Sparse Learning of Nonlinear PDE Dynamics using Kalman Smoothing

Yash Bhangale

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

Master of Science

University of Washington

2024

Committee:
Aleksandr Aravkin
Santosh Devasia
Krithika Manohar

Program Authorized to Offer Degree:
Mechanical Engineering

University of Washington

Abstract

Sparse Learning of Nonlinear PDE Dynamics using Kalman Smoothing

Yash Bhangale

Chair of the Supervisory Committee:
Aleksandr Aravkin
Department of Applied Mathematics

Identifying differential equations from noisy measurement data requires fitting the underlying dynamics and assimilating the data, either implicitly or explicitly. The Sparse Identification of Nonlinear Dynamics (SINDy) method achieves this in two steps: a derivative estimation and smoothing step, followed by sparse regression over a library of candidate functions. Previous implementations of the derivative step in SINDy, including the Python package `pysindy`, relied on methods such as finite difference, L1 total variation minimization, or Savitzky-Golay filtering. Kalman smoothing, a classical approach for data assimilation with known noise statistics, has recently been incorporated into `pysindy` alongside hyperparameter optimization to enhance data denoising and improve differential equation discovery for several ordinary differential equation (ODE) systems. This thesis extends the use of Kalman smoothing in SINDy to partial differential equations (PDEs), enabling effective denoising and sparse identification of a range of PDE systems.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I wish to express my gratitude to my research advisor and Master's thesis committee chair, Professor Aleksandr Aravkin, for his invaluable guidance throughout this project. I also extend my thanks to my thesis committee members, Professor Santosh Devasia and Professor Krithika Manohar, for their support. Additionally, I am grateful to Jake Stevens-Haas, for his mentorship and assistance throughout my research.

DEDICATION

to my parents and friends and their unwavering support and encouragement

Chapter 1

INTRODUCTION

1.1 Motivation

The Sparse Identification of Nonlinear Dynamics (SINDy) [1, 2] method aims to identify a differential or partial differential equation that governs a system based on temporal measurements. It takes input in the form of coordinate data over time, and selects the most suitable ordinary or partial differential equation from a set of candidate terms. However, this method faces challenges when dealing with substantial measurement noise, which is common in real-world systems. On the other hand, Kalman theory [3, 4] has a half-century history of assimilating measurement noise to smooth a trajectory, with well-studied and rigorously characterized noise properties.

A recent study integrated the well-established Kalman theory with SINDy [5], incorporating generalized cross-validation (GCV) [6] for parameter selection to enable systematic and practical applications on Ordinary Differential Equations (ODEs). The study demonstrates that the Kalman-SINDy framework performs competitively compared to other data smoothing and system identification techniques for a number of Ordinary Differential Equations.

Extending the Kalman-SINDy approach to Partial Differential Equations (PDEs) presents additional challenges and opportunities. PDEs are critical for accurately modeling complex spatiotemporal phenomena found in fields such as fluid dynamics, material science and environmental modeling. Kalman smoothing, like the other smoothing methods discussed in this paper, is fundamentally a one-dimensional smoothing technique. While it is not rigorously designed for smoothing PDEs, there is some numerical optimism that it may still prove beneficial, given that existing alternatives are Finite Difference. This study builds on

the work conducted for ODEs by adapting and extending the Kalman-SINDy approach to accommodate for the added complexity of PDEs.

1.2 Background

Model discovery methods have become essential in advancing data-driven engineering design and scientific exploration. With the support of increased computational capabilities, sophisticated optimization techniques, and machine learning advancements, these methods are transforming how sensor measurements from complex systems are utilized. A key focus in this domain is the discovery of dynamic models, which can be constructed using various approaches. These range from simple regression methods, such as Dynamic Mode Decomposition (DMD) [7, 8], to advanced neural network-based techniques like Physics-Informed Neural Networks (PINNs) [9]. The goal of such models is to develop a representation of the observed data that can characterize and reconstruct system behaviors. While DMD offers an interpretable linear framework through modal decomposition, many neural network models remain opaque, providing limited insights into the underlying dynamics. Among the many emerging approaches [10, 11], SINDy has gained prominence for its ability to generate interpretable and explainable dynamic models. Specifically, SINDy seeks to uncover governing equations from data, identifying relationships between spatial and temporal derivatives.

It is important to note that partial differential equations (PDEs), which govern spatiotemporal systems, introduce significantly greater challenges compared to ordinary differential equations (ODEs). Initially, model discovery techniques focused primarily on ODEs and did not account for spatial derivatives or the complexities of spatiotemporal dynamics. However, these approaches have since evolved to address these limitations. Modern extensions now incorporate spatial dependencies, enabling the discovery of governing equations for PDE systems. This evolution has significantly expanded the scope and applicability of data-driven model discovery methods, allowing them to tackle a broader range of physical and engineering problems. Examples include SINDy [12], Kernel Methods [13], Physics Informed Neural Networks [9, 14], Deep Learning [15, 16], and Genetic Algorithms [17, 18].

The SINDy framework is designed to identify sparse symbolic representations of autonomous or controlled systems, expressed as $\dot{x} = f(x)$. At its core, the method estimates derivatives and robustly identifies relationships between them, building on foundational work from [1, 2]. Weak SINDy [19], an extension of the SINDy framework, addresses the sparse regression problem by integrating data over random control volumes, significantly enhancing the algorithm's resistance to noise. Further robustness to noise and performance with limited data can be achieved through ensembling techniques [20], which employ statistical bagging to estimate inclusion probabilities for the sparse terms (ξ), akin to Bayesian inference [21]. Many of these advancements are incorporated into the open-source library `pysindy` [22, 23], streamlining the application of SINDy methods across diverse problems.

SINDy has two prominent approaches for identifying PDEs: the PDE Library method [12] and Weak SINDy [24]. Each has distinct strengths and limitations. Weak SINDy is particularly robust when dealing with noisy measurement data, as it leverages weak formulations that mitigate the impact of noise on derivative estimation. However, a key drawback of Weak SINDy is that it cannot simulate models, as the discovered equations remain in their weak form and are not readily compatible with traditional simulation frameworks. On the other hand, the PDE Library method is capable of simulating models by identifying equations in a form suitable for numerical solvers. Traditionally, it uses finite difference methods to compute spatial derivatives. While effective in noise-free scenarios, this approach struggles with noisy data, as finite difference calculations can amplify measurement errors.

Kalman filtering and smoothing are foundational techniques in optimal estimation, widely used across various fields such as tracking, navigation, radar, econometrics, and weather prediction. These methods excel at incorporating measurement noise into state estimation. Filtering provides real-time updates to estimates using new measurements, while smoothing refines past estimates by leveraging a complete set of observations (batch processing). A Kalman smoother, in particular, enables the reconstruction of historical state trajectories, enhancing the precision of state estimates through later observations.

A key development in this area is the generalization of classic Kalman updates to sup-

port maximum likelihood or best-fit perspectives. This extension allows for a broad family of efficient Kalman smoothing algorithms [25] that can handle scenarios involving outlier-corrupted signals, nonlinear models, or constraints introduced by external information. Recent advancements have also focused on identifying noise parameters directly from data [6, 26], moving beyond fixed assumptions. While these approaches often introduce hyperparameters and lack guarantees of convergence, they provide robust alternatives to traditional maximum likelihood estimation.

