_b(:,1) = uv(:,1);     % initial excitation field
```

```
fo = figure;              % figure for dynamical display
for k=1:Nt                % temporal loop
    disp(['k=' num2str(k)]);
    for j=1:N;            % loop over grid points
        kv      = round( rmat(j,:) );
        for xi=1:N        % collecting influence
           ec(xi,1) = f(uv_b(xi,max(1,k-kv(xi))),eta);
        end               % ODE Euler Step
        du(j,1) = ht/tau*(-uv_b(j,k) + Wmat_alpha(j,:)*ec);
    end                   % integrate NFE
    uv_b(:,k+1) = uv_b(:,k)+ du;

    subplot(1,2,1)          % show original dynamics
    colormap(bone)          %
    colormap(flipud(colormap))
    uvmat = reshape(uv(:,k+1),N2,N1);
    surf(x1,x2,uvmat)       % surf plot
    set(gca,'FontSize',14) % plot parameter settings
    view(3);                %
    axis([a1 b1 a2 b2 0 1]);
    axis equal; set(gca,'ztick',[])
    title('Original')
    hold off;

    hold off;
    drawnow;
    %
    subplot(1,2,2)          % show reconstructed dynamics
    colormap(pink)          %
    colormap(flipud(colormap))
    uvmat = reshape(uv_b(:,k+1),N2,N1);
    surf(x1,x2,uvmat)       % surf plot
    set(gca,'FontSize',14)  % plot parameter settings
    view(3);                %
    axis([a1 b1 a2 b2 0 1]);
    axis equal; set(gca,'ztick',[])
    title('Reconstruction')
    hold off;

    hold off;
```

```
    drawnow;
    saveas(fo,['dynamics_' sprintf('%02d',k) '.png'],'png')
end
```

A.3.2 Figures of Section 6.2

we used this code to produce the figure 6.5.

```
% run control_delay_1d.m first !!!
%
close all;

sv = 2*pi/(Nt+1)*(1:(Nt+1));
eps = 0.01;

rand ("seed", 10)
for jj=1:N
  cr = 2*(rand(1,1)-0.5);
  uv(jj,:) = uv0(jj,:) + eps*(sin(sv+cr*2*pi));
end

if( 1 == 1 )
fo = figure;
%
subplot(2,1,1)
colormap(flipud(pink))
surf(uv0);
set(gca,'FontSize',14)
view(2);
shading flat;
caxis([0 1])
axis tight;
xlabel('time steps')
ylabel('point index')
title('input signal u')
colorbar;

%
subplot(2,1,2)
```

```
colormap(flipud(pink))
surf(uv-uv0);
set(gca,'FontSize',14)
view(2);
shading flat;
caxis([-eps eps])
axis tight;
xlabel('time steps')
ylabel('point index')
title('noise on input');
colorbar;

%saveas(fo,'sensitivity_input.png','png');
end

j0 = 20;
for j=j0;  %1:N
  disp(['j=' num2str(j)]);
  %
  kv      = round( rmat(j,:) );

  for k=1:(Nt-1)

      psi(k) = tau*(uv(j,k+1)-uv(j,k))/ht + uv(j,k);
      for xi=1:N
        phi(xi,k) = f(uv(xi,max(1,k-kv(xi))),eta);
      end
  end
  %
  A = phi';
  rhs = psi';
  alpha = 0.01;
  nn = size(A,2);
  % solving A*Wmat =
  Wmat_alpha(j,:) = (inv( alpha*eye(nn,nn)+A'*A )*A'*rhs)';

end

fo = figure;
plot(Wmat(j0,:),'LineWidth',3);
```

```
hold on;
plot(Wmat_alpha(j0,:),'LineWidth',3);
set(gca,'FontSize',14);
xlabel('point index');
```

The next code is for Figure 6.6:

```
disp('--- p3_inverse.m ----------------')
for j=1:N
  disp(['j=' num2str(j)]);
  fflush(stdout);
  %
  kv      = round( rmat(j,:) );

  for k=1:(Nt-1)

      psi(k) = tau*(uv(j,k+1)-uv(j,k))/ht + uv(j,k);
      for xi=1:N
        phi(xi,k) = f(uv(xi,max(1,k-kv(xi))),eta);
      end
  end
  %
  A = phi';
  rhs = psi';
  alpha = 0.1;
  nn = size(A,2);
  % solving A*Wmat =
  Wmat_alpha(j,:) = (inv( alpha*eye(nn,nn)+A'*A )*A'*rhs)';

end

%-----------------------------------------------
% solve the inverse problem next
%-----------------------------------------------
ca = 0.2;
fo = figure;
      colormap(pink)
      colormap(flipud(colormap))

surf(p,p,Wmat); shading interp; caxis([0 ca]); colorbar;
axis tight; view(30,50); axis([0 a 0 a -0.01 ca])
```

```
set(gca,'FontSize',14)
saveas(fo,['kernel_original_alpha=' num2str(alpha) '.png'],'png')

fo = figure;
      colormap(pink)
      colormap(flipud(colormap))

surf(p,p,Wmat_alpha); shading interp; caxis([0 ca]); colorbar;
axis tight; view(30,50); axis([0 a 0 a -0.01 ca])
set(gca,'FontSize',14)
saveas(fo,['kernel_reconstruction_alpha=' num2str(alpha) '.png'],'png')

fo = figure;
      colormap(pink)
      colormap(flipud(colormap))

surf(p,p,Wmat-Wmat_alpha); shading interp; colorbar;
axis tight; view(30,50); axis([0 a 0 a -0.01 ca])
set(gca,'FontSize',14)
saveas(fo,['kernel_differences_alpha=' num2str(alpha) '.png'],'png')
```

A.3.3 Figures of Section 6.3

These two codes are used for Figures 6.7 and 6.8.

```
The visualization of the background error covariance matrix
is produced by the code:

fobj13 = figure(13);
%set(fobj13,'Renderer','zbuffer');

surf(B);
shading interp;
axis equal;
axis tight;
view(2);
set(gca,'FontSize',14);
```

```
saveas(fobj13,'B_matrix.png','png');

where B matrix is introduced in the following code

%close all
disp('running neuro_da.m')
% kt = 3;
disp(['kt=' num2str(kt)]);
%
% run neuro_sim.m before!
%
% Setup observation operator H:
m = 50;
% m = 900;
H = zeros(m,N);
ind1 = 1:floor(N/m):N;
chi = zeros(N,1);
for j=1:m
    H(j,ind1(j)) = 1;
    chi(ind1(j)) = 1;
end
x = phi(:,kt);
y = H*x;
yv = zeros(N,1);
for j=1:m
    yv(ind1(j))= y(j);
end
yAll(:,kt,iter)=yv;
%
% calculate simulation data
%
% Setup B matrix
sigmaB = 0.1;
x1vecM = repmat(x1vec,1,N);
x2vecM = repmat(x2vec,1,N);
B = exp(-sigmaB*sqrt((x1vecM-x1vecM').^2+(x2vecM-x2vecM').^2));
% figure; surf(B);view(2);shading flat;
%
% reconstruct x from measurements
alphaR = 0.01;
alphaR2 = 0;
```

```
xb = phi4(:,kt);
xa = xb + B*H'*inv(alphaR*eye(m,m)+H*B*H')*(y-H*xb);
xa0 = B*H'*inv(alphaR2*eye(m,m)+H*B*H')*(y);

if( 1==1)
    % visualize data versus
    fobj = figure(5);
    %set(fobj,'Renderer','zbuffer');
    subplot(4,1,1)
    surf(x1,x2,reshape(x,N2,N1));
    caxis([0,1])
    view(2);
    shading flat;
    axis equal;
    axis tight;
    %
    subplot(4,1,2)
    surf(x1,x2,reshape(yv,N2,N1));
    caxis([0,1])
    view(2);
    shading flat;
    axis equal;
    axis tight;
    %
    subplot(4,1,3)
    surf(x1,x2,reshape(xa0,N2,N1));
    caxis([0,1])
    view(2);
    shading flat;
    axis equal;
    axis tight;
    %
    subplot(4,1,4)
    surf(x1,x2,reshape(xa,N2,N1));
    caxis([0,1])
    view(2);
    shading flat;
    axis equal;
    axis tight;
```

