

A thesis submitted in partial fulfillment of the
requirements for the degree Master of Science

Identification of Stochastic Partial Differential Equations from Data

Alina Barbara Steinberg
Born 8.10.92 in Bochum

2020

Institute of Theoretical Physics
Westfälische Wilhelms-Universität Münster



Primary supervisor:	Prof. Dr. U. Thiele
Secondary supervisor:	Dr. O. Kamps
Handover date:	27. August 2020

Abstract

This thesis seeks to identify stochastic partial differential equations from field data by way of extending the method of maximum likelihood estimation to encompass partial differential equations.

Applying this method to generated one-dimensional field data sets leads to the conclusion that there is no universal rule as to how noisy the data can be for the method to succeed. The one-dimensional reaction-diffusion equation gives good results even for relatively big noise levels, while data generated with the Swift-Hohenberg equation needs a low noise level for successful model identification. Nevertheless, given a reasonable noise level, the method results in reliable results especially if combined with sparsity promoting methods such as BIC or thresholding.

Applying the maximum likelihood estimation method to data generated from the two-dimensional Brusselator equation further proves that successful identification is possible in multiple dimensions.

Compared with the SINDy method for identification of stochastic partial differential equations, the maximum likelihood estimation can accurately identify models from data sets with bigger noise levels.

Applying the maximum likelihood estimation to field data of the Ising model, which is generated as a cellular automaton, does not show results which match the original data set. Some cases don't even match the dynamic, others grow far beyond the size of the spin values.

Contents

1	Introduction	1
2	Basics of Stochastic Time Series Analysis	2
2.1	Stochastic Time Series Analysis	2
2.2	Bayesian Estimation of Parameters	6
2.3	Sparse Identification of Nonlinear Dynamics	11
3	Stochastic Time Series Analysis for Spatio-Temporal Data	14
3.1	Stochastic Spatio-Temporal Systems	14
3.2	Spatio-Temporal Short-Time Propagator	14
3.3	Spatio-Temporal Maximum Likelihood Estimation	15
3.4	Multi-Dimensional Maximum Likelihood Estimation	15
3.5	The Maximum Likelihood Estimation Program	16
4	Identifying Models of Generated Spatio-Temporal Data Sets	18
4.1	1D Reaction-Diffusion Equation	18
4.2	Swift-Hohenberg Equation	28
4.3	The Brusselator	35
5	Identifying a Stochastic Partial Differential Equation for Field Data of the Ising Model	42
5.1	The Two-Dimensional Ising Model	42
5.2	Generating the Data	43
5.3	Results	44
6	Conclusion and Outlook	50
A	Appendix	52
A.1	Parameters for Data Set Generation and their MLE Results	52
A.2	Hyperparameter Optimization Results	54
	Bibliography	58

List of Figures

4.1	Snapshots of 1D reaction-diffusion data sets with different noise levels.	19
4.2	Histograms of the probability distributions for the parameters of the 1D reaction-diffusion equation, where the blue lines represent the real parameter values.	23
4.3	Snapshots of Swift-Hohenberg data sets with different noise levels.	29
4.4	Histograms of the probability distributions for the parameters for the Swift-Hohenberg equation, where the blue lines represent the real parameter values.	34
4.5	Snapshot of a Brusselator data set with the noise magnitude $q_1 = q_2 = 0$	35
4.6	Snapshot of a Brusselator data set with the noise magnitude $q_1 = q_2 = 0.5$	36
4.7	Histograms of the probability distributions for the parameters of the u field of the Brusselator equation, where the blue lines represent the real parameter values.	40
4.8	Histograms of the probability distributions for the parameters of the v field of the Brusselator equation, where the blue lines represent the real parameter values.	41
5.1	Example for the development of spins on a grid according to the Ising model, where black and white denote different spin directions.	43
5.2	Development of a blurred Ising data set at $T = 0.01$, where black and green denote different spin directions.	44
5.3	Snapshots of data sets calculated from the results of the maximum likelihood estimation seen in table 5.1.	46
5.4	Snapshots of the data set calculated from the result of the maximum likelihood estimation with the threshold $\lambda = 0.15$ seen in table 5.2.	48
5.6	Snapshot of a blurred Ising data set at $t = 0$ (left) as well as after three time steps (middle) and a snapshot from a data set calculated from equation (5.3.3) with the $t = 0$ field as starting field (right).	49

- 5.7 Snapshot of a blurred Ising data set at $t = \tau$ (left) as well as after two more time steps (middle) and a snapshot from a data set calculated from equation (5.3.3) with the $t = \tau$ field as starting field (right). . 49

List of Tables

4.1	Maximum likelihood estimation results for 1D reaction-diffusion data sets with different noise levels and a minimum of parameters according to equations (4.1.2) and (4.1.3), as well as the percent error δ . . .	19
4.2	SINDy results with ordinary least squares regression for 1D reaction-diffusion data sets with different noise levels and a minimum of parameters according to equation (4.1.2).	20
4.3	Percent error δ for the $D^{(1)}$ results of the maximum likelihood estimation and the SINDy method for 1D reaction-diffusion data sets with different noise levels as found in tables 4.1 and 4.2.	20
4.4	Maximum likelihood estimation results for a 1D reaction-diffusion data set with the noise level $q_1 = q_2 = 1$, possible parameters up to the third polynomial of $1, u, \Delta u$ and $\Delta^2 u$ and different numbers of iterations of minimization.	24
4.5	SINDy results with iterative hard-thresholding for a 1D reaction-diffusion data set with a noise level of $q_1 = q_2 = 1$ and possible parameters up to the third polynomial of $1, u, \Delta u$ and $\Delta^2 u$. Only parameters $\neq 0$ are listed.	24
4.6	Excerpts of the results of the hyperparameter optimization for the thresholding parameter λ applied to a 1D reaction-diffusion data set with ten parameters and one iteration of minimization.	25
4.7	All tested thresholds from the hyperparameter optimization applied to a 1D reaction-diffusion data set with ten parameters and one iteration of minimization. Optimal thresholds are marked in gray. . .	25
4.8	Maximum likelihood estimation results for a 1D reaction-diffusion data set with thresholding at $\lambda = 0.1$, different numbers of iterations of minimization and a shrinking amount of possible parameters according to the previous result.	26
4.9	Excerpt of the results of the hyperparameter optimization for the thresholding parameter λ applied to a 1D reaction-diffusion data set with the BIC as cost function, as well as a result without the BIC. .	27
4.10	Maximum likelihood estimation results for Swift-Hohenberg data sets with varying noise levels and a minimum of parameters according to equations (4.2.2) and (4.2.3), as well as the percent error δ	30

4.11	SINDy results with ordinary least squares regression and a minimum of parameters according to equation (4.2.2) for Swift-Hohenberg data sets with varying noise levels.	31
4.12	Maximum likelihood estimation results for a Swift-Hohenberg data set with the noise level $q_1 = q_2 = 0.01$, possible parameters up to the third polynomial of $1, u, \Delta u$ and $\Delta^2 u$ and different numbers of iterations of minimization as well as thresholding with the parameter λ	32
4.13	SINDy results with different regression methods for a Swift-Hohenberg data set with the noise level $q_1 = q_2 = 0.01$ and possible parameters up to the third polynomial of $1, u, \Delta u$ and $\Delta^2 u$	33
4.14	Maximum likelihood estimation results for Brusselator data sets with varying noise levels and a minimum of parameters according to equations (4.3.2) and (4.3.4).	37
4.15	Maximum likelihood estimation results for Brusselator data sets with varying noise levels and a minimum of parameters according to equations (4.3.3) and (4.3.4).	37
4.16	Percent error δ for the results of the maximum likelihood estimation for Brusselator data sets with varying noise levels as found in tables 4.14 and 4.15.	38
4.17	SINDy results with ordinary least squares (gray cells) or iterative hard-thresholding with a threshold of $\lambda = 0.001$ for Brusselator data sets with varying noise levels. A minimum of possible parameters is used according to equations (4.3.2) and (4.3.3).	38
4.18	Percent error δ for the $D^{(1)}$ results of the maximum likelihood estimation and the SINDy method for Brusselator data sets with varying noise levels as found in tables 4.14, 4.15 and 4.17.	39
5.1	Maximum likelihood estimation results for a blurred Ising data set at $T = 0.01$ with parameters up to the third polynomial of $u, \Delta u$ and $\Delta^2 u$. Different thresholds λ are used for truncation.	45
5.2	Maximum likelihood estimation results for a blurred Ising data set at $T = 0.01$ with parameters up to the third polynomial of $u, \Delta u$ and $\Delta^2 u$ and an additional parameter for the sum of the neighboring spins. Different thresholds λ are used for truncation.	47
A.1	MLE results for $D^{(1)}$ for 1D reaction-diffusion data sets generated with different parameters.	52
A.2	MLE results for $D^{(1)}$ for 1D Swift-Hohenberg data sets generated with different parameters.	53

A.3	MLE results for $D^{(1)}$ for 2D Brusselator data sets generated with different parameters.	54
A.4	Part one of the results of the MLE for the 1D reaction-diffusion equation with ten parameters, hyperparameter optimization for thresholding and one iteration of minimization. Grey cells mark optimal results.	54
A.5	Part two of the results of the MLE for the 1D reaction-diffusion equation with ten parameters, hyperparameter optimization for thresholding and one iteration of minimization. Grey cells mark optimal results.	55
A.6	Part three of the results of the MLE for the 1D reaction-diffusion equation with ten parameters, hyperparameter optimization for thresholding and one iteration of minimization. Grey cells mark optimal results.	56
A.7	Results of the MLE for the 1D reaction-diffusion equation with ten parameters, BIC and hyperparameter optimization for thresholding after one iteration of minimization. Grey cells mark results closest to expected values, blue cells mark smallest BIC values.	57

List of Abbreviations

BIC Bayesian information criterion

IHT iterative hard-thresholding

LASSO least absolute shrinkage and selection operator

MCMC Markov chain Monte Carlo

MLE maximum likelihood estimation

PDF probability density function

SINDy sparse identification of nonlinear dynamics

SPDE stochastic partial differential equation

STLS sequential thresholded least-squares

1 Introduction

The topic of this thesis is the derivation of models in the form of stochastic partial differential equations from field data.

The ability to derive models from data is of interest to many different areas of science, such as ecology, climate science, epidemiology or neuroscience [7], as it allows both a deeper understanding of the process and gives a chance at predicting its behavior.

Models derived from spatially extended systems are particularly interesting for studying pattern formation [19], fluid dynamics [24] and molecular dynamics [6]. An additional hope is to translate cellular automata into differential equations.

Many different methods for model derivation from time series data exist, such as using the SINDy algorithm [7], possibly in combination with kernel regression methods to straighten the data [14], or a maximum likelihood approach [16].

The SINDy method has been extended for use with partial differential equations [24], but can only handle small noise levels. Hoping that it will prove more resistant to noise, this thesis focuses on adapting the maximum likelihood approach for use on spatially extended data sets.

As a first step the basics of stochastic time series analysis for ordinary differential equations are explained, starting with general information on models which describe stochastic time series data (see subsection 2.1), followed by the basics of Bayesian estimation of parameters (see subsection 2.2), which include the maximum likelihood estimation and Markov chain Monte Carlo methods. The SINDy method is explained in subsection 2.3 for the purpose of comparison.

Section 3 describes the adaption of the maximum likelihood approach to field data and how the method is realized in a Python program. The method is then tested on generated data sets for one- and two-dimensional fields, the results of which are described in section 4.

Finally the maximum likelihood method is tested on field data of the two-dimensional Ising model. The theory as well as the results are detailed in section 5.

2 Basics of Stochastic Time Series Analysis

2.1 Stochastic Time Series Analysis

The key points of stochastic time series analysis are explained in the following sections. For a more exhaustive overview see reference [14].

2.1.1 Stochastic Processes

The model of a stochastic process can be described by an N -point probability density function (PDF) as defined in equation (2.1.1), where N is the amount of points in time.

$$f_N(q_N, t_N; q_{N-1}, t_{N-1}; \dots; q_1, t_1) \quad (2.1.1)$$

The PDF gives the probability that a stochastic process $Q(t)$ returns the values q_1, \dots, q_N at times t_1, \dots, t_N and can be split into a product, as shown in equation (2.1.2), where p is a conditional PDF and f_{N-1} is the $(N-1)$ -point PDF.

$$f_N(q_N, t_N; \dots; q_1, t_1) = p(q_N, t_N | q_{N-1}, t_{N-1}; \dots; q_1, t_1) \cdot f_{N-1}(q_{N-1}, t_{N-1}; \dots; q_1, t_1) \quad (2.1.2)$$

2.1.2 Normal Distribution

The normal or Gaussian distribution (see also [2]) is a continuous probability distribution with the form seen in equation (2.1.3).

$$N(\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right) \quad (2.1.3)$$

μ represents the mean which gives the position of the maximum of the distribution and therefore the value that a random variable drawn from this distribution is most likely to take.

σ gives the standard deviation, which influences the width of the distribution and therefore how likely it is for a random variable to stray far from the mean.

2.1.3 Markov Processes

The Markov process is a stochastic process $Q(t)$ where the probability of producing the state q_N depends on the state q_{N-1} while information on other previous states has no influence. A mathematical description is found in equation (2.1.4).

$$p(q_N, t_N | q_{N-1}, t_{N-1}; \dots; q_1, t_1) = p(q_N, t_N | q_{N-1}, t_{N-1}) \quad (2.1.4)$$

Given a reasonable time step and unless a time delay is included, continuous differential equations can be considered Markov processes since knowing the state q_i and its time derivative \dot{q}_i at the time t_i is all that is needed to make a prediction of the state of the system at the time t_{i+1} .

Applying equation (2.1.2) to a Markov process makes it possible to describe the PDF as a product of conditional PDFs and a PDF of the first state f_1 , as shown in equation (2.1.5).

$$f_N(q_N, t_N; \dots; q_1, t_1) = f_1(q_1, t_1) \prod_{i=1}^{N-1} p(q_{i+1}, t_{i+1} | q_i, t_i) \quad (2.1.5)$$

If the process is stationary, therefore giving it a constant mean and variance, the conditional PDF depends on the time step $\tau = t_{i+1} - t_i$ which is independent of the specific value of i . This conditional PDF can be found in (2.1.6) and in combination with the Chapman-Kolmogorov equation for $p_{2\tau}$ (see (2.1.7)) allows complete information about the Markov process.

$$p_\tau(q' | q) = p(q', t + \tau | q, t) \quad (2.1.6)$$

$$p_{2\tau}(q'' | q) = \frac{f_2(q'', q)}{f_1(q)} = \int p_\tau(q'' | q') p_\tau(q' | q) dq' \quad (2.1.7)$$

2.1.4 Fokker-Planck Equation

The Fokker-Planck equation gives form to the transition PDFs by using the first two moments of a Kramers-Moyal extension of a Taylor expansion of equation (2.1.7), which is valid if the process is Gaussian distributed in the time limit $\tau \rightarrow 0$. It can be found in equation (2.1.8) and uses the initial condition $p(q', t' | q, t) = \delta(q' - q)$.

$$\frac{\partial}{\partial t'} p(q', t' | q, t) = \left[-\frac{\partial}{\partial q'} D^{(1)}(q', t') + \frac{\partial^2}{\partial q'^2} D^{(2)}(q', t') \right] p(q', t' | q, t) \quad (2.1.8)$$

$D^{(1)}$ and $D^{(2)}$ are called Kramers-Moyal coefficients and can be calculated using equation (2.1.9), where $M_\tau^{(n)}$ are called the conditional moments.

$$D^{(n)}(q, t) = \lim_{\tau \rightarrow 0} \frac{1}{n! \tau} \underbrace{\langle (Q(t + \tau) - Q(t))^n | Q(t) = q \rangle}_{M_\tau^{(n)}(q, t)} \quad (2.1.9)$$

$D^{(1)}(q, t)$ is also called drift coefficient and gives information about the time development of the deterministic part of the system. $D^{(2)}(q, t)$ is the diffusion coefficient and gives information on the influence of the noise.