Kalman smoothing has seen significant applications in systems where measurement noise is autocorrelated or varies over time. Adaptations of the method account for these complexities through advanced covariance structures, enhancing its robustness and applicability. Generalized cross-validation (GCV) has emerged as a powerful tool for selecting key parameters, such as the ratio of measurement noise variance to process noise variance, particularly when prior knowledge is limited. By withholding some measurement points during training, GCV evaluates parameter combinations to minimize prediction loss, often producing reliable results in practice.

These innovations have expanded the applicability of Kalman smoothing beyond traditional state estimation, enabling its integration with model discovery techniques like SINDy [5]. By distinguishing between measurement and state variables, Kalman smoothing provides a mechanism for robust derivative estimation in noisy datasets, a critical step in identifying underlying system dynamics.

The objective of this study is to extend the Kalman-SINDy approach to Partial Differential Equations using the PDE Library method. Chapter 2 discusses the theoretical framework of SINDy and Kalman Smoothing. Chapter 3 describes the experimental setup and the experiments. Chapter 4 concludes with ideas for future work.

Chapter 2

THEORY

This chapter is reproduced from the Background section in **Learning Nonlinear Dynamics using Kalman Smoothing**, (J. M. Stevens-Haas, Y. Bhangale, J. Nathan Kutz and A. Aravkin, 2024).

2.1 Sparse Identification of Nonlinear Dynamics (SINDy)

Sparse Identification of Nonlinear Dynamical Systems (SINDy) is a collection of methods designed to uncover the underlying dynamics of systems governed by unknown or partially-known differential equations.

Given some variable of interest $\vec{X}$ and a library of functions Θ (including spatial derivatives, when relevant) SINDy seeks to find the coefficients Ξ of the differential equation:

$$\dot{X} = \Xi\Theta(X) \tag{2.1}$$

where,

$X \in \mathbb{R}^{n\times m} = \vec{x}(t_1)...\vec{x}(t_m)$: system of n coordinates at m timepoints.

$\Theta(X) \in \mathbb{R}^{p\times m}$: library of p functions evaluated at m timepoints

$\Xi \in \mathbb{R}^{n\times p}$: coefficients for n equations of p functions

The function library is written as a time-independent quantity that refers to the collection $\Theta = [\theta_1, \dots \theta_p]^T$, where $\theta_i : \mathbb{R}^n \to \mathbb{R}$. Examples include the family of all degree-2 polynomials of n inputs, mixed sines and cosines of certain frequencies, or a mix of partial derivative and polynomial terms for PDEs, as well as user defined libraries.

The method generally presumes the measurements (Z) faithfully reflect system state (X) and proceeds in two steps:

1. Estimate the state and time derivatives of the system $\widehat{X}, \widehat{\dot{X}} = \mathrm{F}(Z)$ for some smoothing function F.

2. Choosing a sparse regression method, solve the problem $\underset{\text{sparse}\Xi}{\arg\min} \left\| \widehat{\dot{X}} - \Xi\Theta(\widehat{X}) \right\|^2$.

The process of calculating derivatives and applying sparsity in dynamical systems modeling has seen a variety of innovations and approaches. Derivative computation methods are broadly categorized into global methods, such as L1 total variation minimization, and local methods, like Savitzky-Golay smoothing. `pysindy` offers several techniques, including these two, as well as Finite Difference, Smoothed Finite Difference, Spline Derivatives, and Spectral Derivatives.

Sparsity can be applied using methods such as sequentially thresholded linear regression, nonconvex penalties like L0 minimization using relaxation-based techniques, L0 constraints, and Bayesian approaches incorporating prior distributions like spike-and-slab or regularized horseshoe priors. `pysindy` offers several sparse regression methods, like Sequentially Thresholded Least Squares [2], Sparse Relaxed Regularized Regression [27], Mixed Integer Optimized Sparse Regression (MIOSR) [28], Stepwise Sparse Regression [29], and Forward Regression Orthogonal Least Squares.

The Kalman-SINDy method makes SINDy more resilient to noise by taking a data assimilation approach in the following steps:

1. Estimate the state and time derivatives of the system:

$$\widehat{\dot{X}}, \widehat{X} = \underset{X,\dot{X}}{\arg\min}\ \mathrm{L}(X, \dot{X}, Z) = \mathrm{F}(Z)$$

where F applies Kalman smoothing and L is its particular loss function .

2. Choosing a sparse regression method, solve the problem

$$\underset{\text{sparse } \Xi}{\arg\min} \left\| \dot{\widehat{X}} - \Xi\Theta(\widehat{X}) \right\|^2.$$

2.2 Kalman Smoothing

Kalman filtering and smoothing are powerful techniques for state estimation in dynamical systems, particularly when the measurements are corrupted by noise. While Kalman filtering provides real-time updates to state estimates based on sequential measurements, Kalman smoothing refines these estimates by using the complete set of observations (batch processing). This distinction makes Kalman smoothing particularly useful in reconstructing trajectories from noisy datasets.

A Kalman smoother estimates the underlying state trajectory X of a system by minimizing the deviation between the measured variables and the predicted state while maintaining smoothness in the trajectory. The smoothing problem can be expressed as:

$$\underset{X,\dot{X}}{\arg\min} \ \|HX - Z\|^2_{R^{-1}} + \rho\|G[\dot{X}, X]\|^2_{Q^{-1}}, \tag{2.2}$$

where,

$Z =$ measurements of k variables at m time points.

$H =$ a linear transform describing how the process is measured.

$R =$ the covariance matrix of the measurement noise.

$G =$ a linear transform encoding smoothness and state transition dynamics.

$Q =$ the covariance matrix of the process noise.

$\rho = \left(\frac{\sigma_z}{\sigma_x}\right)^2$, the ratio of measurement noise variance σ_z^2 to process noise variance σ_x^2.

The G matrix plays a crucial role in enforcing smoothness across the trajectory. It represents the system dynamics by encoding the consistency of state transitions. For a

single time step, the state transition matrix G^k is defined as:

$$G^k = \begin{bmatrix} I & \Delta t \\ 0 & I \end{bmatrix}, \tag{2.3}$$

where I is the identity matrix of appropriate dimension, and Δt is the time step interval. This matrix governs the evolution of the state $\tilde{X}^k = \begin{bmatrix} X^k \\ \dot{X}^k \end{bmatrix}$ to the next state $\tilde{X}^{k+1}$, satisfying:

$$\tilde{X}^{k+1} = G^k \tilde{X}^k. \tag{2.4}$$

For an entire trajectory of N time steps, the block-structured G matrix encodes transitions across all time steps:

$$G = \begin{bmatrix} -G^1 & I & 0 & \cdots & 0 & 0 \\ 0 & -G^2 & I & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -G^N & I \end{bmatrix}. \tag{2.5}$$

Here, each block row corresponds to a transition constraint $\tilde{X}^{k+1} - G^k \tilde{X}^k = 0$. The first column of each block row applies the state transition matrix G^k to $\tilde{X}^k$, while the second column applies the identity matrix I to $\tilde{X}^{k+1}$. This ensures that the entire trajectory adheres to the dynamics defined by G^k.