```
    saveas(fobj,['da_est_' num2str(kt)],'png');
    drawnow;
end
```

A.3.4 Figures of Section 6.4

The next codes are used to generate Figure 6.9. The first code is `control_all`:

```
% first generate nature run and data
control_delay_1d;

% now do the loop of estimation and inversion
ualpha1 = zeros(N,Nt+1);
Nit = 5;                        % number of iterations
for la=1:Nit                    % iteration loop 3D-VAR <-> kernel rec
  disp(['la=' num2str(la)])
  tic;
  estimation;                   % Estimate
  t = toc;
  disp(['time needed: t=' num2str(t) 's'])
  inverse_delay_1d;             % Learning / Kernel Reconstruction
  t = toc;
  disp(['time needed: t=' num2str(t) 's'])
  ua_all(:,:,la) = ualpha2;
  integrate_kernel;             % forward integration
  t = toc;
  disp(['time needed: t=' num2str(t) 's'])
  ub_all(:,:,la) = uv_b;
  ualpha1=uv_b;                 % first guess for next estimate
end
```

This is the code `control_delay_1d`:

```
a = 2*pi;               % setup grid on circular domain
N = 101;                % number of grid points
h = a/(N-1);            % grid spacing
p = 0:h:2*pi;           % vector of angles
R = 3;                  % radius for visualizations

px = R*cos(p);          % grid points
py = R*sin(p);          % ~
```

```
sigma = 1;              % decay parameter
r1 = R*sqrt( (cos(p)+1).^2 + (sin(p)).^2);
for j=1:N               % distance matrix
  rmat(j,:) = R*sqrt( (cos(p)-cos(p(j))).^2 + (sin(p)-sin(p(j))).^2);
end                     % initial field
uv(:,1) = exp(-sigma*r1.^2);

r2 = R*sqrt( (cos(p)-cos(pi/3)).^2 + (sin(p)-sin(pi/3)).^2);
r3 = R*sqrt( (cos(p)-cos(-pi/3)).^2 + (sin(p)-sin(-pi/3)).^2);
%Wmat = zeros(N,N);
Wmat = 10*(0.3*repmat(exp(-sigma*r2.'.^2),1,N).*...
  repmat(exp(-sigma*r1.^2),N,1)+...
  0.3*repmat(exp(-sigma*r3.'.^2),1,N).*...
  repmat(exp(-sigma*r2.^2),N,1)+...
  0.3*repmat(exp(-sigma*r1.'.^2),1,N).*...
  repmat(exp(-sigma*r3.^2),N,1))*h;


Nt = 25;                % number of time steps
eta = 0.5;              % threshold in neural equation
ht = 0.2;               % time stepping
tau = 1;                % time parameter
for k=1:Nt              % temporal loop
    if( mod(k,10)==0 )
      disp(['k=' num2str(k)]);
    end
    drawnow;
    for j=1:N           % loop over grid points
        kv      = 1.0*round( rmat(j,:) );
        for xi=1:N      % collect temporal influence
           ec(xi,1) = f(uv(xi,max(1,k-kv(xi))),eta);
        end             % ODE Euler Step
        du(j,1) = ht/tau*(-uv(j,k) + Wmat(j,:)*ec);
    end
    uv(:,k+1) = uv(:,k)+ du;

    if( 1==0 )
      fo = figure(1);     % visualize simulated excitation
      plot3(px,py,zeros(size(px)),'k.-');
```

```
      hold on;
      plot3(px,py,uv(:,k+1),'r.-','LineWidth',3);
      axis([-R,R,-R,R,-0.1, 1.1]);
      hold off;
      drawnow;
    end
end

uv0 = uv;
```

Then, we run the estimation code.