In the limit $\tau = t' - t \rightarrow 0$ the transition PDF as shown in equation (2.1.10) can be derived. It is called short-time propagator.

$$p(q', t + \tau | q, t) = \frac{1}{2\sqrt{\pi D^{(2)}(q, t)\tau}} \exp\left(-\frac{(q' - q - D^{(1)}(q, t)\tau)^2}{4D^{(2)}(q, t)\tau}\right) \quad (2.1.10)$$

The Fokker-Planck equation in a multivariate case has the form seen in equation (2.1.11), with a short-time propagator as shown in equation (2.1.12) [23]. d is the dimension of the vectors \mathbf{q} and $\mathbf{D}^{(1)}$. $\mathbf{D}^{(2)}$ is a $d \times d$ matrix.

$$\frac{\partial}{\partial t'} p(\mathbf{q}', t' | \mathbf{q}, t) = \left[-\sum_i^d \frac{\partial}{\partial q'_i} D_i^{(1)}(\mathbf{q}', t') + \sum_{i,j}^d \frac{\partial^2}{\partial q'_i \partial q'_j} D_{ij}^{(2)}(\mathbf{q}', t') \right] p(\mathbf{q}', t' | \mathbf{q}, t) \quad (2.1.11)$$

$$p(\mathbf{q}', t' | \mathbf{q}, t) = \frac{\exp\left(-\frac{1}{4\tau} [\mathbf{q}' - \mathbf{q} - \mathbf{D}^{(1)}(\mathbf{q}, t)\tau] [\mathbf{D}^{(2)}(\mathbf{q}, t)]^{-1} [\mathbf{q}' - \mathbf{q} - \mathbf{D}^{(1)}(\mathbf{q}, t)\tau]\right)}{(2\sqrt{\pi\tau})^d \sqrt{\text{Det}[\mathbf{D}^{(2)}(\mathbf{q}, t)]}} \quad (2.1.12)$$

The accompanying Kramers-Moyal coefficients can be found in equations (2.1.13) and (2.1.14).

$$D_i^{(1)}(\mathbf{q}, t) = \lim_{\tau \rightarrow 0} \frac{1}{\tau} \langle Q_i(t + \tau) - Q_i(t) | \mathbf{Q}(t) = \mathbf{q} \rangle \quad (2.1.13)$$

$$D_{ij}^{(2)}(\mathbf{q}, t) = \lim_{\tau \rightarrow 0} \frac{1}{2\tau} \langle (Q_i(t + \tau) - Q_i(t))(Q_j(t + \tau) - Q_j(t)) | \mathbf{Q}(t) = \mathbf{q} \rangle \quad (2.1.14)$$

2.1.5 Langevin Equation

The Langevin equation is a dynamic evolution equation, derived from the Fokker-Planck equation and can be found in (2.1.15).

$$\dot{q}(t) = h(q(t), t) + g(q(t), t)\Gamma(t) \quad (2.1.15)$$

h represents the deterministic part of the dynamic, while g is the amplitude of the noise represented by a stochastic δ -correlated force $\Gamma(t)$ with zero mean, which therefore has to satisfy equations (2.1.16) and (2.1.17).

$$\langle \Gamma(t) \Gamma(t') \rangle = 2\delta(t - t') \quad (2.1.16)$$

$$\langle \Gamma(t) \rangle = 0 \quad (2.1.17)$$

The connection between the Langevin and the Fokker-Planck equation comes from the connection between h and g with $D^{(1)}$ and $D^{(2)}$, which is shown in equations (2.1.18) and (2.1.19).

$$D^{(1)}(q, t) = h(q, t) \quad (2.1.18)$$

$$D^{(2)}(q, t) = \frac{1}{2} g^2(q, t) \quad (2.1.19)$$

The multi-dimensional version of equation (2.1.15) can be found in equation (2.1.20), where \mathbf{h} and $\Gamma(t)$ are vectors of length d while \mathbf{G} is a $d \times d$ matrix.

$$\dot{\mathbf{q}}(t) = \mathbf{h}(\mathbf{q}(t), t) + \mathbf{G}(\mathbf{q}(t), t) \Gamma(t) \quad (2.1.20)$$

The multi-dimensional $\Gamma(t)$ works under the condition seen in (2.1.21), where δ_{ij} is the Kronecker symbol.

$$\langle \Gamma(t) \Gamma(t') \rangle = \delta_{ij} \delta(t - t') \quad (2.1.21)$$

Individual rows of $\dot{\mathbf{q}}(t)$ can be extracted using equation (2.1.22).

$$\dot{q}_i(t) = h_i(\mathbf{q}(t), t) + \sum_j G_{ij}(\mathbf{q}(t), t) \Gamma_j(t) \quad (2.1.22)$$

The connection to the Kramers-Moyal coefficients in the multi-dimensional case can be found in equations (2.1.23) and (2.1.24).

$$D_i^{(1)}(\mathbf{q}, t) = h_i(\mathbf{q}, t) \quad (2.1.23)$$

$$D_{ij}^{(2)}(\mathbf{q}, t) = \frac{1}{2} \sum_k \mathbf{G}_{ik}(\mathbf{q}, t) \mathbf{G}_{jk}(\mathbf{q}, t) \quad (2.1.24)$$

2.2 Bayesian Estimation of Parameters

The parameter estimation method described below seeks to determine the drift and diffusion coefficients based on Bayesian statistics and works by finding the maximum of the probability that the model describes the data set. The underlying process is Markovian and the dynamic stationary in time.

2.2.1 Bayesian Statistics

Bayesian statistics give probabilities based on the available knowledge.

According to the Bayes' theorem, a conditional probability $p(x|y)$, called posterior, can be described by the likelihood $p(y|x)$, the prior $p(x)$ and the evidence $p(y)$ as seen in equation (2.2.1).

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} \quad (2.2.1)$$

The prior $p(x)$ contains all the current information on the probability of x , prior to involving y . If there is no information a flat prior is used, deeming each outcome equally likely.

The likelihood $p(y|x)$ gives the probability of y under the condition x .

The evidence $p(y)$ primarily functions as a normalizer, which makes it unimportant to the form of the probability distribution.

If the Bayes' theorem is used to extract a model M from data D it has the following form:

$$p(M|D) = \frac{p(D|M)p(M)}{p(D)}$$

Under that condition the prior is the probability of the model without taking the data into account and the likelihood is the probability of the model producing this exact set of data.

A more extensive explanation can be found in reference [2].

2.2.2 Maximum Likelihood Estimation

As described in reference [16] the maximum likelihood estimation (MLE) is used to find a model defined by the parameters $c_j \in \mathbf{c}$ which describes the data set x_0, \dots, x_N .

To that end a conditional probability is used to determined how likely it is for different combinations of parameters to be a good fit for the data set, while also taking the time step τ into consideration. Maximizing this conditional probability leads to the parameter set that best describes the data. It is defined in equation (2.2.2) and derived from the Bayes' theorem as introduced in equation (2.2.1).

$$p(\mathbf{c}|x_N, \dots, x_0; \tau) = \frac{p(x_N, \dots, x_1|x_0; \mathbf{c}, \tau)p(\mathbf{c})}{p(x_N, \dots, x_1|x_0; \tau)} \quad (2.2.2)$$

Without further information a flat prior on a reasonable interval is used, which doesn't affect the position of the maximum. Since the evidence also has no impact on the position of the maximum equation (2.2.2) can be simplified, so that maximizing the posterior is equivalent to maximizing the likelihood, which is depicted in equation (2.2.3).

$$p(\mathbf{c}|x_N, \dots, x_0; \tau) \sim p(x_N, \dots, x_1|x_0; \mathbf{c}, \tau) \quad (2.2.3)$$

$p(x_N, \dots, x_1|x_0; \mathbf{c}, \tau)$ comes from a Markov process, it can therefore be split into a product of conditional probabilities as introduced in equation (2.1.5). Since the logarithm is a strictly monotonic function, applying it means that maximizing the likelihood gives the same result as maximizing $\ln[p(x_N, \dots, x_1|x_0; \mathbf{c}, \tau)]$. The log-likelihood function therefore has the form seen in equation (2.2.4).

$$L(\mathbf{c}) = \sum_{i=1}^N \ln[p(x_i|x_{i-1}; \mathbf{c}, \tau)] \quad (2.2.4)$$

For the conditional probability found in equation (2.2.4) the short-time propagator is the same as for the Fokker-Planck equation (see equation (2.1.10)). The one-dimensional case is shown in equation (2.2.5).

$$L(\mathbf{c}) = -\frac{1}{2} \sum_{i=1}^N \left[\frac{[x_i - x_{i-1} - D^{(1)}(x_{i-1}, \mathbf{c})\tau]^2}{2D^{(2)}(x_{i-1}, \mathbf{c})\tau} + \ln[D^{(2)}(x_{i-1}, \mathbf{c})] + \ln(4\pi\tau) \right] \quad (2.2.5)$$

The $D^{(n)}$ are equations depending on the parameter set \mathbf{c} . An example for a polynomial ansatz for them is shown in equation (2.2.6).

$$D^{(n)}(x, \mathbf{c}) = c_{0,n} + c_{1,n}x + c_{2,n}x^2 + c_{3,n}x^3 + \dots \quad (2.2.6)$$

In case of multiple dimensions the log-likelihood is similarly derived using the short-time propagator seen in equation (2.1.12). Here the \mathbf{x} and $\mathbf{D}^{(1)}$ are vectors of dimension d , while $\mathbf{D}^{(2)}$ is a square matrix of size d . The log-likelihood is depicted in equation (2.2.7).

$$L(\mathbf{c}) = -\frac{1}{2} \sum_{i=1}^N \left[\frac{1}{2\tau} [\mathbf{x}_i - \mathbf{x}_{i-1} - \mathbf{D}^{(1)}\tau][\mathbf{D}^{(2)}]^{-1} [\mathbf{x}_i - \mathbf{x}_{i-1} - \mathbf{D}^{(1)}\tau] + \ln [(4\pi\tau)^d \text{Det}[\mathbf{D}^{(2)}]] \right] \quad (2.2.7)$$

2.2.3 Sparsity Promoting Methods

As described in reference [24] or [5] a lot of models which describe physical processes consist of only a hand full of terms. In a high-dimensional function space the governing equations are therefore sparse.

There are a lot of different methods which promote sparsity when analysing data. All of them, in one way or another, aim to keep the amount of parameters $c_j \neq 0$ small.

Bayesian Information Criterion

The Bayesian information criterion (BIC) as explained in reference [5] is a model selection method which promotes sparsity through a penalty term. The size of the penalty depends on the amount of coefficients $c_j \neq 0$.

Instead of maximizing the likelihood, the BIC, as defined in equation (2.2.8), is minimized.

$$BIC = k \cdot \log(n) - 2 \cdot L_{\max} \quad (2.2.8)$$

Here k is the amount of coefficients $c_j \neq 0$, n is the number of data points and L_{\max} is the result of the maximized log-likelihood function.

The BIC is guaranteed to converge to the correct model in the large data limit, as long as the correct model is part of the tested models.

Thresholding

Another sparsity promoting method is thresholding, following the idea of sequential thresholding which is discussed in subsection 2.3.2.

A thresholding parameter λ is chosen for truncation. All coefficients c_j with $|c_j| < \lambda$ are set to zero during the maximization process.

Hyperparameter Optimization

Hyperparameters are parameters which are part of the model selection process, but not the model itself. Examples for hyperparameters are the thresholding parameter or the weight assigned to a penalty term.

The process of optimization gives the hyperparameter value which leads to the best possible model.

The method used in this thesis is based on Bayesian hyperparameter optimization from the *hyperopt* package in Python (see [3]). It operates by choosing the next hyperparameter as efficiently as possible, while a loss function evaluates how effective the hyperparameter was. From the combination of hyperparameter h and result of the loss function g a probability distribution $p(h|g)$ is formed after a few test runs, which can be described through Bayes' theorem (see equation (2.2.1)). The maximum of the probability distribution gives the best hyperparameter value to choose for the next test, the result of which is then used to update the probability distribution through a new and improved prior.

2.2.4 Markov Chain Monte Carlo Methods

Reference [2] gives a more exhaustive overview over Markov chain Monte Carlo (MCMC) methods, but the key points are explained below.

Monte Carlo methods use sampling from a probability distribution $D(x)$ to approximate its shape. The resulting sample set $\{x_i\}$ can now substitute the actual probability distribution when looking for things like the mean or maximum.

There are different methods of sampling, which mostly work by drawing from an easier distribution $Q(x)$ with $Q(x) \geq D(x)$ and then either keeping or rejecting the samples according to $D(x)$. A more efficient way of sampling which draws preferably from regions with high probability are the Markov chain Monte Carlo methods.

MCMC methods use a random walk on a Markov chain, therefore choosing the next sample x_{i+1} according to the short-time propagator $p(x_{i+1}|x_i)$. For this to work it is necessary that every part of the probability distribution is reachable from other parts of the parameter space.

The MCMC method used in this thesis is the affine-invariant MCMC ensemble sampler from the *emcee* package for Python, as documented in reference [10].

2.3 Sparse Identification of Nonlinear Dynamics

Given a (potentially noisy) data set $(\dot{\mathbf{X}}, \mathbf{X})$ of a (potentially nonlinear) dynamical system, where \mathbf{X} and $\dot{\mathbf{X}}$ are the time series data of the state $\mathbf{x}(t)$ and its time derivative $\dot{\mathbf{x}}(t)$ respectively (see equation (2.3.1) and (2.3.2)), the 'Sparse Identification of Nonlinear Dynamics', or SINDy, (as proposed in references [7] and [24] for ordinary and partial differential equations respectively) seeks to determine the underlying governing equations.

$$\mathbf{X} = \begin{pmatrix} x_1(t_1) & x_2(t_1) & \cdots & x_n(t_1) \\ x_1(t_2) & x_2(t_2) & \cdots & x_n(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ x_1(t_m) & x_2(t_m) & \cdots & x_n(t_m) \end{pmatrix} \quad (2.3.1)$$

$$\dot{\mathbf{X}} = \begin{pmatrix} \dot{x}_1(t_1) & \dot{x}_2(t_1) & \cdots & \dot{x}_n(t_1) \\ \dot{x}_1(t_2) & \dot{x}_2(t_2) & \cdots & \dot{x}_n(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ \dot{x}_1(t_m) & \dot{x}_2(t_m) & \cdots & \dot{x}_n(t_m) \end{pmatrix} \quad (2.3.2)$$

SINDy works under the assumption that the resulting model is sparse in the space of possible functions and can therefore be described by only a few terms. To achieve this it uses a combination of machine learning and sparsity promoting regression methods.

For a dynamical system of the form $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), t)$, where $\mathbf{x}(t)$ represents the state of the system at the time t and $\mathbf{f}(\mathbf{x}(t), t)$ describes the dynamic of the system, a library $\Theta(\mathbf{X}(t))$ of possible building blocks of $\mathbf{f}(\mathbf{X}(t), t)$ in the form of a matrix is constructed from the time series data. An example with polynomials, trigonometric functions and a constant is shown in equation (2.3.3).

$$\Theta(\mathbf{X}) = \begin{pmatrix} | & | & | & \cdots & | & | \\ 1 & \mathbf{X} & \mathbf{X}^{P_2} & \cdots & \sin(\mathbf{X}) & \cos(\mathbf{X}) \\ | & | & | & & | & | \end{pmatrix} \quad (2.3.3)$$

\mathbf{X}^{P_2} represents all the different terms of the second order polynomial as shown in equation (2.3.4).

$$\mathbf{X}^{P_2} = \begin{pmatrix} x_1^2(t_1) & x_1(t_1)x_2(t_1) & \cdots & x_2^2(t_1) & \cdots & x_n^2(t_1) \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ x_1^2(t_m) & x_1(t_m)x_2(t_m) & \cdots & x_2^2(t_m) & \cdots & x_n^2(t_m) \end{pmatrix} \quad (2.3.4)$$

The dynamical system $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), t)$ can now be represented by equation (2.3.5) or rather equation (2.3.6) if we consider a noisy data set. Ξ consists of n columns

of (sparse) vectors which weight the influence of the accompanying building blocks in the library matrix. Γ represents a noise matrix with the noise magnitude η .

$$\dot{\mathbf{X}} = \Theta(\mathbf{X})\Xi \quad (2.3.5)$$

$$\dot{\mathbf{X}} = \Theta(\mathbf{X})\Xi + \eta\Gamma \quad (2.3.6)$$

Equation (2.3.5) can now be viewed as a linear equation and therefore be solved using linear regression, where each column of Ξ requires its own regression. Depending on the data set it might also be necessary to normalize the columns of the library matrix (see [29]) otherwise small elements in polynomials can become so small they show no impact during the regression.

2.3.1 SINDy for Partial Differential Equations

A partial differential equation can be described as seen in equation (2.3.7), therefore knowledge of the spatial derivatives is needed to build a complete library matrix of candidate functions.

$$\partial_t \mathbf{u}(x, t) = \mathbf{f}(\mathbf{u}(x, t), \Delta \mathbf{u}(x, t), \dots) \quad (2.3.7)$$

If the field is homogeneous in its behaviour the amount of data can be reduced by sampling the data only at certain points of the field.

From there on the partial differential equation can be used just like an ordinary differential equation.

2.3.2 Regression Methods

There are different regression methods proposed for SINDy ([7], [24], [19]). This thesis focuses on four of them.

Ordinary Least Squares Regression

The ordinary least squares regression fits a linear model of the form $y = f(x, \alpha) = x \cdot \alpha$ to data points. The corresponding cost function has the form $c(\alpha) = \frac{1}{2} \sum_i (y_i - f(x_i, \alpha))^2$ and is minimized using $\frac{\partial c}{\partial \alpha} = 0$ [9].

For use in the SINDy algorithm, α is considered a vector and x a matrix of candidate functions. This regression method is not inherently sparse, but can still work, depending on the data set and the possible candidate functions.

LASSO Regression

The LASSO (least absolute shrinkage and selection operator) regression, as proposed in reference [27], works a lot like the ordinary least squares regression, but to prevent overfitting, which causes very large coefficients, the cost function gets an extra penalty term to discourage large parameter values.

The cost function now has the form $c(\alpha) + \lambda \sum_j |\alpha_j|$ where the Lagrange parameter λ dictates how harsh the penalty is.

The LASSO regression penalizes every $\alpha_j \neq 0$ and therefore promotes sparsity.

Sequential Thresholded Least-Squares

The sequential thresholded least-squares (STLS) method (see [7]) uses the ordinary least squares regression in combination with a threshold value λ to promote sparsity.

An initial linear regression gives a set of coefficients α_j . Every $|\alpha_j| < \lambda$ is then set to zero. Another least-squares solution with the remaining non-zero parts of the problem is obtained and the result once again thresholded. Repeating this eventually allows a hand full of remaining coefficients to converge.

Iterative Hard-Thresholding

The iterative hard-thresholding (IHT) (see [4]) also works with a thresholding value λ , but unlike the sequential thresholded least-squares method it doesn't work by minimizing a cost function.

The iterative hard-thresholding starts with $\alpha = \vec{0}$ and then iterates, using $\alpha_{n+1} = \alpha_n + x^T(y - x\alpha_n)$. Much like the sequential thresholded least-squares method, every component of α that's smaller than the threshold value is set to zero. As long as $\|x\|_2 < 1$ this method eventually converges to a fixed point.

3 Stochastic Time Series Analysis for Spatio-Temporal Data

3.1 Stochastic Spatio-Temporal Systems

Systems governed by a stochastic partial differential equations (SPDE) can also take the form of a Langevin equation as introduced in reference [15].