The term $\|HX - Z\|^2_{R^{-1}}$ in the cost function measures the deviation between the observed measurements Z and their predicted values HX, weighted by the inverse covariance of the measurement noise. The second term, $\|G[\dot{X}, X]\|^2_{Q^{-1}}$, penalizes deviations from smooth trajectory dynamics, ensuring that state transitions are consistent over time.

To select the parameter ρ, which balances the influence of measurement noise and process noise, generalized cross-validation (GCV) is used. The GCV strategy minimizes the loss on withheld data points, providing an effective way to optimize ρ when prior knowledge of noise levels is limited. Although not guaranteed to find a global minimum, heuristic experience indicates that longer trajectories often lead to better results. By withholding a subset of

data points, GCV optimizes the matrices H, R, G, and Q to produce state estimates $\hat{X}$ that generalize well to unseen data.

2.3 Alternate Methods for Noisy Data Assimilation

Several alternate methods for handling noisy data have been developed, each with unique strengths and limitations. Among these, Total Variation (TV) regularization [30], Savitzky-Golay smoothing [31], and Weak SINDy [19] have garnered significant attention.

Total Variation regularization, originally developed for denoising overhead imagery, minimizes the following objective:

$$\lambda\|\dot{x}\|_1 + \| \int \dot{x} - z\|_2^2,$$

where z represents the measurements, and λ is a smoothing hyperparameter. Similar to Kalman smoothing, TV regularization imposes a regularity condition on the derivative and uses a hyperparameter to balance this regularization with the measurement error term. However, unlike Kalman smoothing, the state in TV regularization is always the direct integral of $\dot{x}$, whereas Kalman smoothing allows for covariance between the state and its derivatives.

Savitzky-Golay smoothing, on the other hand, applies local polynomial regression within a fixed window. This local approach makes it robust to outliers beyond the window width and better suited for nonstationary problems. However, its hyperparameter, the window width, directly impacts runtime, unlike TV and Kalman smoothing.

Weak SINDy offers a compelling alternative by building a regression from the weak form of the differential equations. This approach avoids the need for explicit derivative estimation by satisfying:

$$\begin{aligned} \int \dot{X}\phi(t)\,dt &= \int \Xi\Theta(X)\phi(t)\,dt, \\ -\int X\phi_t(t)\,dt &= \int \Xi\Theta(X)\phi(t)\,dt, \end{aligned}$$

for a test function $\phi(t)$ that vanishes at the integral boundaries. In practice, data space is divided into temporal (or spatiotemporal) bins, effectively averaging away measurement noise. While Weak SINDy is advantageous for noisy data and avoids the need for separate derivative estimation, it cannot utilize a smoothed state in $\Theta(X)$, as Kalman smoothing does. Furthermore, its implementation in `pysindy` remains incomplete, limiting its applicability in certain scenarios.

These methods provide robust alternatives for noisy data assimilation, each tailored to specific requirements and constraints. However, Kalman smoothing stands out for its ability to incorporate both measurement noise and process dynamics cohesively, making it a valuable choice for the experiments described in [5] and this thesis.

Chapter 3

EXPERIMENTS

3.1 Setup

The experiments are designed to be easily reproducible and reusable, with clear separation between the method, experiments, and experiment runner. These components are organized across four distinct packages: `pysindy`, `derivative`, `mitosis` [32], and `pysindy-experiments` [33].

This work enhances the PDE Library method by incorporating smoothed finite difference techniques for spatial derivative computation. This refinement significantly improves the method's robustness in handling noisy data, bridging the gap between noise resilience and simulation accuracy. The incorporation of these techniques expands the PDE Library method's applicability, making it more suitable for real-world systems that involve noisy measurements.

Kalman smoothing is compared to several other noise-mitigation techniques. Two variants of Kalman smoothing are evaluated: a fixed ρ Kalman Smoother and a GCV-optimized Kalman Smoother. These are compared with the Savitzky-Golay Smoother and Total Variation Regularization (TV). Each method is evaluated through a grid search over relevant hyperparameters: λ for TV, ρ for the Kalman Smoothers, and window width for the Savitzky-Golay Smoother. The grid search experiments span five increasing values for both data duration and noise levels, generating distinct data variations for each system.

It is important to note that these denoising techniques are applied exclusively to the temporal axis of the data. For spatial smoothing, smoothed finite differences are employed across all temporal denoising methods.

The grid search experiments are carried out on four well-known partial differential equa-

tion (PDE) systems: the Diffusion Equation, Burgers Equation, Korteweg de Vries (KdV) Equation, and Kuramoto Sivashinsky (KS) Equation. Each PDE is evaluated for a combination of five different data durations and five varying noise levels. This setup results in 25 unique variations of data for each system. Thus, for every PDE, a grid search experiment generates sets of data that span a wide range of temporal and noise conditions.

Since PDE data structures differ from ODE data, modifications were made to both the `pysindy` and `derivative` packages to ensure accurate differentiation across multidimensional data. Initially, data was generated for each grid point, which worked well for ODEs. However, as the complexity of the PDE data grew, it became necessary to separate the data generation process and grid search. This significantly reduced the runtime of the experiment.

The data for each PDE is generated with spectral derivatives from the `pysindy` package applied to ensure that the systems have periodic boundary conditions. The noise is then computed using the maximum amplitude of the PDE data signal in the frequency domain, normalized by the square root of the data length, which in this case is the length of the temporal axis, and scaled according to the input relative noise value r_n. The noise is defined as:

$$\text{noise} = \mathcal{N}(0, \sigma^2), \sigma = \sqrt{\frac{\max(|\text{FFT}(\text{data})|)}{\sqrt{N}} r_n} \tag{3.1}$$

This noise is then added to the true data. After the data generation and noise application, the smoother is applied and smoothed data is generated. The true data, noisy data, and smoothed data are compared for each PDE using the Signal to Noise Ratio (SNR) to compare the true data and noisy data, and the Mean Squared Error (MSE) to compare true data with noisy and smoothed data. SNR is given by

$$\text{SNR} = 10 \cdot \log_{10} \left(\frac{\|\text{true data}\|^2}{\|\text{noisy data} - \text{true data}\|^2} \right) \tag{3.2}$$

Additionally, the corresponding equation identified by SINDy is presented, along with a comparison of the true system coefficients and those estimated by SINDy. This representation reflects one of the many possible configurations, considering different time durations, noise

levels, smoothing hyperparameters, and smoothing types. It is important to note that this may not always represent the most accurate or optimal model.

Furthermore, the performance of the different smoothing methods is evaluated using several metrics. These include Precision, F1 score, Recall, Cofficient Mean Absolute Error (MAE), and Coefficient Mean Squared Error (MSE), calculated across varying time durations and noise levels. These metrics allow for a comprehensive evaluation of each method's ability to recover the true dynamics of the systems, taking into account the impact of both time duration and measurement noise.