```
ND=5;                   % measure every ND-th excitation
M = floor(N/ND);        %
H = zeros(M,N);         % setup observation operator
for j=1:M               % ~
  H(j,ND*j) = 1;        % ~
end                     % ~

for jj=1:Nt
    u = uv(:,jj);       % original field
    y = H*u;            % simulated observation

    alpha = 0.5;        % regularization parameter
    sigma = 2;          % decay parameter for Gaussian kernel
    B = exp(-sigma*rmat.^2);  % Gaussian B matrix
    % 3D-VAR estimation step
    ualpha2(:,jj) = ualpha1(:,jj) ...
        + B*H'*inv(alpha*eye(M,M)+H*B*H')*(y-H*ualpha1(:,jj));

    if( 1 == 0 )
        fo = figure(2); % visualize original u and
        plot3(px,py,zeros(size(px)),'k.-');
        hold on;
        plot3(px,py,u,'r.-','LineWidth',3);
        % visualize first guess ualpha1 and 3D-VAR estimate ualpha2
        plot3(px,py,ualpha1(:,j),'b.-','LineWidth',3,'MarkerSize',15);
        plot3(px,py,ualpha2(:,j),'m.-','LineWidth',3,'MarkerSize',15);
        axis([-R,R,-R,R,-0.1, 1.1]);
        hold off;
        drawnow;
```

```
    end
end
uav = ualpha2;
%saveas(fo,'estimated.png','png');
```

Then, we run `inverse_delay_1d`.

```
for j=1:N                % loop over gridpoints
  if( mod(j,20)==0 )
    disp(['j=' num2str(j)]);
  end
  drawnow;
  %
  kv = round(rmat(j,:)); % distance to index
  for k=1:(Nt-1)         % temporal loop
     psi(k) = tau*(uav(j,k+1)-uav(j,k))/ht + uav(j,k);
     for xi=1:N          % collect temporal influence
        phi(xi,k) = f(uav(xi,max(1,k-kv(xi))),eta);
     end
  end
  %
  A = phi';              % setup reconstruction matrix
  rhs = psi';            % and right-hand side
  alpha = 0.1;           % regularization parameter
  nn = size(A,2);        % dimension
  % kernel reconstruction by Tikhonov Regularization
  Wmat_alpha(j,:) = (inv( alpha*eye(nn,nn)+A'*A )*A'*rhs)';
end

% visualize kernel reconstructions
if( 1 == 0 )
    ca = 0.2;
    fo = figure(3);            % visualize original kernel
    colormap(pink)
    colormap(flipud(colormap))
    surf(p,p,Wmat); shading interp; caxis([0 ca]); colorbar;
    axis tight; view(30,50); axis([0 a 0 a -0.01 ca])
    set(gca,'FontSize',14)
    saveas(fo,['kernel_original_alpha=' num2str(alpha) '.png'],'png')

    fo = figure;               % visualize reconstructed kernel
```

```
    colormap(pink)
    colormap(flipud(colormap))
    surf(p,p,Wmat_alpha); shading interp; caxis([0 ca]); colorbar;
    axis tight; view(30,50); axis([0 a 0 a -0.01 ca])
    set(gca,'FontSize',14)
    saveas(fo,['kernel_reconstruction_alpha=' num2str(alpha) '.png'],'png')

    fo = figure;               % visualize kernel reconstruction error
    colormap(pink)
    colormap(flipud(colormap))
    surf(p,p,Wmat-Wmat_alpha); shading interp; colorbar;
    axis tight; view(30,50); axis([0 a 0 a -0.01 ca])
    set(gca,'FontSize',14)
    saveas(fo,['kernel_differences_alpha=' num2str(alpha) '.png'],'png')
end

uv_b(:,1) = uv(:,1);

for k=1:Nt                  % temporal loop
    if( mod(k,20) == 0)     %
       disp(['k=' num2str(k)]);
    end
    drawnow;
    for j=1:N;              % loop over grid points
        kv      = round( rmat(j,:) );
        for xi=1:N          % collect temporal influence
           ec(xi,1) = f(uv_b(xi,max(1,k-kv(xi))),eta);
        end                 % ODE Euler Step
        du(j,1) = ht/tau*(-uv_b(j,k) + Wmat_alpha(j,:)*ec);
    end
    uv_b(:,k+1) = uv_b(:,k)+ du;
end
```