Taking equation (2.1.19) into account, the Langevin equation for SPDEs takes the form seen in equation (3.1.1).

$$\partial_t u(\mathbf{x}, t) = D^{(1)}[u(\mathbf{x}, t)] + \sqrt{2D^{(2)}[u(\mathbf{x}, t)]}\Gamma(\mathbf{x}, t) \quad (3.1.1)$$

Here $u(\mathbf{x}, t)$ is a state variable on a spatially extended field, which does not depend explicitly on the point \mathbf{x} . Instead that dependence stems from spatial derivatives.

3.2 Spatio-Temporal Short-Time Propagator

An incremental change in equation (3.1.1) leads to equation (3.2.1) [17].

$$u_{i+1} - u_i = D^{(1)}\tau + \sqrt{2D^{(2)}}\Gamma_i \quad (3.2.1)$$

Here Γ_i is an incremental change of a stochastic Gaussian process (or Wiener process), therefore it is normally distributed noise with zero mean (see subsection (2.1.5)) and a standard deviation which depends linearly on $\sqrt{\tau}$ [25]. According to reference [18] equation (3.2.2) can be used to the same effect.

$$\Gamma_i = \sqrt{\tau}N(0, 1) \quad (3.2.2)$$

Applying equation (3.2.2) to equation (3.2.1) leads to equation (3.2.3), where μ and σ are the mean and standard deviation for u_{i+1} .

$$u_{i+1} = \underbrace{u_i + D^{(1)}\tau}_{\mu} + \underbrace{\sqrt{2D^{(2)}}\tau}_{\sigma}N(0, 1) \quad (3.2.3)$$

Following from equation (3.2.3) the short-time propagator is derived in equation (3.2.4) with the end result shown in equation (3.2.5) which matches the case for ordinary differential equations seen in equation (2.1.10).

$$p(u_{i+1}|u_i) \sim N(\mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(u_{i+1} - \mu)^2}{2\sigma^2}\right) \quad (3.2.4)$$

$$= \frac{1}{\sqrt{4\pi D^{(2)}\tau}} \exp\left(-\frac{(u_{i+1} - u_i + D^{(1)}\tau)^2}{4D^{(2)}\tau}\right) \quad (3.2.5)$$

3.3 Spatio-Temporal Maximum Likelihood Estimation

The SPDE equivalent for finding a model to describe a data set through Bayes (as seen in equation (2.2.3)) is depicted in equation (3.3.1), where maximizing the likelihood (right-hand side) maximizes the posterior (left-hand side), therefore giving the best model with parameter set \mathbf{c} to describe the data set u_0, \dots, u_N .

$$p(\mathbf{c}|u_N, \dots, u_0; \tau) \sim p(u_N, \dots, u_1|u_0; \mathbf{c}, \tau) \quad (3.3.1)$$

Applying the Markov chain according to equation (2.1.5) gives the likelihood seen in equation (3.3.2).

$$p(\mathbf{c}|u_N, \dots, u_0; \tau) \sim \prod_{i=0}^{N-1} p(u_{i+1}|u_i; \mathbf{c}, \tau) \quad (3.3.2)$$

As established in section 2.2.2 maximizing the likelihood is the same as maximizing the log-likelihood, which can be found in equation (3.3.3). Adding the short-time propagator from equation (3.2.5) gives the finalized log-likelihood as seen in equation (3.3.4).

$$L(\mathbf{c}) = \sum_{i=1}^N \ln[p(u_i|u_{i-1}; \mathbf{c}, \tau)] \quad (3.3.3)$$

$$= \sum_{i=1}^N \left[-\frac{(u_i - u_{i-1} - D^{(1)}(\mathbf{c})\tau)^2}{4D^{(2)}(\mathbf{c})\tau} - \frac{1}{2} \ln(4D^{(2)}(\mathbf{c})\pi\tau) \right] \quad (3.3.4)$$

3.4 Multi-Dimensional Maximum Likelihood Estimation

Following the same steps as in subsection 3.2 for the multi-dimensional case leads to the short-time propagator depicted in equation (3.4.1), which also matches with the

one for multi-dimensional ordinary differential equations (see equation (2.1.12)).

$$p(\mathbf{u}_i | \mathbf{u}_{i-1}) = \frac{\exp\left(-\frac{1}{4\tau}[\mathbf{u}_i - \mathbf{u}_{i-1} - \mathbf{D}^{(1)}(\mathbf{c})\tau][\mathbf{D}^{(2)}(\mathbf{c})]^{-1}[\mathbf{u}_i - \mathbf{u}_{i-1} - \mathbf{D}^{(1)}(\mathbf{c})\tau]\right)}{(2\sqrt{\pi\tau})^d \sqrt{\text{Det}[\mathbf{D}^{(2)}(\mathbf{c})]}} \quad (3.4.1)$$

For the multivariate case the drift and diffusion coefficients take the forms seen in equations (3.4.2), with the accompanying log-likelihood seen in equation (3.4.3).

$$\mathbf{u} = \begin{pmatrix} u \\ v \\ \vdots \end{pmatrix} \quad \mathbf{D}^{(1)} = \begin{pmatrix} D_u^{(1)} \\ D_v^{(1)} \\ \vdots \end{pmatrix} \quad \mathbf{D}^{(2)} = \begin{pmatrix} D_{uu}^{(2)} & D_{uv}^{(2)} & \dots \\ D_{vu}^{(2)} & D_{vv}^{(2)} & \dots \\ \vdots & \vdots & \ddots \end{pmatrix} \quad (3.4.2)$$

$$L(\mathbf{c}) = -\frac{1}{2} \sum_{i=1}^N \left[\frac{[\mathbf{u}_i - \mathbf{u}_{i-1} - \mathbf{D}^{(1)}(\mathbf{c})\tau][\mathbf{D}^{(2)}(\mathbf{c})]^{-1}[\mathbf{u}_i - \mathbf{u}_{i-1} - \mathbf{D}^{(1)}(\mathbf{c})\tau]}{2\tau} + \ln[(4\pi\tau)^d \text{Det}[\mathbf{D}^{(2)}(\mathbf{c})]] \right] \quad (3.4.3)$$

The easier two-dimensional case where $\mathbf{D}^{(2)}(\mathbf{c})$ is a diagonal matrix gives the coefficients seen in equation (3.4.4) and the log-likelihood seen in equation (3.4.5).

$$\mathbf{u} = \begin{pmatrix} u \\ v \end{pmatrix} \quad \mathbf{D}^{(1)} = \begin{pmatrix} D_u^{(1)} \\ D_v^{(1)} \end{pmatrix} \quad \mathbf{D}^{(2)} = \begin{pmatrix} D_u^{(2)} & 0 \\ 0 & D_v^{(2)} \end{pmatrix} \quad (3.4.4)$$

$$L(\mathbf{c}) = -\frac{1}{2} \sum_{i=1}^N \left[\frac{[u_i - u_{i-1} - D_u^{(1)}(\mathbf{c})\tau]^2}{2D_u^{(2)}(\mathbf{c})\tau} + \frac{[v_i - v_{i-1} - D_v^{(1)}(\mathbf{c})\tau]^2}{2D_v^{(2)}(\mathbf{c})\tau} + \ln[D_u^{(2)}(\mathbf{c}) \cdot D_v^{(2)}(\mathbf{c}) \cdot (4\pi\tau)^2] \right] \quad (3.4.5)$$

3.5 The Maximum Likelihood Estimation Program

The maximum likelihood estimation is programmed in Python. The log-likelihood is estimated for every point in the field and then the mean taken, so that the maximization gives the best possible parameters for as many points of the field as possible.

The maximizing is done by minimizing the negative log-likelihood with local or global minimization methods from the Python package *scipy.optimize*.

The local minimization method is called *fmin* and uses the Nelder-Mead method of minimization. It has $d+1$ test points for a d dimensional problem, which define a simplex in the function space. The worst test point is then consecutively replaced by a point on a line projected through the geometrical centre of the simplex so that it has a smaller value than the previous point. When reaching the minimum the test points move progressively closer together until they converge [20].

The global minimization method is called *differential_evolution*. It uses several start vectors spread out over the function space. Those vectors are mutated through a weighted combination of three of the other vectors. Finally the resulting trial vector is compared to the previous vector and the worse option discarded [26].

4 Identifying Models of Generated Spatio-Temporal Data Sets

4.1 1D Reaction-Diffusion Equation

4.1.1 Generating the Data Sets

The reaction-diffusion equation used in this thesis can be found in equation (4.1.1).

$$\partial_t u(\mathbf{x}, t) = \epsilon \Delta u(\mathbf{x}, t) + u(\mathbf{x}, t) - u^3(\mathbf{x}, t) + \sqrt{q_1 + q_2 u^2(\mathbf{x}, t)} \Gamma(\mathbf{x}, t) \quad (4.1.1)$$

The spatio-temporal time series data sets used in the following sections are generated with the diffusion coefficient $\epsilon = 0.25$ and the magnitudes q_1 and q_2 for the dynamic noise $\Gamma(\mathbf{x}, t) = \sqrt{2\tau}N(0, 1)$. The computational grid has 256^2 points with a length $L = 100$ in x_1 - and x_2 -direction. The time interval is from $T_0 = 0$ to $T_n = 1$ with the time step $\tau = 0.01$, giving it $n = 100$ time steps after a noiseless burn-in period. A short overview over results with other time steps can be found in the appendix in table A.1.

The time steps are calculated using a pseudo-spectral method (see [8]), where the data and the linear part of the equation are moved into the Fourier space using a fast Fourier transformation. The nonlinear parts of the equation are calculated in real space, before also being transformed. The resulting equation is then solved through discretisation with the Runge-Kutta 4 method (see [13]) and the newly generated data moved back to real space, where noise can be added.

Snapshots of data sets with different noise levels can be found in figure 4.1.

4.1.2 Results of the Maximum Likelihood Estimation

Different Noise Levels at a Minimum of Parameters

Giving the maximum likelihood estimation the minimal option of parameters (meaning only the ones actually contained in the equation) as seen in equations (4.1.2) and (4.1.3) while varying the noise levels gives the results detailed in table 4.1. The

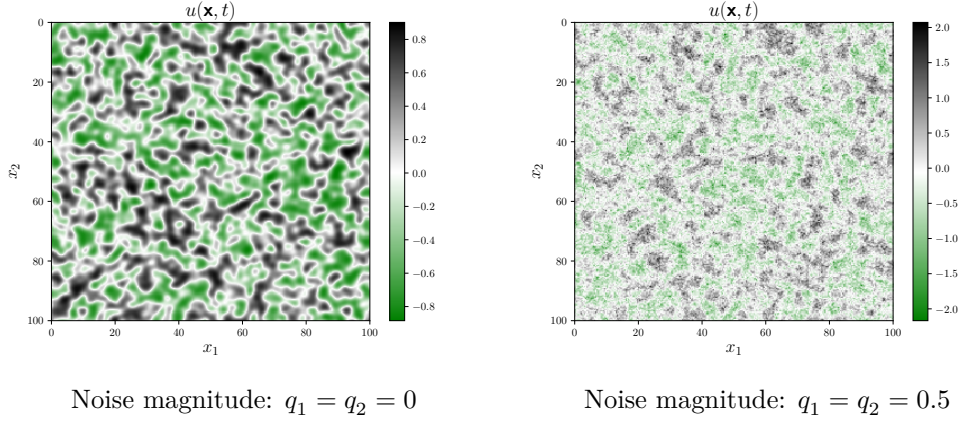


Figure 4.1: Snapshots of 1D reaction-diffusion data sets with different noise levels.

minimization used here is the global minimization, followed by a local one, once the general position of the minimum is found.

$$D^{(1)}(\mathbf{a}) = a_1 u(\mathbf{x}, t) + a_2 \Delta u(\mathbf{x}, t) + a_3 u^3(\mathbf{x}, t) \quad (4.1.2)$$

$$D^{(2)}(\mathbf{b}) = b_1 + b_2 u^2(\mathbf{x}, t) \quad (4.1.3)$$

For a data set without noise the maximum likelihood estimation shows near perfect

q_1	q_2	a_1	a_2	a_3	b_1	b_2	δ
0	0	0.997	0.249	-0.997	0	0	0.3%
0.5	0	0.953	0.229	-0.955	0.501	-0.001	4.5%
1	0	0.909	0.229	-0.931	1.000	-0.001	6.6%
0.5	0.5	0.924	0.230	-0.921	0.509	0.474	7.2%
1	1	0.848	0.229	-0.870	1.021	0.924	10.7%

Table 4.1: Maximum likelihood estimation results for 1D reaction-diffusion data sets with different noise levels and a minimum of parameters according to equations (4.1.2) and (4.1.3), as well as the percent error δ .

results. For noisy data sets only the data set with $q_1 = q_2 = 1$ leads to a result with a relative error over ten percent to the expected result. All other noise levels show parameter values which closely match the expected ones.

For comparison the SINDy method is applied to the same data sets, using a sample of 400 data points and ordinary least squares regression, as there is no need for

sparsity with a minimum of possible parameters. The results can be found in table 4.2. The percent error to the real parameter values for both the maximum likelihood estimation and the SINDy method are listed in table 4.3.

q_1	q_2	a_1	a_2	a_3
0	0	1	0.25	-1
0.5	0	0.919	0.221	-0.909
1	0	0.789	0.238	-0.833
0.5	0.5	0.846	0.226	-1.029
1	1	0.644	0.228	-0.713

Table 4.2: SINDy results with ordinary least squares regression for 1D reaction-diffusion data sets with different noise levels and a minimum of parameters according to equation (4.1.2).

q_1	q_2	δ_{SINDy}	δ_{MLE}
0	0	0.0 %	0.3 %
0.5	0	8.7 %	4.8 %
1	0	18.8 %	8.1 %
0.5	0.5	11.0 %	7.8 %
1	1	31.9 %	14.0 %

Table 4.3: Percent error δ for the $D^{(1)}$ results of the maximum likelihood estimation and the SINDy method for 1D reaction-diffusion data sets with different noise levels as found in tables 4.1 and 4.2.

For noisy data the results obtained through SINDy deviate much further from the expected values than the ones from the maximum likelihood. In fact only the data set with $q_1 = 0.5$ and $q_2 = 0$ leads to a percent error under ten percent. The data set without noise on the other hand shows marginally better results with the SINDy method.

Excess Amount of Parameters at Constant Multiplicative Noise

At a set noise level of $q_1 = q_2 = 1$ a full set of parameters up to the third polynomial of $1, u, \Delta u$ and $\Delta^2 u$ is tested via local minimization, where all starting values for the parameters are set to 0. At twenty-two parameters minimizing the negative log-likelihood only once isn't enough to get a useful result; instead the result of the previous minimization is set as the new starting value. This process is repeated for several iterations with the final results shown in table 4.4.

The results for twenty-one iterations have a relative error of $\delta = 10.6\%$ to the expected values. For thirty-one iterations the relative error is $\delta = 7.2\%$.

Running twenty-one or even thirty-one iterations of a twenty-two parameter minimization takes some time, but the final results can be considered pretty good, especially for thirty-one iterations which lead to a relative error of less than ten percent.

Applying the SINDy method with the same set of possible parameters while using a sample of 400 data points leads to the result seen in table 4.5. For the regression the iterative hard-thresholding is used.

The expected level of sparsity can be reached with thresholds between 0.13 and 0.17 and the result is reached much faster than with the maximum likelihood estimation method, once a suitable threshold is found, but the resulting parameter values are further from the expected ones with a percent error of 32.1 % to the real parameter values.

Excess Amount of Parameters at Constant Multiplicative Noise with Thresholding and Hyperparameter Optimization

The thresholding method is applied to a data set with the noise $q_1 = q_2 = 1$ while all starting values in the local minimization are set to 0.5 (as to not start under the likelier thresholding values). The optimal threshold λ is derived through hyperparameter optimization, the full results of which can be found in the appendix in tables A.4, A.5 and A.6. The important parts are summarized in tables 4.6 and 4.7.

Table 4.6 shows that one iteration of maximizing the likelihood without a threshold isn't enough to give useful results, while thresholds over $\lambda = 0.507$ cause all parameters to go to 0, most likely because the starting values are too far below the threshold for the the minimization to find better values.

$\lambda = 0.012$ comes pretty close to the expected result, but the first threshold with the optimal log-likelihood is $\lambda = 0.0236$, though the difference between their negative log-likelihood values $-L(\mathbf{c})$ is very small.

An overview showing which of the other thresholds give optimal results can be found in table 4.7. The optimal thresholds are marked in gray and it can be seen that there is no range of optimal threshold sizes, which makes finding them without the optimization process somewhat difficult.

Applying the thresholding method with $\lambda = 0.1$ to a parameter set up to the third polynomial and then reducing the number of possible parameters with every parameter that is truncated through thresholding gives the results seen in table 4.8. The minimization method is local, starting with all parameter values at 0.5.

It takes only five iterations of minimization in total to reach the expected result, less than half of the amount it takes without thresholding.

Excess Amount of Parameters at Constant Multiplicative Noise with BIC and Thresholding

Applying the BIC to a system with a noise level of $q_1 = q_2 = 1$ and then optimizing the threshold λ for a minimum BIC value with starting values of 0.5 for the local minimization leads to the results seen in table A.7. The important parts are summarized in table 4.9.

The application of the BIC without any thresholds shows better results than not using any sparsity promoting methods.

Applying the right threshold in addition to the BIC can set the expected parameters to zero as depicted in the second to last column in table 4.9. The last column shows the result if the system is optimized for the ideal thresholding hyperparameter with the smallest possible BIC value. Most parameter values are set to zero, including some that are part of the correct model.

In conclusion, hyperparameter optimization for thresholding is not guaranteed to give ideal results when combined with the BIC.