By employing these experimental setups, this research provides a robust comparison of various noise-mitigation techniques in the context of SINDy, offering insights into their performance across different PDE systems with varying levels of noise and data availability.

3.2 Results

Figure 3.1 shows the 1-dimensional spatiotemporal representation of the PDE field u with the temporal axis t and spatial axis x. All subsequent PDE data is shown in this format. This figure does not represent 2-dimensional or 3-dimensional PDEs, as they will have more spatial axes (y and z).

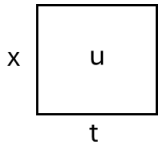

Figure 3.1: Representation of PDE Data

3.2.1 Diffusion Equation

Diffusion Equation is a PDE that exhibits linear dissipative behaviour. It is used to model processes like Heat conduction. Figure 3.2 shows a representation of the diffusion equation which was run for 32 seconds, with a Signal to Noise ratio of 6.25988 and smoothed using the GCV Optimized Kalman Smoother. The underlying equation of this particular smoothed data identified by SINDy is $u_t = 0.054u_{xx}$.

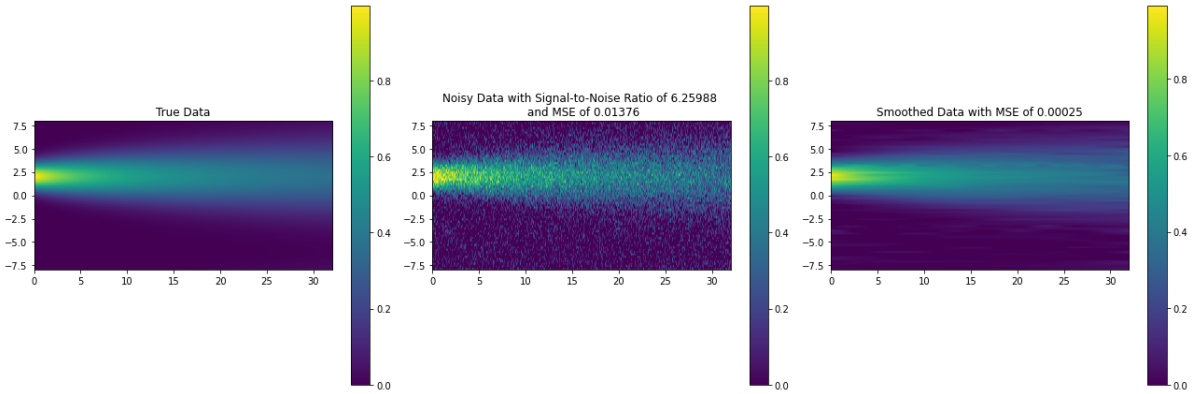

Figure 3.2: Representation of True Data, Noisy Data and Smoothed Data for Diffusion Equation

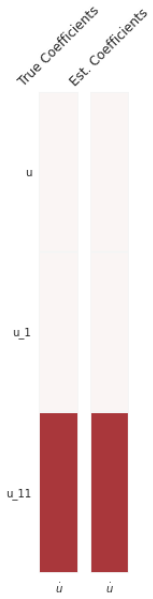

Figure 3.3: Comparison of the true coefficients and the estimated coefficients of the smoothed data from Figure 3.2

Figure 3.3 shows the comparison of coefficients of the model identified by SINDy from the smoothed data in Figure 3.2. For this particular set of time, noise and smoothing method, SINDy identifies the correct coefficient. Figures 3.2 and 3.3 represent only one of numerous experiments throughout the gridsearch, which vary with different time, noise, smoothing method and smoothing hyperparameters. Table 3.1 shows the details of the Diffusion equation, the initial condition utilized to solve the equation on the spatial grid x with n gridpoints and time step of dt for the different t and r_n values. The table also states the hyperparameters that are gridsearched upon for the smoothing methods.

Figure 3.4 shows the full results of the gridsearch, where the evaluation metrics are plotted

Table 3.1: Parameters for Diffusion Equation

PDE Parameters	**Details**
Equation	$\frac{\partial u}{\partial t} = 0.1\frac{\partial^2 u}{\partial x^2}$
Initial Condition	$u(x, 0) = e^{-0.5(x+2)^2}$
Discretization	$x \in (-8, 8)$, $n = 64$, $dt = 0.1$
Time Duration	$t \in$ [4, 8, 16, 32, 64]
Noise Values	$r_n \in$ [0.001, 0.005, 0.01, 0.05, 0.1]
Smoothing Types	Kalman Smoother $\rho \in$ [0.0001, 0.001, 0.01, 0.1, 1] Total Variation Regularization $\lambda \in$ [100, 562.34, 3162.27, 17782.79, 100000] Savitzky Golay Smoother Width $\in$ [5, 8, 12, 15]

against time and noise values for the Diffusion Equation. Both Kalman smoother variants demonstrate superior performance, achieving perfect F1, precision, and recall scores for increasing time duration, indicating they consistently identify the true terms. The GCV-optimized Kalman smoother outperforms the fixed ρ variant in terms of coefficient mean squared error (MSE) and mean absolute error (MAE). Interestingly, while the MSE and MAE initially increase with time due to the accumulation of errors in longer simulations, they eventually decrease as the smoothers leverage the growing data to refine estimates. The Savitzky-Golay smoother generally captures the true terms but incurs significantly higher errors.

The total variation (TV) regularization performs the worst, failing to identify terms in most cases and producing the largest errors. For increasing noise, the GCV-optimized Kalman smoother proves most robust, identifying correct terms more often and

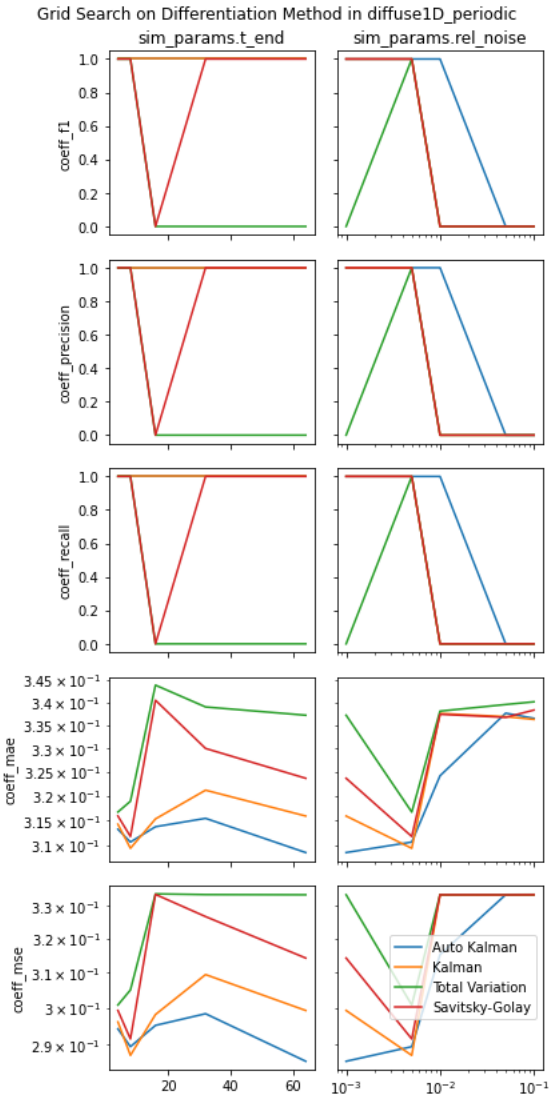

Figure 3.4: Gridsearch Results of the Difusion Equation

maintaining the lowest errors. Other methods degrade more quickly, with the TV regularization particularly sensitive to noise. However, all smoothers fail to sufficiently mitigate the effects of excessive noise, leading to a sharp rise in errors at higher noise levels, with the GCV-optimized Kalman Smoother having gradual increase, in contrast to the others.

3.2.2 Burgers Equation

Burgers Equation is a second order PDE with a quadratic nonlinearity. It is used to model turbulent fluid motion. Figure 3.5 shows a representation of the Burgers equation which was run for 16 seconds, with a Signal to Noise ratio of 7.44292 and smoothed using the GCV Optimized Kalman Smoother. The underlying equation of this particular smoothed data identified by SINDy is $u_t = -0.918uu_x + 0.24uu_{xx}$.

Figure 3.6 shows the comparison of coefficients of the model identified by SINDy from the smoothed data in Figure 3.5. For this particular set of time, noise and smoothing method, SINDy identifies the correct coefficient. Figures 3.5 and 3.6 represent only one of numerous experiments throughout the gridsearch, which vary with different time, noise, smoothing method and smoothing hyperparameters. Table 3.2 shows the details of the

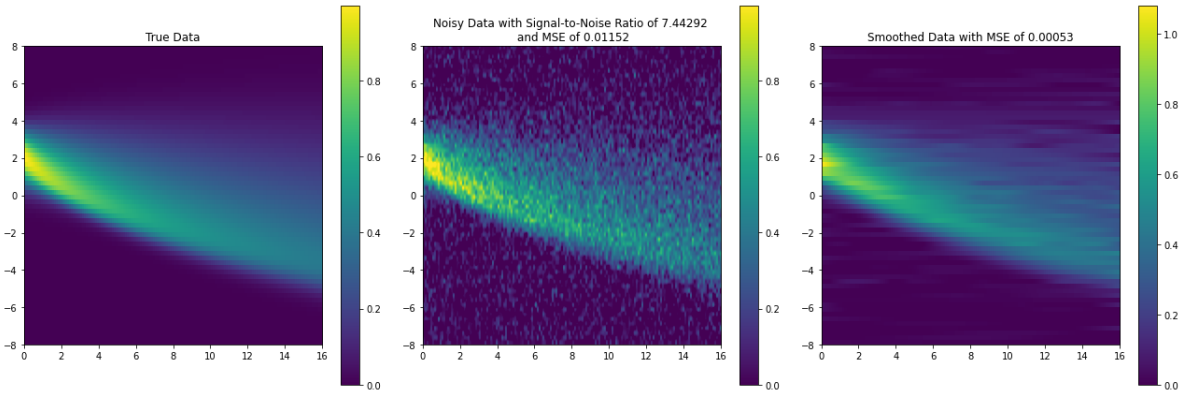

Figure 3.5: Representation of True Data, Noisy Data and Smoothed Data for Burgers Equation

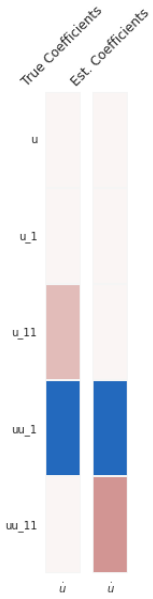

Figure 3.6: Comparison of the true coefficients and the estimated coefficients of the smoothed data from Figure 3.5

Burgers equation, the initial condition utilized to solve the equation on the spatial grid x with n gridpoints and time step of dt for the different t and r_n values. The table also states the hyperparameters that are gridsearched upon for the smoothing methods.

Figure 3.7 shows the full results of the gridsearch, where metrics are plotted against time and relative noise values. The Savitzky-Golay smoother successfully identifies both true terms at one point but fails to do so at other stages, leading to higher overall errors. Although it achieves perfect F1, precision, and recall at that specific instance, its mean absolute error (MAE) and mean squared error (MSE) tend to increase with time, indicating its limited ability to handle noisy data over extended durations. The fixed ρ

Kalman smoother consistently identifies the nonlinear term but struggles with the linear term, showing partial but stable performance throughout.

Table 3.2: Parameters for Burgers Equation

PDE Parameters	**Details**
Equation	$\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x} + 0.1\frac{\partial^2 u}{\partial x^2}$
Initial Condition	$u(x, 0) = e^{-0.5(x+2)^2}$
Discretization	$x \in (-8, 8)$, $n = 64$, $dt = 0.1$
Time Duration	$t \in [4, 8, 16, 32, 64]$
Noise Values	$r_n \in [0.001, 0.005, 0.01, 0.05, 0.1]$
Smoothing Types	Kalman Smoother $\rho \in [0.0001, 0.001, 0.01, 0.1, 1]$ Total Variation Regularization $\lambda \in [100, 562.34, 3162.27, 17782.79, 100000]$ Savitzky Golay Smoother Width $\in [5, 8, 12, 15]$

The GCV-optimized Kalman smoother performs similarly to the fixed ρ smoother but with a notable anomaly at one point where it fails to capture both terms, leading to a sudden spike in errors. Outside of this, the GCV smoother generally exhibits slightly higher errors compared to the fixed Kalman smoother. TV regularization continues to perform the worst, failing to identify the correct terms and showing the highest errors across all time points. When examining the impact of increasing noise, the results follow a similar pattern: both Kalman smoothers, particularly the fixed ρ variant, outperform the other methods, except at the one point when Savitzky Golay smoother identifies both terms. As noise increases beyond a certain threshold, all methods experience a sharp rise in errors, with GCV showing a more pronounced degradation in performance compared to the fixed Kalman smoother and

Savitzky Golay smoother.

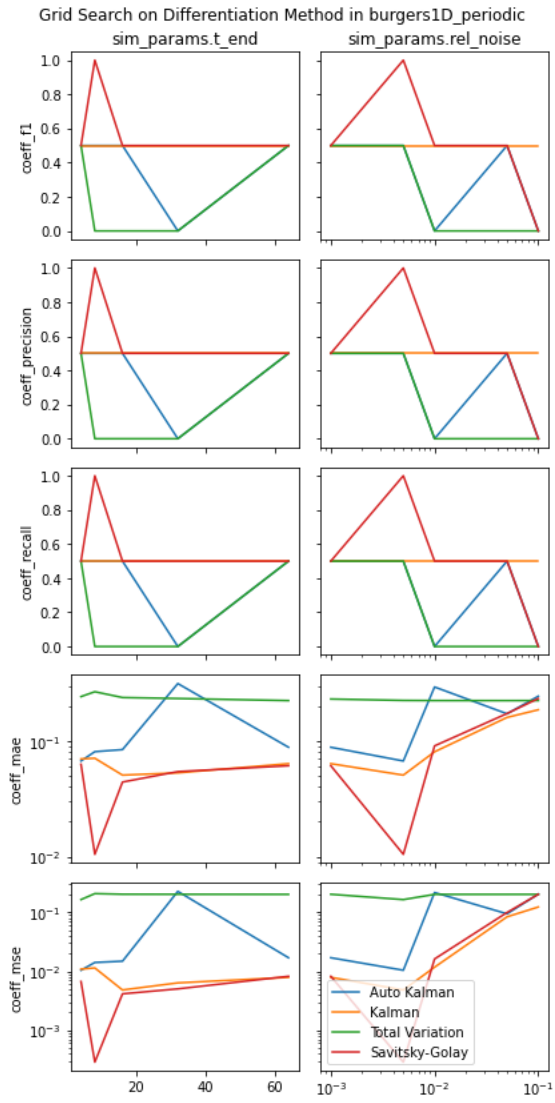

Figure 3.7: Gridsearch Results of the Burgers Equation

3.2.3 Korteweg De Vries

Korteweg de Vries equation is a third order nonlinear PDE. It is used to model waves on shallow water surfaces.