4.1.3 Probability Distributions

The following probability distributions are calculated using Markov chain Monte Carlo with only the minimum of parameters. Ten walkers are used, which each draw 5000 samples after a burn-in period of 100 sample draws. The prior is set to keep the parameter values within $-50 < a_i < 50$ and the noise positive.

The resulting histograms (see figure 4.2) show relatively clear peaks which fit with the expected parameter values, with only some small disturbances. Those disturbances explain the differences between the maximum likelihood estimation results and the expected results.

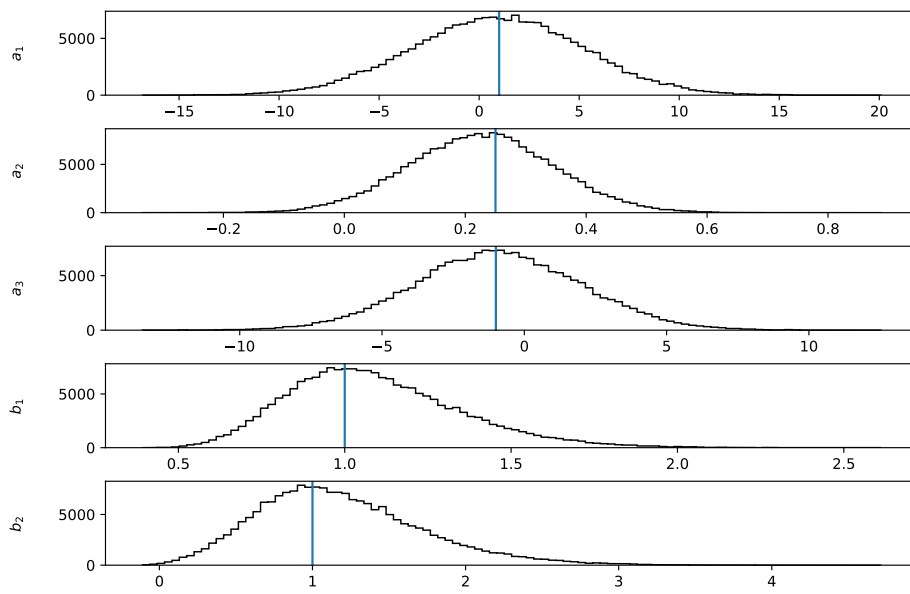


Figure 4.2: Histograms of the probability distributions for the parameters of the 1D reaction-diffusion equation, where the blue lines represent the real parameter values.

	real values	21 iterations	31 iterations
b_1	1	1.021	1.021
b_2	1	0.924	0.924
u	1	0.833	0.901
Δu	0.25	0.231	0.249
$\Delta^2 u$	0	0	0
u^2	0	0.045	0.035
$u\Delta u$	0	0.002	0
$u\Delta^2 u$	0	0	0
$(\Delta u)^2$	0	0	0
$(\Delta u)(\Delta^2 u)$	0	0	0
$(\Delta^2 u)^2$	0	0	0
u^3	-1	-0.911	-0.937
$u^2\Delta u$	0	0.013	-0.005
$u^2\Delta^2 u$	0	0	0
$u(\Delta u)^2$	0	0.003	0.001
$u(\Delta u)(\Delta^2 u)$	0	0	0
$u(\Delta^2 u)^2$	0	0	0
$(\Delta u)^3$	0	0	0
$(\Delta u)^2(\Delta^2 u)$	0	0	0
$(\Delta u)(\Delta^2 u)^2$	0	0	0
$(\Delta^2 u)^3$	0	0	0
1	0	-0.029	-0.006

Table 4.4: Maximum likelihood estimation results for a 1D reaction-diffusion data set with the noise level $q_1 = q_2 = 1$, possible parameters up to the third polynomial of 1, u , Δu and $\Delta^2 u$ and different numbers of iterations of minimization.

u	Δu	u^3
0.636	0.227	-0.718

Table 4.5: SINDy results with iterative hard-thresholding for a 1D reaction-diffusion data set with a noise level of $q_1 = q_2 = 1$ and possible parameters up to the third polynomial of 1, u , Δu and $\Delta^2 u$. Only parameters $\neq 0$ are listed.

λ	0.0000	0.0120	0.0236	0.5070	0.5319
$-L(\mathbf{c})$	-33.31163	-33.62259	-33.62264	26.11632	inf
u	-0.34	0.84	0.86	0	0
Δu	0.29	0.23	0.23	0	0
u^2	1.14	-0.01	0	0	0
$u\Delta u$	0.03	0	0	0	0
$(\Delta u)^2$	0	0	0	0	0
u^3	0.61	-0.86	-0.88	0	0
$\Delta^2 u$	0	0	0	0	0
1	0.63	0	0	0	0
q_1	1.02	1.02	1.02	0.53	0
q_2	0.95	0.93	0.93	0	0

Table 4.6: Excerpts of the results of the hyperparameter optimization for the thresholding parameter λ applied to a 1D reaction-diffusion data set with ten parameters and one iteration of minimization.

0	0.0228	0.0382	0.0572	0.0949	0.1469	0.2553	0.5319
0.0117	0.0236	0.0395	0.0574	0.1000	0.1586	0.2674	0.5942
0.0120	0.0262	0.0414	0.0639	0.1018	0.1613	0.2912	0.6242
0.0127	0.0263	0.0425	0.0640	0.1098	0.1717	0.3009	0.6893
0.0129	0.0267	0.0431	0.0715	0.1122	0.1734	0.3094	0.7704
0.0146	0.0280	0.0435	0.0737	0.1139	0.1778	0.3490	0.7992
0.0158	0.0300	0.0483	0.0743	0.1212	0.1792	0.3510	0.9850
0.0161	0.0302	0.0488	0.0802	0.1234	0.1959	0.3758	0.9869
0.0165	0.0319	0.0489	0.0812	0.1271	0.2020	0.4166	0.4477
0.0189	0.0336	0.0504	0.0849	0.1313	0.2057	0.4189	0.4624
0.0193	0.0350	0.0520	0.0871	0.1321	0.2208	0.4477	0.5070
0.0203	0.0359	0.0537	0.0897	0.1416	0.2238	0.4624	0.5319
0.0206	0.0364	0.0571	0.0931	0.1424	0.2329	0.5070	

Table 4.7: All tested thresholds from the hyperparameter optimization applied to a 1D reaction-diffusion data set with ten parameters and one iteration of minimization. Optimal thresholds are marked in gray.

	real values	1 iteration	2 iterations	2 iterations
b_1	1	0.53	1.10	1.02
b_2	1	0.49	1.19	0.93
u	1	0.44	0.51	0.87
Δu	0.25	0.40	0.63	0.23
$\Delta^2 u$	0	0.94	0	-
u^2	0	0.70	0.70	0
$u\Delta u$	0	1.05	0.70	0
$u\Delta^2 u$	0	0.95	0.70	0
$(\Delta u)^2$	0	0.82	0.65	0
$(\Delta u)(\Delta^2 u)$	0	0.78	0	-
$(\Delta^2 u)^2$	0	0	-	-
u^3	-1	0.53	0.57	-0.89
$u^2(\Delta u)$	0	0.72	0.72	0
$u^2(\Delta^2 u)$	0	0.75	0.50	0
$u(\Delta u)^2$	0	0.77	0.81	0
$u(\Delta u)(\Delta^2 u)$	0	0.72	0.17	0
$u(\Delta^2 u)^2$	0	-0.16	0	-
$(\Delta u)^3$	0	0.53	-0.19	0
$(\Delta u)^2(\Delta^2 u)$	0	0.37	0	-
$(\Delta u)(\Delta^2 u)^2$	0	0	-	-
$(\Delta^2 u)^3$	0	0	-	-
1	0	0	-	-

Table 4.8: Maximum likelihood estimation results for a 1D reaction-diffusion data set with thresholding at $\lambda = 0.1$, different numbers of iterations of minimization and a shrinking amount of possible parameters according to the previous result.

method	no BIC	BIC	BIC	BIC
λ	0.00	0.00	0.01	0.19
BIC	-	-33.6	-44.2	-53.3
u	-0.34	0.88	0.86	0
Δu	0.29	0.25	0.23	0.24
u^2	1.14	0.04	0	0
$u\Delta u$	0.03	0	0	0
$(\Delta u)^2$	0	0	0	0
u^3	0.61	-0.85	-0.88	0
$\Delta^2 u$	0	0	0	0
1	0.63	-0.02	0	0
q_1	1.02	1.02	1.02	1.02
q_2	0.95	0.93	0.93	0.93

Table 4.9: Excerpt of the results of the hyperparameter optimization for the thresholding parameter λ applied to a 1D reaction-diffusion data set with the BIC as cost function, as well as a result without the BIC.

4.2 Swift-Hohenberg Equation

4.2.1 Generating the Data Sets

The Swift-Hohenberg equation used in this section is depicted in equation (4.2.1). With the right parameter r it shows the pattern formation of stripes [11].

$$\partial_t u(\mathbf{x}, t) = ru(\mathbf{x}, t) - (\Delta + 1)^2 u(\mathbf{x}, t) - u^3(\mathbf{x}, t) + \sqrt{q_1 + q_2 u^2(\mathbf{x}, t)} \Gamma(\mathbf{x}, t) \quad (4.2.1)$$

The spatio-temporal time series data sets used in the following sections are generated with $r = 0.3$ and the magnitudes q_1 and q_2 for the dynamic noise $\Gamma(\mathbf{x}, t) = \sqrt{2\tau}N(0, 1)$. The computational grid has 128^2 points with a length $L = 120$ in x_1 - and x_2 -direction. The time interval is from $T_0 = 0$ to $T_n = 0.01$ with the time step $\tau = 0.0001$, giving it $n = 100$ time steps after a noiseless burn-in period. A short overview over results with other integration parameters can be found in the appendix in table A.2.

The time steps are calculated using a pseudo-spectral method (see [8]) in combination with an implicit-explicit Euler method (see [1]). The data and equation are first moved into the Fourier space. Then Euler is applied, where the linear part of the equation is treated implicitly, while the nonlinear part is treated explicitly. Finally the equation is reshaped so that only the data for the next time step is on the left-hand side of the equation. The newly generated data is then moved back to real space where noise can be added.

Snapshots of data sets with and without noise after a noiseless burn-in period can be found in figure 4.3.

4.2.2 Results of the Maximum Likelihood Estimation

Different Noise Levels at a Minimum of Parameters

Running the maximum likelihood estimation with only those parameters that are part of the equation as seen in (4.2.2) and (4.2.3) while varying the noise level leads to the results detailed in table 4.10. The minimization method is local with starting values of zero for all parameters.

$$D^{(1)}(\mathbf{a}) = a_1 u(\mathbf{x}, t) + a_2 \Delta u(\mathbf{x}, t) + a_3 \Delta^2 u(\mathbf{x}, t) + a_4 u^3(\mathbf{x}, t) \quad (4.2.2)$$

$$D^{(2)}(\mathbf{b}) = b_1 + b_2 u^2(\mathbf{x}, t) \quad (4.2.3)$$

As with the one-dimensional reaction-diffusion equation the result for a data set

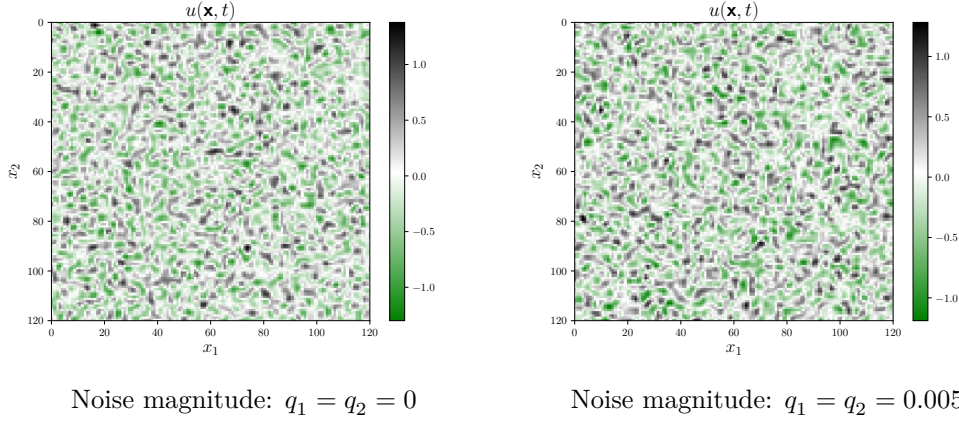


Figure 4.3: Snapshots of Swift-Hohenberg data sets with different noise levels.

without noise is a near perfect match for the expected values.

The noise levels the data sets can exhibit before the maximum likelihood estimation results differ more than ten percent from the expected result are a lot smaller than they were for the one-dimensional reaction-diffusion equation. The highest noise levels with good results are for data sets with $q_1 = q_2 = 0.01$ and with $q_1 = 0.05$ and $q_2 = 0$ respectively.

Applying the SINDy method to the same data sets while using a sample of 400 data points and ordinary least squares regression leads to the results seen in table 4.11.

The results for the maximum likelihood estimation for noisy data sets are a lot closer to the expected results than the ones from the SINDy method. Without noise the methods are nearly equal, though SINDy is faster.

Excess Amount of Parameters at Constant Multiplicative Noise with Thresholding

At a set level of multiplicative noise of $q_1 = q_2 = 0.01$ a full set of parameters up to the third polynomial of $1, u, \Delta u$ and $\Delta^2 u$ is tested via local minimization, where all starting values for the parameters are set to 0. It is also tested with a threshold of $\lambda = 0.05$ and starting values of 0.5 for all parameters. The parameters for the noise levels are exempt from truncation.

q_1	q_2	a_1	a_2	a_3	a_4	b_1	b_2	δ
0	0	-0.699	-1.997	-0.999	-1.000	0.000	0.000	0.1 %
0.001	0	-0.682	-1.968	-0.989	-0.989	0.001	0.000	1.6 %
0.005	0	-0.649	-1.961	-0.993	-0.990	0.005	0.000	2.6 %
0.01	0	-0.683	-1.895	-0.979	-1.041	0.010	0.000	4.6 %
0.05	0	-0.719	-1.941	-0.981	-0.859	0.050	0.000	6.1 %
0.1	0	-0.480	-1.995	-0.978	-1.368	0.100	0.000	16.8 %
0.001	0.001	-0.709	-1.978	-0.992	-0.989	0.001	0.001	1.1 %
0.005	0.005	-0.696	-1.936	-0.980	-1.015	0.005	0.005	2.7 %
0.01	0.01	-0.573	-1.855	-0.970	-1.059	0.010	0.010	8.0 %
0.05	0.05	-1.021	-2.048	-0.994	-0.626	0.050	0.050	19.4 %
0.1	0.1	-0.718	-1.851	-0.976	-0.715	0.100	0.101	12.7 %

Table 4.10: Maximum likelihood estimation results for Swift-Hohenberg data sets with varying noise levels and a minimum of parameters according to equations (4.2.2) and (4.2.3), as well as the percent error δ .

Minimizing the negative log-likelihood only once with or without thresholding is not enough to find a minimum. Instead the result of the previous minimization is set as the new starting value, which is then repeated for several iterations. The results are shown in table 4.12.

The maximum likelihood estimation without thresholding reaches its minimum after eighteen iterations, but doesn't give the expected result.

If thresholding is added some of the parameters which are part of the correct result are truncated and the minimization process is unable to correct in further iterations. To prevent this normal minimization and thresholded minimization are alternated, allowing for the reversal of incorrect truncations during the normal minimization.

Adding this alternating thresholding leads to the expected results. That it works better than the normal minimization is to be expected because small changes in parameter values which were not originally part of the equation are then compensated through other additional parameters. As wrong parameters are set to zero, there is less need for compensation, leaving only the correct result.

Using an excess amount of parameters with thresholding leads to a percent error of $\delta = 2.1\%$, which is better than the result for the minimum amount of possible parameters with the same noise level.

Applying the SINDy method to the same problem, again with a sample size of 400, gives the results found in table 4.13. There are no good results for either of

q_1	q_2	a_1	a_2	a_3	a_4
0	0	-0.698	-1.996	-0.999	-1
0.001	0	-0.349	-1.973	-1.011	-1.377
0.005	0	-0.371	-1.632	-0.966	-0.964
0.01	0	-0.921	-1.824	-0.973	-0.039
0.05	0	-1.248	-2.512	-1.111	0.708
0.1	0	-1.763	-1.984	-0.875	-1.892
0.001	0.001	-0.77	-1.951	-0.971	-0.779
0.005	0.005	-1.422	-2.77	-1.175	-0.572
0.01	0.01	-0.329	-1.651	-0.908	-2.197
0.05	0.05	-0.832	-2.077	-0.998	-1.383
0.1	0.1	1.019	-1.406	-0.973	-2.813

Table 4.11: SINDy results with ordinary least squares regression and a minimum of parameters according to equation (4.2.2) for Swift-Hohenberg data sets with varying noise levels.

the tested regression methods. The respective thresholding or penalty values are chosen so that they truncate as many parameters as possible, without truncating those which are part of the expected result.

4.2.3 Probability Distributions

The following probability distributions are calculated using Markov chain Monte Carlo with only the minimum of parameters. Twelve walkers are used, which each draw 5000 samples after a burn-in period of 500 sample draws. The prior is set to keep the parameter values within $-50 < a_i < 50$ and the noise positive. The resulting histograms can be found in figure 4.4.