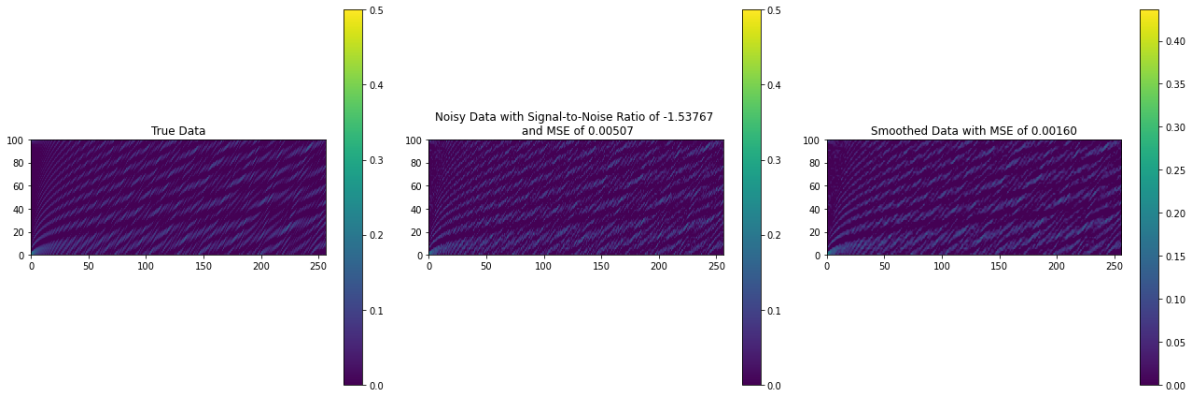

Figure 3.8: Representation of True Data, Noisy Data and Smoothed Data for Korteweg de Vries Equation

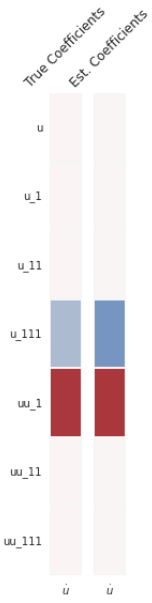

Figure 3.9: Comparison of the true coefficients and the estimated coefficients of the smoothed data from Figure 3.8

Figure 3.8 shows a representation of the Korteweg de Vries equation which was run for 256 seconds, with a Signal to Noise ratio of -1.53767 and smoothed using the GCV Optimized Kalman Smoother. The underlying equation of this particular smoothed data identified by SINDy is $u_t = -0.69u_{xxx} + 1.496uu_x$.

Figure 3.9 shows the comparison of coefficients of the model identified by SINDy from the smoothed data in Figure 3.8. For this particular set of time, noise and smoothing method, SINDy identifies the correct coefficient.

Figures 3.8 and 3.9 represent only one of numerous experiments throughout the grid-search, which vary with different time, noise, smoothing method and smoothing hyperpa-

rameters. Table 3.3 shows the details of the Korteweg de Vries equation, the initial condition utilized to solve the equation on the spatial grid x with n gridpoints and time step of dt for the different t and r_n values. The table also states the hyperparameters that are gridsearched upon for the smoothing methods.

Table 3.3: Parameters for Korteweg de Vries Equation

PDE Parameters	**Details**
Equation	$\frac{\partial u}{\partial t} = 6u\frac{\partial u}{\partial x} - \frac{\partial^3 u}{\partial x^3}$
Initial Condition	$u(x, 0) = 0.5sech^2(x)$
Discretization	$x \in (0, 100)$, $n = 64$, $dt = 0.1$
Time Duration	$t \in$ [16, 32, 64, 128, 256]
Noise Values	$r_n \in$ [0.0005, 0.001, 0.005, 0.01, 0.05]
Smoothing Types	Kalman Smoother $\rho \in$ [0.0001, 0.001, 0.01, 0.1, 1] Total Variation Regularization $\lambda \in$ [100, 562.34, 3162.27, 17782.79, 100000] Savitzky Golay Smoother Width $\in$ [5, 8, 12, 15]

Figure 3.10 shows the full results of the gridsearch, where metrics are plotted against time and relative noise values. For the Korteweg de Vries (KDV) Equation, the fixed ρ Kalman smoother consistently identifies both terms accurately, maintaining perfect F1, precision, and recall across all time points. In contrast, the GCV-optimized Kalman smoother starts strong but exhibits variability as time increases, with performance dropping to 0.5 for certain stages, leading to inconsistent results. The Savitzky-Golay smoother performs similarly to GCV, capturing the correct terms only sporadically, with its performance fluctuating throughout time. TV regularization continues to perform poorly, with F1, precision, and recall values

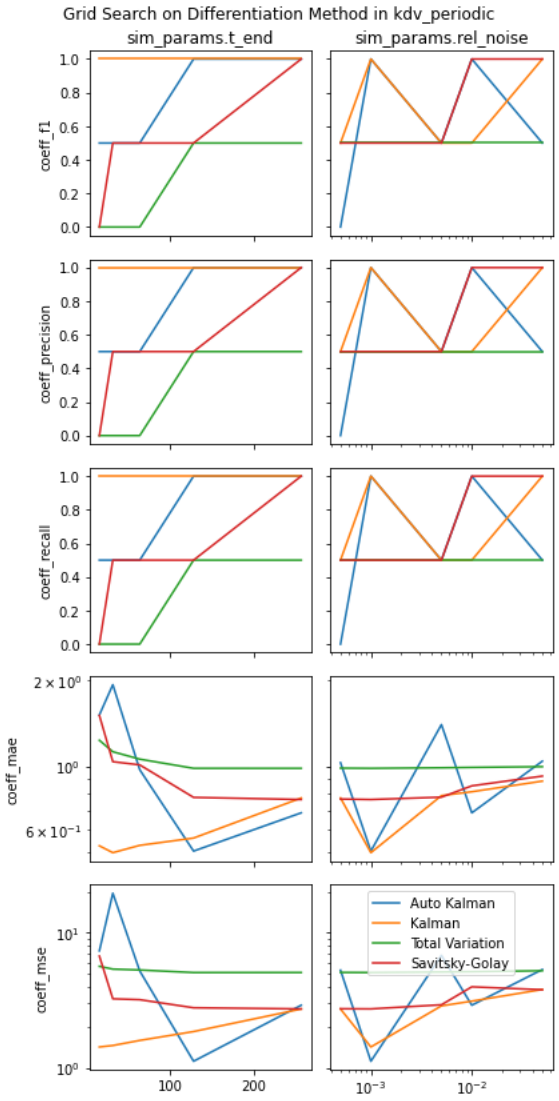

Figure 3.10: Gridsearch Results of the Korteweg de Vries Equation

hovering around 0 for the majority of time steps, reflecting its inability to capture the true dynamics of the system. The GCV Kalman smoother experiences high MAE and MSE errors at the shortest time values, but its error decreases drastically as time progresses, reaching a low point before increasing again. The fixed ρ Kalman smoother, on the other hand, starts with the lowest error but shows a gradual increase as time advances. TV remains consistent with the errors, increasing or decreasing only slightly. Overall, the fixed ρ Kalman smoother provides the most stable and accurate performance across both time and noise regimes.