If the peaks were smoothed, their maximum would fit with the expected parameter values. In reality there are some local maxima at the top of the peaks, which are often close, but do not match the real values. This explains why the maximum likelihood estimation has trouble with the noisier data sets.

	real values	18 iterations	20 iterations
λ	-	0	0.05
b_1	0.01	0.01	0.01
b_2	0.01	0.01	0.01
u	-0.7	-0.788	-0.69
Δu	-2	-2.145	-2.025
$\Delta^2 u$	-1	-1.026	-1
u^2	0	0.493	0
$u\Delta u$	0	0.627	0
$u\Delta^2 u$	0	0.101	0
$(\Delta u)^2$	0	0.257	0
$(\Delta u)(\Delta^2 u)$	0	0.118	0
$(\Delta^2 u)^2$	0	0.016	0
u^3	-1	-0.659	-1.045
$u^2(\Delta u)$	0	0.465	0
$u^2(\Delta^2 u)$	0	0.125	0
$u(\Delta u)^2$	0	0.117	0
$u(\Delta u)(\Delta^2 u)$	0	0.167	0
$u(\Delta^2 u)^2$	0	0.032	0
$(\Delta u)^3$	0	0.055	0
$(\Delta u)^2(\Delta^2 u)$	0	0.127	0
$(\Delta u)(\Delta^2 u)^2$	0	0.05	0
$(\Delta^2 u)^3$	0	0.005	0
1	0	-0.039	0

Table 4.12: Maximum likelihood estimation results for a Swift-Hohenberg data set with the noise level $q_1 = q_2 = 0.01$, possible parameters up to the third polynomial of 1 , u , Δu and $\Delta^2 u$ and different numbers of iterations of minimization as well as thresholding with the parameter λ .

	real values	STLS $\lambda = 0.4$	LASSO $\lambda = 0.00027$	IHT $\lambda = 0.35$
u	-0.7	-0.464	-0.003	0.435
Δu	-2	-1.688	-1.367	-2.309
$\Delta^2 u$	-1	-0.841	-0.77	-0.471
u^2	0	-1.491	-0.798	0
$u\Delta u$	0	-0.771	-0.549	-2.318
$u\Delta^2 u$	0	0	0	0
$(\Delta u)^2$	0	0	-0.189	0
$(\Delta u)(\Delta^2 u)$	0	0	0	0
$(\Delta^2 u)^2$	0	0	0.023	0
u^3	-1	5.74	-1.028	3.001
$u^2(\Delta u)$	0	18.225	0	-4.995
$u^2(\Delta^2 u)$	0	4.278	0	0.601
$u(\Delta u)^2$	0	11.387	0	1.111
$u(\Delta u)(\Delta^2 u)$	0	5.155	0.114	0
$u(\Delta^2 u)^2$	0	0.435	-0.088	0
$(\Delta u)^3$	0	0	0	0
$(\Delta u)^2(\Delta^2 u)$	0	0	0	0
$(\Delta u)(\Delta^2 u)^2$	0	0	0	0
$(\Delta^2 u)^3$	0	0	0	0
1	0	0	0	5.155

Table 4.13: SINDy results with different regression methods for a Swift-Hohenberg data set with the noise level $q_1 = q_2 = 0.01$ and possible parameters up to the third polynomial of 1 , u , Δu and $\Delta^2 u$.

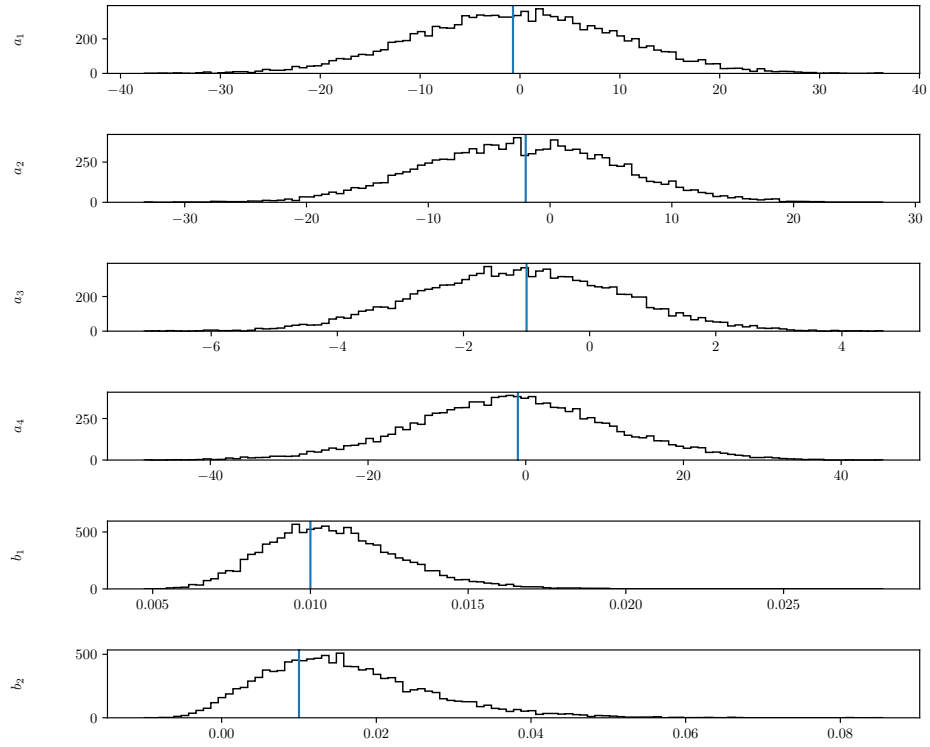


Figure 4.4: Histograms of the probability distributions for the parameters for the Swift-Hohenberg equation, where the blue lines represent the real parameter values.

4.3 The Brusselator

4.3.1 Generating the Data Sets

The Brusselator is a two dimensional SPDE system. It's a model for an oscillating chemical autocatalytic reaction [28] which is depicted in equation (4.3.1). It consists of two fields $u = u(\mathbf{x}, t)$ and $v = v(\mathbf{x}, t)$.

$$\begin{aligned}\partial_t u &= d_u \Delta u + a - (b + 1)u + u^2 v + \sqrt{q_1 + q_2 u^2} \Gamma(\mathbf{x}, t) \\ \partial_t v &= d_v \Delta v + bu - u^2 v + \sqrt{q_1 + q_2 v^2} \Gamma(\mathbf{x}, t)\end{aligned}\quad (4.3.1)$$

The spatio-temporal time series data sets used in the following sections are generated with $d_u = 5$, $d_v = 12$, $a = 3$ and $b = 9$. q_1 and q_2 are the magnitudes for the dynamic noise $\Gamma(\mathbf{x}, t) = \sqrt{2\tau}N(0, 1)$. The computational grid has 128^2 points with a length $L = 120$ in x_1 - and x_2 -direction. The time interval is from $T_0 = 0$ to $T_n = 0.1$ with the time step $\tau = 0.001$, giving it $n = 100$ time steps. The initial condition is $u(\mathbf{x}, t) = v(\mathbf{x}, t) = 3$ with some normal distributed noise with a magnitude of 0.1 on top. A short overview over results with other time steps can be found in the appendix in table A.3.

As with the one-dimensional reaction-diffusion equation, the time steps are calculated using a pseudo-spectral method (see [8]) in combination with the Runge-Kutta 4 method (see [13]), before reversing the Fourier transformation on the newly generated data and adding the noise.

Snapshots of data sets after a noiseless burn-in period can be found in figures 4.5 and 4.6.

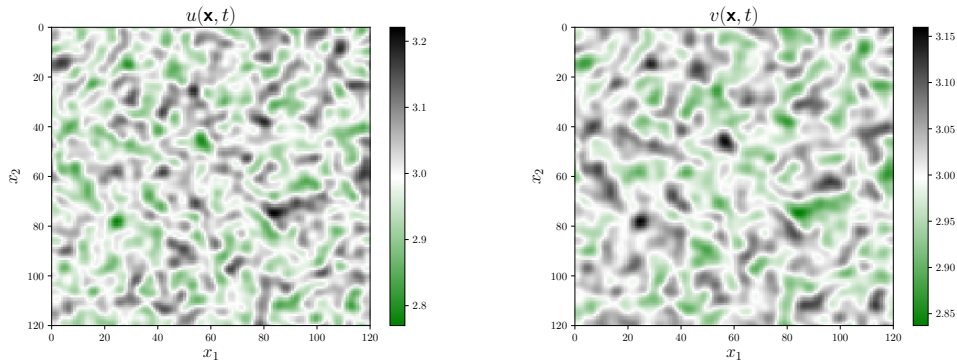


Figure 4.5: Snapshot of a Brusselator data set with the noise magnitude $q_1 = q_2 = 0$.

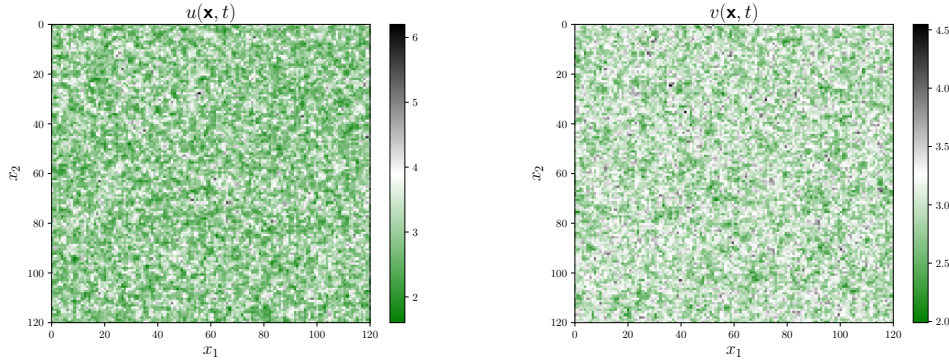


Figure 4.6: Snapshot of a Brusselator data set with the noise magnitude $q_1 = q_2 = 0.5$.

4.3.2 Results of the Maximum Likelihood Estimation

Different Noise Levels at a Minimum of Parameters

The maximum likelihood estimation is done with the minimum of possible parameters, as seen in equations (4.3.2) to (4.3.4), while the noise level is changed. A local minimization is used with zeroes as starting values for all parameters. The results can be found in tables 4.14 and 4.15 for the u and v fields respectively.

$$D_u^{(1)}(\mathbf{a}) = a_1 u(\mathbf{x}, t) + a_2 \Delta u(\mathbf{x}, t) + a_3 u^2(\mathbf{x}, t) v(\mathbf{x}, t) + a_4 \quad (4.3.2)$$

$$D_v^{(1)}(\mathbf{a}) = a_5 \Delta v(\mathbf{x}, t) + a_6 u(\mathbf{x}, t) + a_7 u^2(\mathbf{x}, t) v(\mathbf{x}, t) \quad (4.3.3)$$

$$D^{(2)}(\mathbf{b}) = \begin{pmatrix} b_1 + b_2 u^2(\mathbf{x}, t) & 0 \\ 0 & b_3 + b_4 v^2(\mathbf{x}, t) \end{pmatrix} \quad (4.3.4)$$

The percent error δ to the expected results can be found in table 4.16.

The maximum likelihood estimation gives good results overall as the percent error never rises over ten percent, but there is also no clear correlation between rising noise levels and the percent error, which might be due to the unpredictable influence of random noise.

As is to be expected, the noiseless data set gives the best results. Beyond that it can be noted that the parameters a_2 , a_3 , a_6 and a_7 are close to the expected values, no matter the noise level.

q_1	q_2	a_1	a_2	a_3	a_4	b_1	b_2
0	0	-10.008	4.999	1.001	3.008	0	0
0.1	0	-9.875	4.779	0.97	3.415	0.099	0
0.5	0	-9.02	4.907	0.939	1.737	0.495	0.001
1	0	-9.801	4.809	0.951	3.731	0.989	0.001
0.1	0.1	-10.355	4.868	0.999	4.047	0.117	0.098
0.5	0.5	-9.499	4.927	0.983	2.034	0.557	0.494
1	1	-9.624	4.861	0.953	3.171	1.102	0.987

Table 4.14: Maximum likelihood estimation results for Brusselator data sets with varying noise levels and a minimum of parameters according to equations (4.3.2) and (4.3.4).

q_1	q_2	a_5	a_6	a_7	b_3	b_4
0	0	12.513	9.369	-1.041	0	0
0.1	0	11.26	9.051	-1.006	0.101	-0
0.5	0	11.188	8.725	-0.968	0.496	0.001
1	0	11.176	9.137	-1.016	1.011	-0.001
0.1	0.1	11.104	8.728	-0.969	0.144	0.095
0.5	0.5	11.189	8.746	-0.969	0.697	0.478
1	1	11.068	8.914	-0.978	1.477	0.944

Table 4.15: Maximum likelihood estimation results for Brusselator data sets with varying noise levels and a minimum of parameters according to equations (4.3.3) and (4.3.4).

a_1 and a_5 stray a bit further from the expected values, but are still close enough to be within one percent of the expected value. a_1 shows a bit more variation in values, while a_5 consistently shows values closer to eleven than twelve.

The parameter a_4 is the one which strays furthest from the expected value when noise is added, but considering it functions as an offset it is the parameter most directly influenced by unbalanced combinations of noise and dynamic evolution.

The noise levels are identified nearly perfectly in the case of $q_2 = 0$. If $q_2 \neq 0$ the results for b_1 and b_3 are a little less precise.

The results for the SINDy method with a sample of 400 data points of the same data sets can be found in table 4.17. The regression is either ordinary least squares (gray cells) or iterative hard-thresholding with a threshold of $\lambda = 0.001$, depending on which led to better results.

q_1	q_2	δ
0	0	3.3
0.1	0	4.7
0.5	0	9.6
1	0	6
0.1	0.1	7.7
0.5	0.5	7.3
1	1	6

Table 4.16: Percent error δ for the results of the maximum likelihood estimation for Brusselator data sets with varying noise levels as found in tables 4.14 and 4.15.

In table 4.18 the results for SINDy and the maximum likelihood estimation are compared to the expected values through a percent error. With the exception of the data set without noise the maximum likelihood estimation gives consistently better results.

That different regression methods lead to different results becomes a problem when the true result isn't known, therefore leading to a need for additional tests to determine which method to use for which data set, countering the advantage of speed SINDy usually has over the maximum likelihood estimation.

q_1	q_2	a_1	a_2	a_3	a_4	a_5	a_6	a_7
0	0	-10.009	4.999	1.001	3.008	12.002	9.013	-1.001
0.1	0	-10.518	5.063	1.04	3.424	11.159	9.379	-1.043
0.5	0	-9.093	5.245	1.001	0.18	11.09	9.728	-1.144
1	0	-11.033	4.944	0.952	7.375	11.036	9.676	-1.08
0.1	0.1	-6.822	5.096	0.866	-2.776	11.386	7.189	-0.81
0.5	0.5	-6.804	5.172	0.842	-2.112	11.324	8.607	-0.95
1	1	-9.105	4.826	0.9	3.703	11.271	9.654	-1.019

Table 4.17: SINDy results with ordinary least squares (gray cells) or iterative hard-thresholding with a threshold of $\lambda = 0.001$ for Brusselator data sets with varying noise levels. A minimum of possible parameters is used according to equations (4.3.2) and (4.3.3).

q_1	q_2	δ_{SINDy}	δ_{MLE}
0	0	0.1	3.3
0.1	0	6.0	4.7
0.5	0	16.8	9.6
1	0	24.5	6
0.1	0.1	36.2	7.7
0.5	0.5	32.0	7.3
1	1	8.0	5.4

Table 4.18: Percent error δ for the $D^{(1)}$ results of the maximum likelihood estimation and the SINDy method for Brusselator data sets with varying noise levels as found in tables 4.14, 4.15 and 4.17.

4.3.3 Probability Distributions

The probability distributions are calculated using Markov chain Monte Carlo with only the minimum of parameters. Twenty-two walkers are used, which each draw 10000 samples after a burn-in period of 500 sample draws. The prior is set to keep the parameter values within $-50 < a_i < 50$ and the noise positive.

The resulting histograms can be found in figures 4.7 and 4.8.

A lot of the peaks are not as distinct as they were for the one-dimensional reaction diffusion system, but the expected results are in most cases at least close to the maximum.

The most notable exception is the parameter a_4 which describes the offset and is expected to show a maximum at the value 3. Instead the probability distribution is nearly flat, which makes identifying a clear maximum impossible.

But while the maximum likelihood estimation also didn't identify the expected parameter value it still fell into a range of 3 ± 1 which is a lot less spread out than the histogram suggests. This is probably due to the fact that during the maximum likelihood estimation a local minimization was used, which found the closest local maximum for that value, instead of the global maximum.

The parameters a_1 and a_6 show similar problems. Their curves are not quite flat and the peaks are still spread enough that a wide range of values could correspond to the maximum. During the maximum likelihood estimation both were found with more precision than the histogram suggests.

The histogram for a_5 shows a peak at about eleven which explains why the maximum likelihood never gave the expected result of twelve.

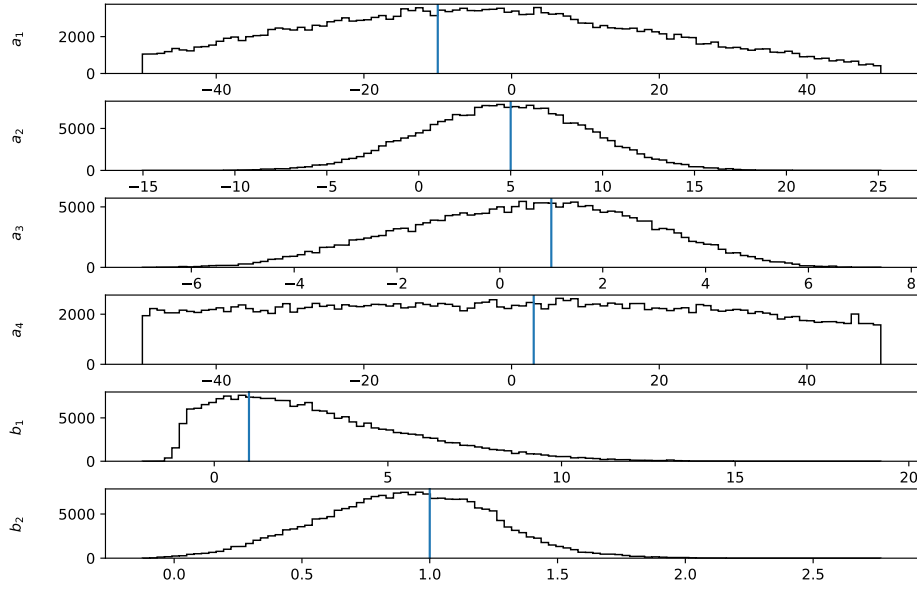


Figure 4.7: Histograms of the probability distributions for the parameters of the u field of the Brusselator equation, where the blue lines represent the real parameter values.