In terms of noise, the performance of both Kalman smoothers is erratic, with the GCV smoother fluctuating between perfect and poor performance as the noise increases. The fixed ρ Kalman smoother also exhibits irregular results but remains relatively more stable compared to GCV, showing a mix of good and poor performance. The Savitzky-Golay smoother behaves similarly, with its performance swinging between 0.5 and 1 at different noise levels. TV's performance is constant at 0.5 across all noise levels, reflecting its inability to adapt to noise variations. In terms of errors, the GCV Kalman smoother shows erratic trends, while the fixed Kalman smoother achieves the lowest error overall. Savitzky-Golay

follows closely behind, but TV consistently has the highest error, reinforcing its overall poor performance in this context.

3.2.4 Kuramoto Sivashinsky Equation

Kuramoto Sivashinsky is a fourth order chaotic PDE. It is used to model flame propogation. Figure 3.11 shows a representation of the diffusion equation which was run for 160 seconds,

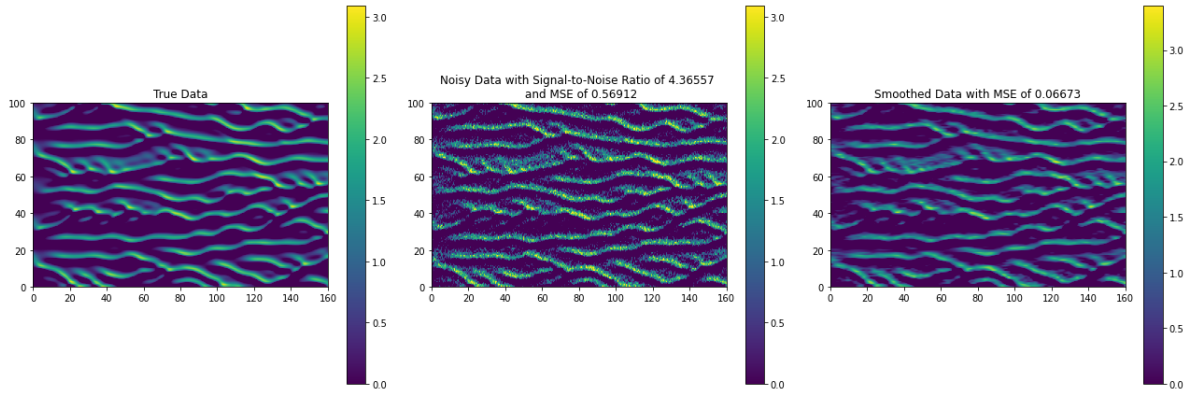

Figure 3.11: Representation of True Data, Noisy Data and Smoothed Data for Kuramoto Sivashinsky Equation

with a Signal to Noise ratio of 4.36557 and smoothed using the GCV Optimized Kalman Smoother. The underlying equation of this particular smoothed data identified by SINDy is $u_t = -0.042u - 0.077u_{xx} - 0.099uu_x$.

Figure 3.12 shows the comparison of coefficients of the model identified by SINDy from the smoothed data in Figure 3.11. For this particular set of time, noise and smoothing method, SINDy identifies the correct coefficient. Figures 3.11 and 3.12 represent only one of numerous experiments throughout the gridsearch, which vary with different time, noise, smoothing method and smoothing hyperparameters. Table 3.4 shows the details of the Kuramoto Sivashinsky equation, the initial condition utilized to solve the equation on the

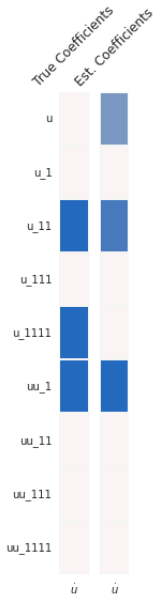

Figure 3.12: Comparison of the true coefficients and the estimated coefficients of the smoothed data from Figure 3.11

spatial grid x with n gridpoints and time step of dt for the different t and r_n values. The table also states the hyperparameters that are gridsearched upon for the smoothing methods.

Figure 3.13 shows the full results of the gridsearch, where metrics are plotted against time and relative noise values. For the Kuramoto-Sivashinsky equation, none of the smoothing methods in conjunction with SINDy were able to correctly identify the higher-order term u_{xxxx}, which led to suboptimal identification of the system dynamics. The F1 score, precision, and recall were consistently lower across all methods, with no smoother achieving perfect performance.

Regarding time evolution, both the GCV and fixed Kalman smoothers displayed similar behavior. The Savitzky-Golay smoother was able to capture the correct terms only at the smallest time point but struggled to maintain this accuracy at later stages. The TV method displayed an unusual pattern, identifying the correct terms only at the first time point and showing good performance at later stages, achieving a better result for the intermediate and later time points.

In terms of errors, both Kalman smoothers followed a similar pattern, with MAE initially decreasing before increasing, with the GCV smoother showing a divergence at later time points. MSE followed a similar trend but showed a sharp increase starting from the second time point for both Kalman smoothers. On the other hand, the Savitzky-Golay smoother exhibited a consistent decrease in MAE but showed an increase in MSE as time progressed. TV, despite identifying the correct terms at later time points, had the highest errors overall.

Table 3.4: Parameters for Kuramoto Sivashinsky Equation

PDE Parameters	**Details**
Equation	$\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x} - \frac{\partial^2 u}{\partial x^2} - \frac{\partial^4 u}{\partial x^4}$
Initial Condition	$u(x,0) = sin(0.1\pi x)(1 + 0.5sin(0.1x))$
Discretization	$x \in (0, 100)$, $n = 128$, $dt = 0.4$
Time Duration	$t \in$ [32, 64, 96, 128, 160]
Noise Values	$r_n \in$ [0.0005, 0.001, 0.005, 0.01, 0.05]
Smoothing Types	Kalman Smoother $\rho \in$ [0.0001, 0.001, 0.01, 0.1, 1] Total Variation Regularization $\lambda \in$ [1000, 3162.27, 10000, 31622.77, 100000] Savitzky Golay Smoother Width $\in$ [5, 8, 12, 15]

For varying levels of noise, the fixed Kalman smoother performed more consistently, maintaining a stable F1 score across all noise levels. In contrast, the GCV Kalman smoother displayed erratic behavior, with fluctuating performance at different noise levels, starting with low performance and then showing improved results at higher noise levels. The Savitzky-Golay smoother showed more variation in performance, with one specific noise level yielding better results. TV had a distinct pattern, performing poorly at the lowest noise level but improving with higher noise values, though its results were still inconsistent compared to other methods.