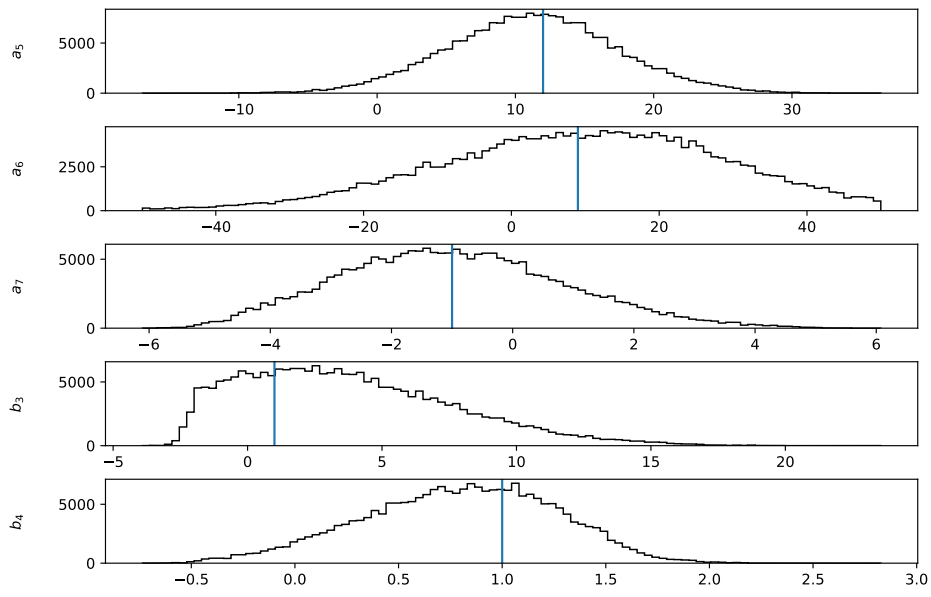


Figure 4.8: Histograms of the probability distributions for the parameters of the v field of the Brusselator equation, where the blue lines represent the real parameter values.

5 Identifying a Stochastic Partial Differential Equation for Field Data of the Ising Model

5.1 The Two-Dimensional Ising Model

Ferromagnetic materials can show magnetization even without an external magnetic field to induce it, if their temperature T is below the material specific Curie temperature T_C . This is caused by the alignment of the electron spins with their neighbors, due to an inner magnetic field which in turn is caused by spin interactions. Above the Curie temperature thermal fluctuations of spin directions counter the ordering effect of the alignment, leading to a phase transition from ferromagnetic to paramagnetic behavior [12].

The phenomenon can be described by the Ising model. The following explanations will be limited to the two-dimensional Ising model without influence of an external magnetic field.

According to the model the ferromagnetic material consists of discrete magnetic dipole moments (spins) arranged on a two-dimensional lattice of periodic structure. Each spin S_i can take on the values $+1$ and -1 and interacts with the spins on neighboring lattice sites.

The energy of the system is given by the Hamiltonian function seen in equation (5.1.1), where i and j are next neighbors, while J is an interaction parameter independent of the particular combination of i and j [21].

$$H = -J \sum_{(i,j)} S_i S_j \quad (5.1.1)$$

Spins change their direction to minimize energy if too many of their neighbors have opposing spin values. The process is opposed by thermal fluctuations: The higher the temperature the more likely spins are to change direction even if it is not energetically favorable.

Figure 5.1 shows the development of domains with different spin directions at temperatures $T < T_C$, starting from random spin directions.

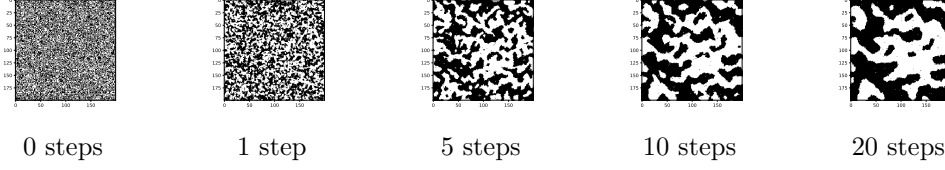


Figure 5.1: Example for the development of spins on a grid according to the Ising model, where black and white denote different spin directions.

5.2 Generating the Data

The data set which simulates the Ising model is created using a cellular automaton on a two-dimensional rectangular grid with periodic boundary conditions. The algorithm is described below and follows the steps used in reference [22], while also following their suggestion of updating non-interacting lattice sites simultaneously to speed up computation time.

For every non-interacting lattice site the energy $\Delta H = H_{\text{current state}} - H_{\text{flipped state}}$ is calculated according to equation (5.1.1).

If $\Delta H \leq 0$ the spin is flipped. If $\Delta H > 0$ a random number r between 0 and 1 is generated. If $r < \exp\left(\frac{\Delta H}{T}\right)$ the spin is flipped anyway, representing the effect of thermal fluctuation.

Repeating the above algorithm so that all lattice sites have been updated once makes up one time step in the dynamic evolution of the grid. The final data set consists of 100 time steps and to make the data more continuous it is blurred through a Gaussian kernel provided by the Python package *scipy.ndimage.filters*. The function is called *gaussian_filter* and the accompanying filter matrix is depicted in equation (5.2.1).

$$\frac{1}{256} \begin{pmatrix} 0.87 & 3.62 & 5.97 & 3.62 & 0.87 \\ 3.62 & 15.01 & 24.73 & 15.01 & 3.62 \\ 5.97 & 24.73 & 40.74 & 24.73 & 5.97 \\ 3.62 & 15.01 & 24.73 & 15.01 & 3.62 \\ 0.87 & 3.62 & 5.97 & 3.62 & 0.87 \end{pmatrix} \quad (5.2.1)$$

Snapshots of the blurred data can be found in figure 5.2.

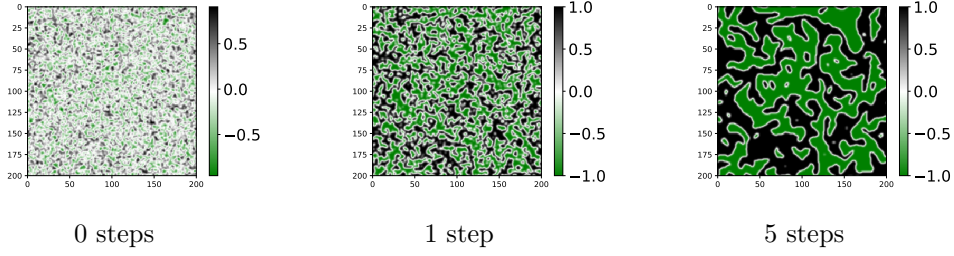


Figure 5.2: Development of a blurred Ising data set at $T = 0.01$, where black and green denote different spin directions.

5.3 Results

The maximum likelihood estimation is tested on a data set of the Ising model generated at low temperatures ($T = 0.01$) to minimize thermal fluctuations, thus allowing to ignore noise parameters during the maximum likelihood estimation. The data is blurred and the maximum likelihood estimation applied in combination with thresholding with different thresholds λ .

Using a full set of polynomials up to third order of u , Δu and $\Delta^2 u$ as parameters leads to the results depicted in table 5.1.

The resulting equation at a threshold of $\lambda = 0.2$ is depicted in equation (5.3.1).

$$\partial_t u = -0.311u - 0.728\Delta u + 0.311u^3 + 0.783u^2\Delta u \quad (5.3.1)$$

To test the validity of those results a new data set is calculated, starting from the first field of the Ising data set and then adding new time steps using the pseudo-spectral method in combination with Runge-Kutta 4, while the maximum likelihood results build the right-hand side of the equation.

The resulting dynamics have no resemblance to the ones expected of the Ising model. Snapshots can be found in figure 5.3.

Using the sum of the neighboring grid points in addition to the polynomials leads to the results depicted in table 5.2.

The resulting equation at a threshold of $\lambda = 0.2$ is depicted in equation (5.3.2) where N is a field consisting of the sum of the next neighbors.

$$\partial_t u = -1.485u - 0.699\Delta u - 0.362u^3 + 0.674u^2\Delta u + 0.232N \quad (5.3.2)$$

Building a data set with any of these results requires time steps of $\tau = 10^{-13}$ and

λ	0	0.05	0.1	0.15	0.2
u	-0.296	-0.284	-0.313	-0.279	-0.311
Δu	-0.910	-0.849	-0.836	-0.777	-0.728
$\Delta^2 u$	0.175	0.157	0.144	0.151	0
u^2	-0.001	0	0	0	0
$u\Delta u$	0.036	0	0	0	0
$u\Delta^2 u$	0.008	0	0	0	0
$(\Delta u)^2$	-0.068	-0.055	0	0	0
$(\Delta u)(\Delta^2 u)$	0.039	0	0	0	0
$(\Delta^2 u)^2$	-0.003	0	0	0	0
u^3	0.297	0.284	0.312	0.279	0.311
$u^2\Delta u$	0.980	0.935	0.821	0.855	0.783
$u^2\Delta^2 u$	-0.160	-0.155	-0.160	-0.161	0
$u(\Delta u)^2$	0.012	0	0	0	0
$u(\Delta u)(\Delta^2 u)$	0.035	0	0	0	0
$u(\Delta^2 u)^2$	0.001	0	0	0	0
$(\Delta u)^3$	0.220	0.128	0.100	0	0
$(\Delta u)^2(\Delta^2 u)$	-0.151	-0.050	0	0	0
$(\Delta u)(\Delta^2 u)^2$	0.034	0	0	0	0
$(\Delta^2 u)^3$	-0.002	0	0	0	0

Table 5.1: Maximum likelihood estimation results for a blurred Ising data set at $T = 0.01$ with parameters up to the third polynomial of u , Δu and $\Delta^2 u$. Different thresholds λ are used for truncation.

leads to a dynamic which shows at least some ordering effect similar to the one expected from the Ising model, even while the range of the field u quickly grows beyond the spin values.

The results look very similar no matter which threshold is used. Snapshots of the results for the case of $\lambda = 0.15$ can be found in figure 5.4.

Despite what those results suggest, it is not impossible to recreate dynamics which look similar to those of the Ising model. Using equation (5.3.3) for data set creation leads to the results seen in figures 5.6 and 5.7.

Figure 5.6 shows that starting from total chaos (left) leads to a result (right) which only vaguely resemble the result of the blurred Ising data set after three time steps (middle).

Figure 5.7 shows that using the more ordered first time step (left) as a starting value leads to a result (right) which matches the expected result (middle) fairly

5 Identifying a Stochastic Partial Differential Equation for Field Data of the Ising Model

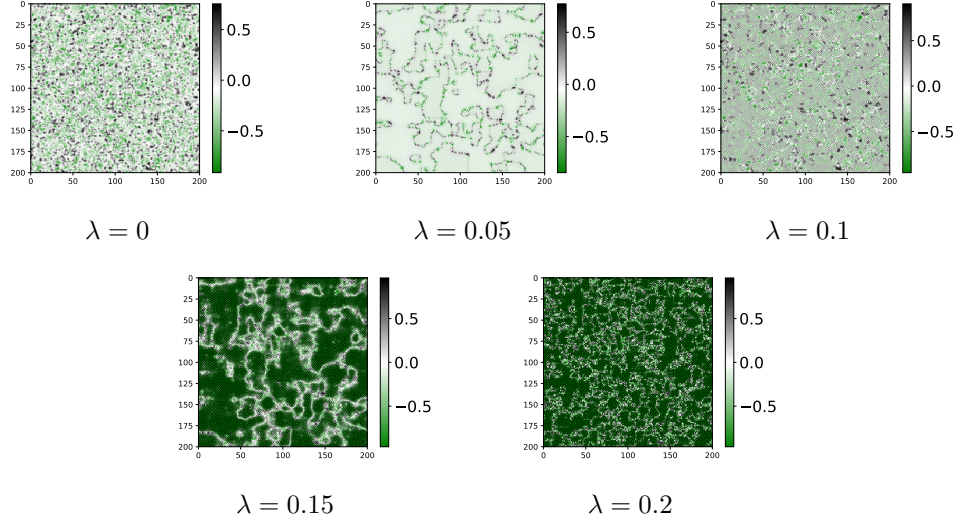


Figure 5.3: Snapshots of data sets calculated from the results of the maximum likelihood estimation seen in table 5.1.

well.

The maximum likelihood estimation however does not consider these good results.

$$\partial_t u = u + \Delta u - u^3 \quad (5.3.3)$$

λ	0	0.05	0.1	0.15
u	-1.467	-1.449	-1.461	-1.485
Δu	-0.883	-0.834	-0.805	-0.699
$\Delta^2 u$	0.178	0.147	0.138	0
u^2	0	0	0	0
$u\Delta u$	0.099	0.100	0.108	0
$u\Delta^2 u$	0.010	0	0	0
$(\Delta u)^2$	-0.038	0	0	0
$(\Delta u)(\Delta^2 u)$	0.040	0	0	0
$(\Delta^2 u)^2$	-0.004	0	0	0
u^3	-0.348	-0.356	-0.353	-0.362
$u^2\Delta u$	0.645	0.661	0.650	0.674
$u^2\Delta^2 u$	-0.156	-0.135	-0.137	0
$u(\Delta u)^2$	-0.176	-0.169	-0.100	0
$u(\Delta u)(\Delta^2 u)$	-0.033	0	0	0
$u(\Delta^2 u)^2$	0	0	0	0
$(\Delta u)^3$	0.271	0.146	0.100	0
$(\Delta u)^2(\Delta^2 u)$	-0.201	-0.050	0	0
$(\Delta u)(\Delta^2 u)^2$	0.034	0	0	0
$(\Delta^2 u)^3$	-0.002	0	0	0
Σ neighbors	0.227	0.226	0.227	0.232

Table 5.2: Maximum likelihood estimation results for a blurred Ising data set at $T = 0.01$ with parameters up to the third polynomial of u , Δu and $\Delta^2 u$ and an additional parameter for the sum of the neighboring spins. Different thresholds λ are used for truncation.

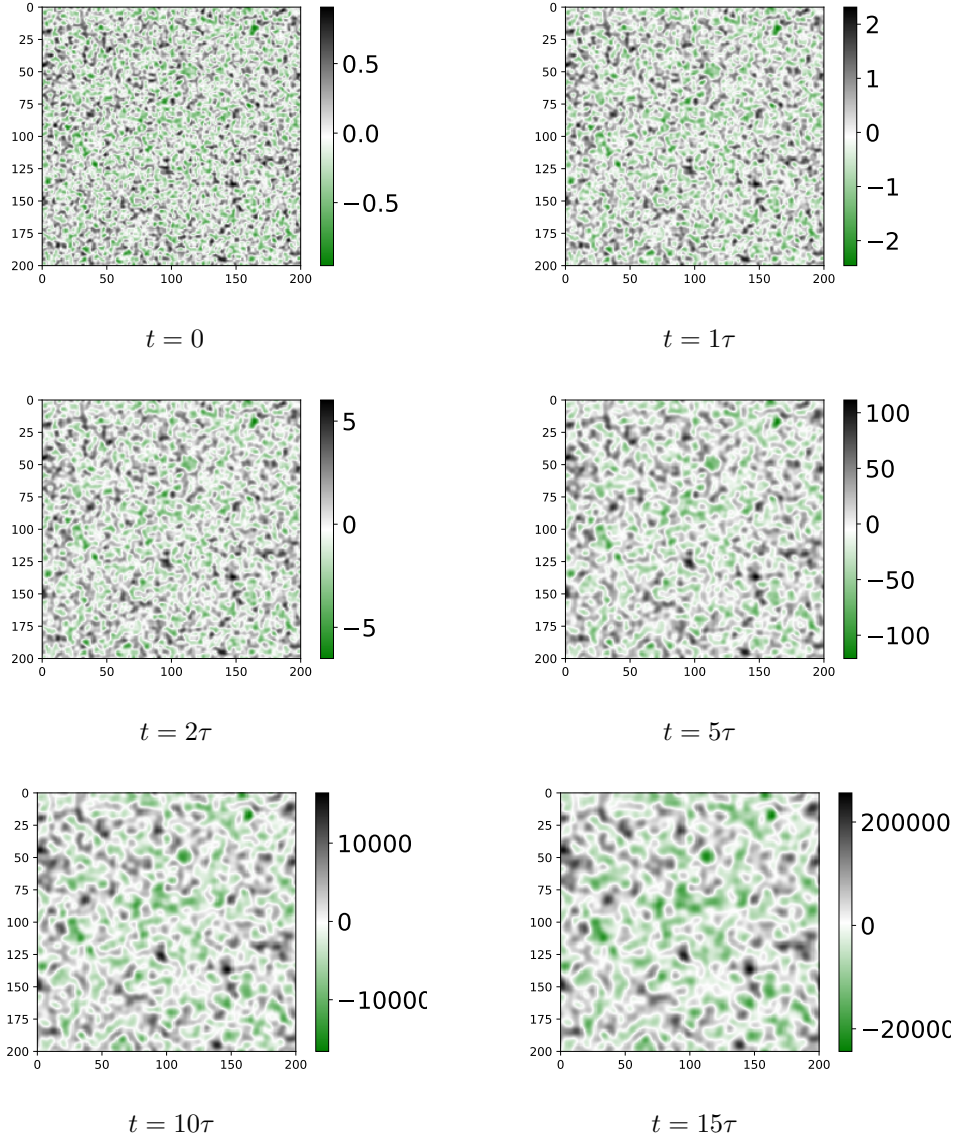
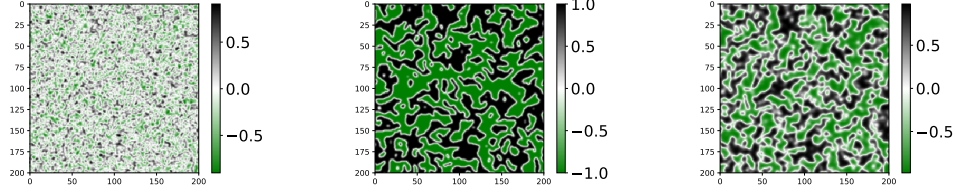
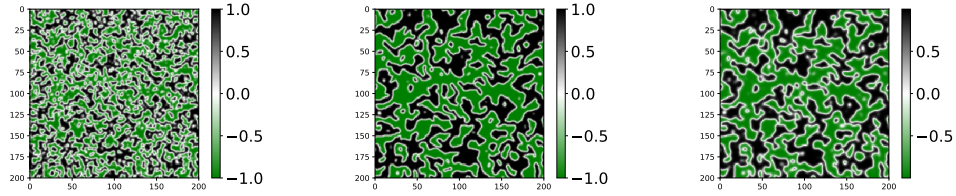


Figure 5.4: Snapshots of the data set calculated from the result of the maximum likelihood estimation with the threshold $\lambda = 0.15$ seen in table 5.2.



Blurred Ising data at $t = 0$ Blurred Ising data at $t = 3\tau$ Integrated data

Figure 5.6: Snapshot of a blurred Ising data set at $t = 0$ (left) as well as after three time steps (middle) and a snapshot from a data set calculated from equation (5.3.3) with the $t = 0$ field as starting field (right).



Blurred Ising data at $t = \tau$ Blurred Ising data at $t = 3\tau$ Integrated data

Figure 5.7: Snapshot of a blurred Ising data set at $t = \tau$ (left) as well as after two more time steps (middle) and a snapshot from a data set calculated from equation (5.3.3) with the $t = \tau$ field as starting field (right).

6 Conclusion and Outlook

Applying the maximum likelihood estimation to generated one-dimensional field data sets while only allowing a minimum of building blocks for the equation leads to reliable results given a reasonable noise level. What passes as a reasonable noise level varies depending on the equation and the parameters the data set is generated from.

While the one-dimensional reaction-diffusion equation gives good results even for relatively big noise levels, the Swift-Hohenberg equation needs a low noise level and smaller time steps for the method to succeed.

Compared to the maximum likelihood approach SINDy works faster and with equal or even minimal better precision on data sets without noise, but it can't handle noisy data nearly as well.

Giving the maximum likelihood estimation an excess amount of parameters only gives good results for the reaction-diffusion equation, though it takes a lot of iterations until the minimization is successful. Adding sparsity promoting methods such as BIC or thresholding can reduce those iterations and in case of the thresholding also allows for reliable results for the Swift-Hohenberg equation.

The hyperparameter optimization for the thresholding parameter exposed some weaknesses of the sparsity promoting methods, such as the difficulty in choosing the right threshold outside of optimization, as well as problems when combining thresholding with the BIC, where too many parameters were truncated. Finding a sparsity promoting method which works reliably for a variety of problems instead of only in localized instances could be a topic of further research.

Applying the maximum likelihood estimation to the two-dimensional data set generated from the Brusselator equation demonstrates that the method is valid in multiple dimensions, but also shows that some parameters are easier to identify correctly than others.

Trying the method on a data set generated from the two-dimensional Ising model, which is a cellular automaton, does not lead to results which match the original dynamic when only polynomials of u , Δu and $\Delta^2 u$ up to third order are used.

It takes adding a parameter for the sum of the neighboring grid points to get a dynamic that looks at least a little like the one from the Ising model, but even then the results do not stay in a range from -1 to 1, but grows rapidly after a few very small time steps.

Further testing is needed to determine if and how better results can be found, and whether these difficulties exist because cellular automata can not be translated into partial differential equations with this method, or because the Ising model in particular is not suited to it, due to only allowing two spin directions.

A Appendix

A.1 Parameters for Data Set Generation and their MLE Results

How much information about the dynamic evolution a data set holds depends on the parameters it is generated with, namely the amount of points in the discretisation grid, the size of the grid, the amount of time steps, the size of those time steps and the noise level. The following tables show test results with different parameter combinations and how the maximum likelihood estimation handles the resulting data sets.

The results for the one dimensional reaction-diffusion equation can be found in table A.1, the results for the Swift-Hohenberg equation in table A.2 and the results for the Brusselator in table A.3. As it is of interest if the maximum likelihood estimation can identify the dynamic evolution, only the results for $D^{(1)}$ are depicted.

parameters						MLE results		
grid		time step		noise		$D^{(1)}$		
points	length	size	amount	q_1	q_2	u	Δu	u^3
256^2	$[0.100]^2$	0.01	101	0	0	1.00	0.25	-1.00
256^2	$[0.100]^2$	0.01	101	0.5	0	0.95	0.23	-0.96
256^2	$[0.100]^2$	0.01	101	1	0	0.91	0.23	-0.93
256^2	$[0.100]^2$	0.01	101	0.5	0.5	0.92	0.23	-0.92
256^2	$[0.100]^2$	0.01	101	1	1	0.85	0.23	-0.87
256^2	$[0.100]^2$	0.05	101	0.5	0.5	0.72	0.16	-0.72
256^2	$[0.100]^2$	0.005	101	0.5	0.5	0.95	0.24	-0.99

Table A.1: MLE results for $D^{(1)}$ for 1D reaction-diffusion data sets generated with different parameters.

parameters						MLE results			
grid		time step		noise		$D^{(1)}$			
points	length	size	amount	q_1	q_2	u	Δu	$\Delta^2 u$	u^3
128^2	$[0, 32]^2$	1e-2	101	0	0	-0.75	-2.01	-0.99	-0.98
128^2	$[0, 32]^2$	1e-2	101	1e-8	0	-0.68	-1.98	-0.99	-1.04
128^2	$[0, 32]^2$	1e-2	101	1e-7	0	-0.59	-1.66	-0.90	-1.22
128^2	$[0, 32]^2$	1e-2	101	1e-4	0	2.3	1.75	-0.21	0.87
64^2	$[0, 32]^2$	1e-2	101	1e-7	0	-0.70	-2.00	-1.00	-1.05
64^2	$[0, 32]^2$	1e-2	101	1e-6	0	-0.70	-2.00	-1.00	-1.05
64^2	$[0, 32]^2$	1e-2	101	1e-5	0	-0.73	-2.01	-1.00	-0.97
64^2	$[0, 32]^2$	1e-2	101	1e-4	0	-0.51	-1.78	-0.95	-1.23
64^2	$[0, 32]^2$	1e-3	101	1e-4	0	-0.73	-1.99	-1.00	-0.96
64^2	$[0, 32]^2$	1e-3	101	1e-3	0	-0.58	-1.94	-0.99	-1.04
64^2	$[0, 32]^2$	1e-4	101	1e-3	0	-0.66	-1.95	-1.00	-0.95
64^2	$[0, 32]^2$	1e-4	101	1e-2	0	0.63	-1.87	-1.00	-0.80
64^2	$[0, 32]^2$	1e-5	101	1e-2	0	1.94	-1.89	-1.00	-1.57
32^2	$[0, 32]^2$	1e-3	101	1e-3	0	-0.65	-1.95	-0.99	-0.97
32^2	$[0, 32]^2$	1e-3	101	1e-2	0	-0.90	-2.07	-1.01	-0.88
32^2	$[0, 32]^2$	1e-3	101	1e-2	1e-2	0.22	-1.65	-0.97	-0.91
128^2	$[0, 90]^2$	1e-4	101	1e-4	0	-0.70	-1.98	-0.99	-0.98
128^2	$[0, 90]^2$	1e-4	101	1e-3	0	-0.67	-1.93	-0.97	-1.02
128^2	$[0, 90]^2$	1e-4	101	1e-2	0	-0.56	-1.75	-0.93	-0.87
128^2	$[0, 100]^2$	1e-4	101	1e-2	0	-0.62	-1.82	-0.95	-1.07
128^2	$[0, 100]^2$	1e-3	101	1e-2	0	0.06	-0.89	-0.71	-0.97
128^2	$[0, 100]^2$	1e-4	101	1e-2	1e-2	-0.54	-1.85	-0.96	-1.10
64^2	$[0, 60]^2$	1e-4	101	1e-2	1e-2	-1.07	-2.03	-0.97	-0.33
128^2	$[0, 120]^2$	1e-4	101	1e-2	1e-2	-0.57	-1.85	-0.97	-1.06
128^2	$[0, 120]^2$	1e-4	101	1e-3	1e-3	-0.71	-1.98	-0.99	-0.99
128^2	$[0, 120]^2$	1e-4	101	5e-3	5e-3	-0.70	-1.94	-0.98	-1.01
128^2	$[0, 120]^2$	1e-4	101	5e-3	0	-0.65	-1.96	-0.99	-0.99
128^2	$[0, 120]^2$	1e-4	101	1e-3	0	-0.68	-1.97	-0.99	-0.99
128^2	$[0, 120]^2$	1e-4	101	1e-2	0	-0.68	-1.90	-0.98	-1.04
128^2	$[0, 120]^2$	1e-4	101	1e-1	0	-0.48	-1.99	-0.98	-1.37
128^2	$[0, 120]^2$	1e-4	101	0	0	-0.70	-2.00	-1.00	-1.00
128^2	$[0, 120]^2$	1e-4	101	1e-1	1e-1	-0.72	-1.85	-0.98	-0.72

Table A.2: MLE results for $D^{(1)}$ for 1D Swift-Hohenberg data sets generated with different parameters.

parameters						MLE results						
grid		time step		noise		$D_u^{(1)}$				$D_v^{(1)}$		
points	length	size	amount	$q_1=q_3$	$q_2=q_4$	u	Δu	u^2v	1	Δv	u	u^2v
128^2	$[0, 120]^2$	0.005	101	0	0	-10.00	5.00	1.00	3.00	12.00	9.00	-1.00
128^2	$[0, 120]^2$	0.005	101	0.5	0.5	-7.85	4.37	0.79	2.12	8.44	7.83	-0.86
128^2	$[0, 120]^2$	0.001	101	0	0	-10.00	5.01	1.00	3.00	12.01	9.00	-1.00
128^2	$[0, 120]^2$	0.001	101	0.5	0	-9.51	4.85	0.94	3.17	11.11	8.86	-0.99
128^2	$[0, 120]^2$	0.001	101	1	0	-8.94	4.86	0.93	1.84	11.23	8.51	-0.95
128^2	$[0, 120]^2$	0.001	101	0.5	0.5	-10.34	4.75	0.98	4.97	11.20	8.70	-0.97
128^2	$[0, 120]^2$	0.001	101	1	1	-9.62	4.86	0.95	3.17	11.07	8.92	-0.98

Table A.3: MLE results for $D^{(1)}$ for 2D Brusselator data sets generated with different parameters.

A.2 Hyperparameter Optimization Results

The full results for the hyperparameter optimization for thresholding with the negative log-likelihood $-L(\mathbf{c})$ as a loss function, which is discussed in subsection 4.1.2, are detailed in tables A.4 to A.6.

The full results for the hyperparameter optimization discussed in subsection 4.1.2, which optimizes a thresholding parameter with the BIC as loss function can be found in table A.7.

θ	$-L(\mathbf{c})$	u	Δu	u^2	$u\Delta u$	$(\Delta u)^2$	u^3	$\Delta^2 u$	1	q_1	q_2
0.0000	-33.31163	-0.34	0.29	1.14	0.03	0.00	0.61	0.00	0.63	1.02	0.95
0.0117	-4.93939	0.48	0.94	0.25	0.80	0.02	0.75	0.01	0.36	1.38	0.04
0.0120	-33.62259	0.84	0.23	-0.01	0.00	0.00	-0.86	0.00	0.00	1.02	0.93
0.0127	-20.80984	-1.79	0.54	-1.48	0.58	0.01	3.89	0.00	-0.05	1.77	0.73
0.0129	-33.47890	0.90	0.24	2.16	0.06	0.00	-0.64	0.00	-0.79	1.04	0.92
0.0146	-30.94999	-0.16	0.34	0.19	0.03	0.00	1.26	0.00	0.65	0.72	0.95
0.0158	19.26082	0.24	0.95	0.03	0.87	0.02	1.12	0.00	0.34	1.26	-0.05
0.0161	19.26082	0.24	0.95	0.03	0.87	0.02	1.12	0.00	0.34	1.26	-0.05
0.0165	5.61403	0.12	1.06	-0.17	0.85	0.03	1.47	0.02	0.41	1.74	0.00
0.0189	8.83601	0.12	1.07	-0.18	0.87	0.03	1.48	0.00	0.41	1.76	0.00
0.0193	8.61585	0.12	1.07	-0.19	0.87	0.03	1.48	0.00	0.41	1.76	0.00
0.0203	7.59982	0.12	1.06	-0.18	0.86	0.02	1.48	0.00	0.41	1.74	0.00
0.0206	8.37096	0.14	1.03	-0.12	0.83	0.03	1.42	0.00	0.43	1.66	0.03
0.0228	15.85587	0.14	1.04	0.09	0.86	0.03	1.24	0.00	0.39	1.57	-0.08
0.0236	-33.62264	0.86	0.23	0.00	0.00	0.00	-0.88	0.00	0.00	1.02	0.93

Table A.4: Part one of the results of the MLE for the 1D reaction-diffusion equation with ten parameters, hyperparameter optimization for thresholding and one iteration of minimization. Grey cells mark optimal results.

θ	\log	u	Δu	u^2	$u\Delta u$	$(\Delta u)^2$	u^3	$\Delta^2 u$	1	q_1	q_2
0.0262	17.57620	0.39	0.98	-0.03	1.01	0.03	1.14	0.00	0.25	1.39	0.06
0.0263	17.57620	0.39	0.98	-0.03	1.01	0.03	1.14	0.00	0.25	1.39	0.06
0.0267	23.71658	0.31	0.98	-0.13	1.00	0.03	1.12	0.00	0.28	1.42	-0.06
0.0280	-33.56906	-0.10	0.24	0.00	0.00	0.00	0.00	0.00	0.00	1.02	0.93
0.0300	11.97995	0.21	0.93	0.16	0.96	0.03	1.11	0.00	0.39	1.61	0.00
0.0302	11.97995	0.21	0.93	0.16	0.96	0.03	1.11	0.00	0.39	1.61	0.00
0.0319	10.90231	0.00	1.09	0.11	0.93	0.03	1.38	0.00	0.40	1.95	0.00
0.0336	15.35780	0.06	1.06	0.14	0.93	0.04	1.29	0.00	0.40	1.86	0.00
0.0350	15.35780	0.06	1.06	0.14	0.93	0.04	1.29	0.00	0.40	1.86	0.00
0.0359	15.35780	0.06	1.06	0.14	0.93	0.04	1.29	0.00	0.40	1.86	0.00
0.0364	15.35780	0.06	1.06	0.14	0.93	0.04	1.29	0.00	0.40	1.86	0.00
0.0382	-33.35133	0.31	0.25	1.18	0.06	0.00	-0.19	0.00	0.65	0.99	0.96
0.0395	-33.62264	0.86	0.23	0.00	0.00	0.00	-0.88	0.00	0.00	1.02	0.93
0.0414	-33.62264	0.86	0.23	0.00	0.00	0.00	-0.88	0.00	0.00	1.02	0.93
0.0425	-33.32378	0.20	0.25	1.12	0.07	0.00	0.22	0.00	0.57	1.00	0.92
0.0431	-33.62264	0.86	0.23	0.00	0.00	0.00	-0.88	0.00	0.00	1.02	0.93
0.0435	-33.62264	0.86	0.23	0.00	0.00	0.00	-0.88	0.00	0.00	1.02	0.93
0.0483	-33.62264	0.86	0.23	0.00	0.00	0.00	-0.88	0.00	0.00	1.02	0.93
0.0488	-33.62264	0.86	0.23	0.00	0.00	0.00	-0.88	0.00	0.00	1.02	0.93
0.0489	-33.62264	0.86	0.23	0.00	0.00	0.00	-0.88	0.00	0.00	1.02	0.93
0.0504	-33.62264	0.86	0.23	0.00	0.00	0.00	-0.88	0.00	0.00	1.02	0.93
0.0520	-33.59932	0.00	0.23	0.06	0.00	0.00	-0.46	0.00	-0.06	1.02	0.93
0.0537	-33.62264	0.86	0.23	0.00	0.00	0.00	-0.88	0.00	0.00	1.02	0.93
0.0571	-33.10567	-0.11	0.25	1.30	0.07	0.00	0.41	0.00	0.92	0.99	1.08
0.0572	-33.10567	-0.11	0.25	1.30	0.07	0.00	0.41	0.00	0.92	0.99	1.08
0.0574	-32.99509	-0.17	0.27	1.31	0.08	0.00	0.56	0.00	1.00	0.96	1.13
0.0639	-33.10869	0.00	0.26	1.32	0.08	0.00	0.40	0.00	0.95	1.05	0.92
0.0640	-33.10869	0.00	0.26	1.32	0.08	0.00	0.40	0.00	0.95	1.05	0.92
0.0715	-29.99966	0.00	0.24	0.00	0.00	0.00	-0.29	0.00	-0.42	1.61	0.00
0.0737	-33.62264	0.86	0.23	0.00	0.00	0.00	-0.88	0.00	0.00	1.02	0.93
0.0743	-33.62264	0.86	0.23	0.00	0.00	0.00	-0.88	0.00	0.00	1.02	0.93
0.0802	-33.59930	0.00	0.23	0.00	0.00	0.00	-0.45	0.00	0.00	1.02	0.93
0.0812	-33.59930	0.00	0.23	0.00	0.00	0.00	-0.45	0.00	0.00	1.02	0.93
0.0849	-33.23563	-0.09	0.27	0.75	0.00	0.00	0.99	0.00	-0.79	1.01	0.96
0.0871	-33.62264	0.86	0.23	0.00	0.00	0.00	-0.88	0.00	0.00	1.02	0.93
0.0897	-33.62264	0.86	0.23	0.00	0.00	0.00	-0.88	0.00	0.00	1.02	0.93
0.0931	-33.62264	0.86	0.23	0.00	0.00	0.00	-0.88	0.00	0.00	1.02	0.93
0.0949	-33.59930	0.00	0.23	0.00	0.00	0.00	-0.45	0.00	0.00	1.02	0.93
0.1000	-33.62264	0.86	0.23	0.00	0.00	0.00	-0.88	0.00	0.00	1.02	0.93
0.1018	1.57314	0.65	0.93	0.59	0.55	0.00	0.93	0.00	0.52	1.26	0.14
0.1098	1.57314	0.65	0.93	0.59	0.55	0.00	0.93	0.00	0.52	1.26	0.14
0.1122	1.57314	0.65	0.93	0.59	0.55	0.00	0.93	0.00	0.52	1.26	0.14
0.1139	1.57314	0.65	0.93	0.59	0.55	0.00	0.93	0.00	0.52	1.26	0.14
0.1212	-1.46512	0.65	0.92	0.58	0.53	0.00	0.95	0.00	0.54	1.30	0.16

Table A.5: Part two of the results of the MLE for the 1D reaction-diffusion equation with ten parameters, hyperparameter optimization for thresholding and one iteration of minimization. Grey cells mark optimal results.

θ	\log	u	Δu	u^2	$u\Delta u$	$(\Delta u)^2$	u^3	$\Delta^2 u$	1	q_1	q_2
0.1234	-33.62264	0.86	0.23	0.00	0.00	0.00	-0.88	0.00	0.00	1.02	0.93
0.1271	-33.62264	0.86	0.23	0.00	0.00	0.00	-0.88	0.00	0.00	1.02	0.93
0.1313	-0.84717	0.59	0.92	0.58	0.54	0.00	1.01	0.00	0.66	1.29	0.15
0.1321	-0.84717	0.59	0.92	0.58	0.54	0.00	1.01	0.00	0.66	1.29	0.15
0.1416	-29.96135	-0.39	0.25	0.00	0.00	0.00	0.00	0.00	0.00	1.61	0.00
0.1424	-29.96135	-0.39	0.25	0.00	0.00	0.00	0.00	0.00	0.00	1.61	0.00
0.1469	-29.96135	-0.39	0.25	0.00	0.00	0.00	0.00	0.00	0.00	1.61	0.00
0.1586	2.48781	0.63	1.01	0.48	0.63	0.00	1.21	0.00	0.65	1.50	0.20
0.1613	2.48781	0.63	1.01	0.48	0.63	0.00	1.21	0.00	0.65	1.50	0.20
0.1717	-33.62264	0.86	0.23	0.00	0.00	0.00	-0.88	0.00	0.00	1.02	0.93
0.1734	-33.62264	0.86	0.23	0.00	0.00	0.00	-0.88	0.00	0.00	1.02	0.93
0.1778	0.76396	0.62	0.97	0.49	0.63	0.00	1.15	0.00	0.64	1.45	0.22
0.1792	0.76396	0.62	0.97	0.49	0.63	0.00	1.15	0.00	0.64	1.45	0.22
0.1959	-33.62264	0.86	0.23	0.00	0.00	0.00	-0.88	0.00	0.00	1.02	0.93
0.2020	-32.80158	0.35	0.33	-1.22	0.00	0.00	0.61	0.00	1.00	1.06	0.66
0.2057	-32.88287	0.33	0.30	-0.79	0.00	0.00	0.55	0.00	1.34	0.88	1.24
0.2208	-33.62264	0.86	0.23	0.00	0.00	0.00	-0.88	0.00	0.00	1.02	0.93
0.2238	-33.62264	0.86	0.23	0.00	0.00	0.00	-0.88	0.00	0.00	1.02	0.93
0.2329	-32.10818	0.54	0.30	0.90	0.00	0.00	1.03	0.00	0.98	1.21	0.49
0.2553	-33.53857	0.80	0.26	-0.60	0.00	0.00	-0.71	0.00	0.73	1.02	0.94
0.2674	-33.53903	0.00	0.27	0.00	0.00	0.00	0.00	0.00	0.00	1.02	0.93
0.2912	-33.41063	0.44	0.29	-0.31	0.00	0.00	0.00	0.00	0.65	1.03	0.92
0.3009	-33.36636	0.45	0.30	-0.37	0.00	0.00	0.00	0.00	0.65	1.01	0.95
0.3094	-33.37760	0.35	0.31	0.00	0.00	0.00	0.00	0.00	0.00	1.03	0.93
0.3490	-32.99775	0.00	0.35	0.00	0.00	0.00	0.48	0.00	0.50	1.02	0.98
0.3510	-33.06134	0.00	0.35	0.00	0.00	0.00	0.35	0.00	0.00	1.03	0.93
0.3758	-29.78457	0.57	0.45	0.50	0.00	0.00	0.63	0.00	0.64	0.82	0.71
0.4166	-32.32744	0.00	0.42	0.00	0.00	0.00	0.51	0.00	0.00	1.06	0.93
0.4189	-32.29926	0.00	0.42	0.00	0.00	0.00	0.66	0.00	0.00	1.06	0.92
0.4477	-11.11968	0.53	0.47	0.54	0.48	0.00	0.60	0.00	0.51	0.57	0.56
0.4624	-31.66911	0.00	0.00	0.00	0.00	0.00	-1.81	0.00	0.00	1.09	0.91
0.5070	26.11632	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.53	0.00
0.5319	inf	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.5942	inf	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.6242	inf	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.6893	inf	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.7704	inf	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.7992	inf	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.9850	inf	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.9869	inf	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table A.6: Part three of the results of the MLE for the 1D reaction-diffusion equation with ten parameters, hyperparameter optimization for thresholding and one iteration of minimization. Grey cells mark optimal results.

λ	BIC	u	Δu	u^2	$u\Delta u$	$(\Delta u)^2$	u^3	$\Delta^2 u$	1	b_1	b_2
0.000	-33.626	0.88	0.25	0.04	0.00	0.00	-0.85	0.00	-0.02	1.02	0.93
0.012	-44.219	0.86	0.23	0.00	0.00	0.00	-0.88	0.00	0.00	1.02	0.93
0.013	-30.135	-0.41	0.25	0.79	0.02	0.00	0.41	0.00	-0.48	1.02	0.94
0.021	-41.724	0.81	0.23	0.00	0.00	0.00	-0.85	0.00	0.00	1.60	0.00
0.022	-41.724	0.81	0.23	0.00	0.00	0.00	-0.85	0.00	0.00	1.60	0.00
0.041	-44.219	0.86	0.23	0.00	0.00	0.00	-0.88	0.00	0.00	1.02	0.93
0.049	-48.778	0.00	0.23	0.00	0.00	0.00	-0.45	0.00	0.00	1.02	0.93
0.069	-41.724	0.81	0.23	0.00	0.00	0.00	-0.85	0.00	0.00	1.60	0.00
0.074	-44.219	0.86	0.23	0.00	0.00	0.00	-0.88	0.00	0.00	1.02	0.93
0.085	-48.778	0.00	0.23	0.00	0.00	0.00	-0.45	0.00	0.00	1.02	0.93
0.117	-44.219	0.86	0.23	0.00	0.00	0.00	-0.88	0.00	0.00	1.02	0.93
0.134	-41.724	0.81	0.23	0.00	0.00	0.00	-0.85	0.00	0.00	1.60	0.00
0.168	-53.321	0.00	0.24	0.00	0.00	0.00	0.00	0.00	0.00	1.02	0.93
0.177	-44.219	0.86	0.23	0.00	0.00	0.00	-0.88	0.00	0.00	1.02	0.93
0.183	-34.635	0.58	0.23	-0.30	0.00	0.00	-0.43	0.00	1.07	1.02	0.97
0.187	-53.321	0.00	0.24	0.00	0.00	0.00	0.00	0.00	0.00	1.02	0.93
0.192	-53.321	0.00	0.24	0.00	0.00	0.00	0.00	0.00	0.00	1.02	0.93
0.245	-53.320	0.00	0.25	0.00	0.00	0.00	0.00	0.00	0.00	1.02	0.93
0.272	-53.238	0.00	0.27	0.00	0.00	0.00	0.00	0.00	0.00	1.02	0.93
0.306	-48.369	0.33	0.31	0.00	0.00	0.00	0.00	0.00	0.00	1.03	0.93
0.345	-47.567	0.00	0.35	0.00	0.00	0.00	0.00	0.00	0.78	1.02	0.99
0.366	-44.984	1.04	0.00	0.00	0.00	0.00	-2.32	0.00	0.00	1.08	0.91
0.463	-44.984	1.04	0.00	0.00	0.00	0.00	-2.32	0.00	0.00	1.08	0.91
0.507	-49.662	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.70	0.00
0.533	inf	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table A.7: Results of the MLE for the 1D reaction-diffusion equation with ten parameters, BIC and hyperparameter optimization for thresholding after one iteration of minimization. Grey cells mark results closest to expected values, blue cells mark smallest BIC values.

Bibliography

- [1] Uri M. Ascher et al. “Implicit-Explicit Methods for Time-Dependent Partial Differential Equations”. In: *SIAM Journal on Numerical Analysis* 32.3 (June 1995), pp. 797–823.
- [2] Coryn A. L. Bailer-Jones. *Practical Bayesian Inference*. Cambridge: Cambridge University Press, 2017.
- [3] James Bergstra et al. “Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures”. In: *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*. ICML’13. Atlanta, GA, USA: JMLR.org, 2013, pp. I–115–I–123.
- [4] Thomas Blumensath and Mike Davies. “Iterative Thresholding for Sparse Approximations”. In: *Journal of Fourier Analysis and Applications* 14 (Dec. 2008), pp. 629–654. DOI: 10.1007/s00041-008-9035-z.
- [5] Steven L. Brunton and J. Nathan Kutz. *Data-Driven Science and Engineering*. Cambridge: Cambridge University Press, 2019.
- [6] Steven L. Brunton and J. Nathan Kutz. “Methods for data-driven multiscale model discovery for materials”. In: *Journal of Physics: Materials* 2.4 (July 2019), p. 044002. DOI: 10.1088/2515-7639/ab291e. URL: <https://doi.org/10.1088/5C%%202F2515-7639/5C%%202Fab291e>.
- [7] Steven L. Brunton et al. “Discovering governing equations from data by sparse identification of nonlinear dynamical systems”. In: *Proceedings of the National Academy of Sciences* 113.15 (2016), pp. 3932–3937. ISSN: 0027-8424. DOI: 10.1073/pnas.1517384113. eprint: <https://www.pnas.org/content/113/15/3932.full.pdf>. URL: <https://www.pnas.org/content/113/15/3932>.
- [8] Claudio Canuto et al. *Spectral Methods in Fluid Dynamics*. Berlin Heidelberg: Springer-Verlag, 1988.
- [9] Jean Dubé and Diègo Legros. *Spatial Econometrics Using Microdata*. London/Hoboken: ISTE Ltd and John Wiley & Sons, Inc., 2014.

-
- [10] Daniel Foreman-Mackey et al. “emcee: The MCMC Hammer”. In: *Publications of the Astronomical Society of the Pacific* 125.925 (Mar. 2013), pp. 306–312. ISSN: 1538-3873. DOI: 10.1086/670067. URL: <http://dx.doi.org/10.1086/670067>.
- [11] H. S. Greenside and W. M. Coughran. “Nonlinear pattern formation near the onset of Rayleigh-Bénard convection”. In: *Phys. Rev. A* 30.1 (July 1984), pp. 398–428. DOI: 10.1103/PhysRevA.30.398. URL: <https://link.aps.org/doi/10.1103/PhysRevA.30.398>.
- [12] Rudolf Gross and Achim Marx. *Festkörperphysik*. München: Oldenburg Wissenschaftsverlag GmbH, 2012.
- [13] Martin Hermann. *Numerik gewöhnlicher Differentialgleichung*. 2nd ed. Berlin/Boston: De Gruyter, 2017.
- [14] Christopher Honisch. “Analysis of complex systems: From stochastic time series to pattern formation in microscopic fluidic films”. Dissertation. WWU Münster, 2013.
- [15] Oliver Kamps and Joachim Peinke. “Analysis of Noisy Spatio-Temporal Data”. In: *Understanding Complex Systems*. Jan. 2016, pp. 319–324. ISBN: 978-3-319-27633-5. DOI: 10.1007/978-3-319-27635-9_22.
- [16] David Kleinhans. “Estimation of drift and diffusion functions from time series data: A maximum likelihood framework”. In: *Phys. Rev. E* 85 (2 Feb. 2012), p. 026705. DOI: 10.1103/PhysRevE.85.026705. URL: <https://link.aps.org/doi/10.1103/PhysRevE.85.026705>.
- [17] Peter E. Kloeden and Eckhard Platen. *Numerical Solution of Stochastic Differential Equations*. 2nd ed. Berlin Heidelberg: Springer-Verlag, 1995.
- [18] Andreas Löffler and Lutz Kruschwitz. *The Brownian Motion*. Cham: Springer Nature Switzerland AG, 2019.
- [19] Suryanarayana Maddu et al. *Stability selection enables robust learning of partial differential equations from limited noisy data*. 2019. arXiv: 1907.07810 [math.NA].
- [20] J. A. Nelder and R. Mead. “A Simplex Method for Function Minimization”. In: *The Computer Journal* 7.4 (Jan. 1965), pp. 308–313. ISSN: 0010-4620. DOI: 10.1093/comjnl/7.4.308. eprint: <https://academic.oup.com/comjnl/article-pdf/7/4/308/1013182/7-4-308.pdf>. URL: <https://doi.org/10.1093/comjnl/7.4.308>.
- [21] Wolfgang Nolting. *Grundkurs Theoretische Physik 6*. 5th ed. Berlin Heidelberg: Springer-Verlag, 2005.

- [22] Tobias Preis et al. “GPU accelerated Monte Carlo simulation of the 2D and 3D Ising model”. In: *Journal of Computational Physics* 228.12 (2009), pp. 4468–4477. ISSN: 0021-9991. DOI: <https://doi.org/10.1016/j.jcp.2009.03.018>. URL: <http://www.sciencedirect.com/science/article/pii/S0021999109001387>.
- [23] Hannes Risken. *The Fokker Planck Equation*. Berlin Heidelberg: Springer-Verlag, 1996.
- [24] Samuel H. Rudy et al. “Data-driven discovery of partial differential equations”. In: *Science Advances* 3.4 (2017). DOI: 10.1126/sciadv.1602614. eprint: <https://advances.sciencemag.org/content/3/4/e1602614.full.pdf>. URL: <https://advances.sciencemag.org/content/3/4/e1602614>.
- [25] René L. Schilling and Lothar Partzsch. *Brownian Motion*. 2nd ed. Berlin/Boston: De Gruyter, 2014.
- [26] Rainer Storn and Kenneth Price. “Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces”. In: *Journal of Global Optimization* 11 (Jan. 1997), pp. 341–359. DOI: 10.1023/A:1008202821328.
- [27] Robert Tibshirani. “Regression Shrinkage and Selection via the Lasso”. In: *Journal of the Royal Statistical Society. Series B (Methodological)* 58.1 (1996), pp. 267–288. ISSN: 00359246. URL: <http://www.jstor.org/stable/2346178>.
- [28] E. H. Twizell et al. “A second-order scheme for the ”Brusselator” reaction-diffusion system”. In: *Journal of Mathematical Chemistry* 26.4 (Nov. 1999), pp. 297–316. DOI: 10.1023/A:1019158500612. URL: <https://doi.org/10.1023/A:1019158500612>.
- [29] Wen-Xu Wang et al. “Predicting Catastrophes in Nonlinear Dynamical Systems by Compressive Sensing”. In: *Physical review letters* 106 (Apr. 2011), p. 154101. DOI: 10.1103/PhysRevLett.106.154101.

Danksagung

Mein herzlicher Dank geht an Dr. Oliver Kamps, der meine Arbeit betreut hat, die Atmosphäre entspannt hielt und für alle Fragen und Probleme ein offenes Ohr hatte.

Ich bedanke mich auch bei Prof. Dr. Uwe Thiele der die Erstkorrektur übernommen hat und stets interessiert an meinen Fortschritten war.

Auch bei den restlichen Mitarbeitern der AG Thiele möchte ich mich bedanken, die immer freundlich und hilfreich waren, wenn ich Fragen oder Probleme hatte.

Zum Schluss bedanke ich mich bei meiner Familie und meinen Freunden, die mir während meines Studiums und besonders während dieser Arbeit, den Rücken frei gehalten haben und sich als Korrekturleser bereit stellten.

Declaration of Academic Integrity

I hereby confirm that this thesis on Identification of Stochastic Partial Differential Equations from Data is solely my own work and that I have used no sources or aids other than the ones stated. All passages in my thesis for which other sources, including electronic media, have been used, be it direct quotes or content references, have been acknowledged as such and the sources cited.

(date and signature of student)

I agree to have my thesis checked in order to rule out potential similarities with other works and to have my thesis stored in a database for this purpose.

(date and signature of student)