Regarding errors, the fixed Kalman smoother consistently exhibited the lowest MAE and MSE, with both metrics increasing as noise levels increased. The GCV smoother showed high variation in error but generally remained more consistent than Savitzky-Golay, which exhibited higher MAE errors. TV, however, had the highest MSE errors throughout all noise

levels, reflecting its poor overall performance in comparison to the other smoothing methods.

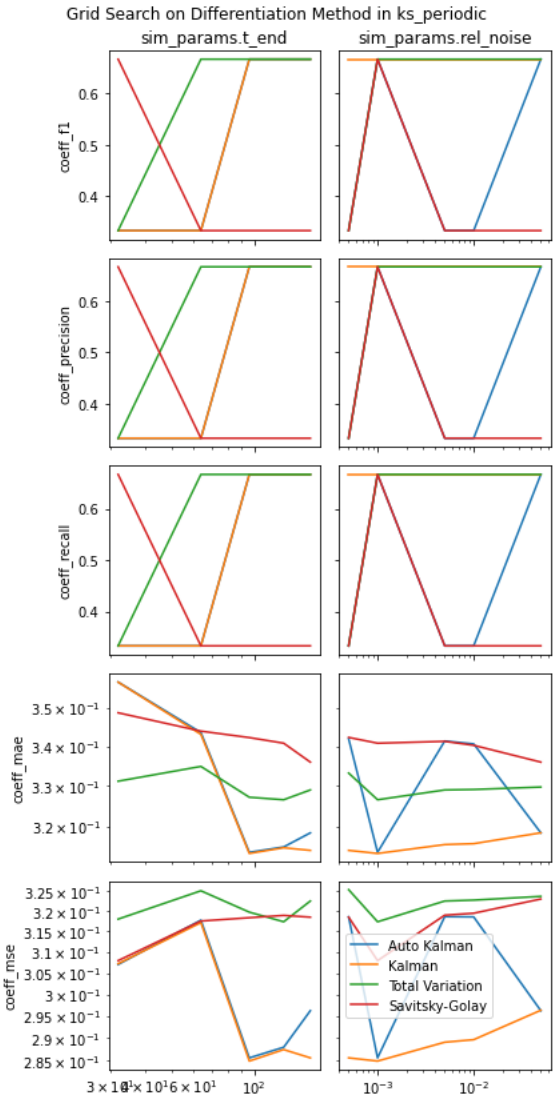

Figure 3.13: Gridsearch Results of the Kuramoto Sivashinsky Equation

Chapter 4

CONCLUSION

The Kalman smoother, particularly with fixed ρ, is a robust and reliable choice for denoising in SINDy, especially when compared to Savitzky-Golay and TV. The GCV-optimized Kalman smoother provides additional flexibility, though it exhibits more erratic behavior in certain cases. Kalman smoothing methods consistently outperform Savitzky-Golay and TV in terms of error metrics like MAE and MSE, and they offer a more consistent approach to identifying the correct terms, making them particularly well-suited for noisy dynamical system discovery tasks. TV, in particular, is not suitable for systems with higher noise levels or more complex dynamics, as it struggles to identify the correct terms and yields high errors across all systems. This aligns with the results of [5].

There are several avenues for enhancing the robustness and efficiency of PDE discovery using SINDy. One promising direction is to implement simultaneous smoothing in both space and time, rather than relying on separate smoothing of the spatial derivatives using smoothed finite differences. This could improve model stability and simulation accuracy, particularly when dealing with noisy data.

Additionally, a key development would be to streamline the process by integrating the derivative estimation and sparse regression steps into a single operation. This would eliminate the current two-step process and enable a more efficient model discovery framework.

While WeakSINDy models are already incorporated into the existing framework and show particular promise in handling noisy measurement data, one limitation is their inability to be simulated directly. Addressing this limitation would be an important step in making WeakSINDy models more versatile for real-world applications.

Enhancing Kalman smoothing techniques is another avenue for future work. Incorporat-

ing advanced variants such as Extended Kalman Smoothers or Unscented Kalman Smoothers could offer improved accuracy, particularly in systems with complex nonlinearities.

Finally, there is a need to explore faster differentiation methods, as the computational burden of adapting existing methods to 2D and 3D PDEs grows significantly. While preliminary efforts were successful in adapting the framework to 2D systems, including a 2D diffusion equation, a 1D 2-field Gray-Scott model, and a 2D 2-field reaction-diffusion equation, the full gridsearch experiments for these higher-dimensional systems proved computationally expensive and were ultimately abandoned.

In conclusion, future work will focus on integrating these advancements to improve the scalability and flexibility of SINDy, enabling more efficient discovery and simulation of PDE models, even in higher-dimensional settings.

BIBLIOGRAPHY

[1] Steven L Brunton and J Nathan Kutz. *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press, 2022.

[2] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113, 2016.

[3] R E Kalman and R S Bucy. New results in linear filtering and prediction theory. *Trans. ASME J. Basic Eng*, 83:95–108, 1961.

[4] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[5] Jacob M. Stevens-Haas, Yash Bhangale, J. Nathan Kutz, and Aleksandr Aravkin. Learning nonlinear dynamics using kalman smoothing. *IEEE Access*, 12:138564–138574, 2024.

[6] Shane Barratt and Stephen Boyd. Fitting a kalman smoother to data, 2019.

[7] J Nathan Kutz, Steven L Brunton, Bingni W Brunton, and Joshua L Proctor. *Dynamic mode decomposition: data-driven modeling of complex systems*. SIAM, 2016.

[8] Sara M. Ichinaga, Francesco Andreuzzi, Nicola Demo, Marco Tezzele, Karl Lapo, Gianluigi Rozza, Steven L. Brunton, and J. Nathan Kutz. Pydmd: A python package for robust dynamic mode decomposition, 2024.

[9] M Raissi, P Perdikaris, and G E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

[10] Salvatore Cuomo, Vincenzo Schiano Di Cola, Fabio Giampaolo, Gianluigi Rozza, Maziar Raissi, and Francesco Piccialli. Scientific machine learning through physics–informed neural networks: Where we are and what's next. *Journal of Scientific Computing*, 92(3):88, 2022.

[11] Joshua S North, Christopher K Wikle, and Erin M Schliep. A review of data-driven discovery for dynamic systems. *International Statistical Review*, 91(3):464–492, 2023.

[12] Samuel H. Rudy, Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Data-driven discovery of partial differential equations. *Science Advances*, 3(4):e1602614, 2017.

[13] Da Long, Nicole Mrvaljevic, Shandian Zhe, and Bamdad Hosseini. A kernel approach for pde discovery and operator learning, 2023.

[14] Hao Xu, Junsheng Zeng, and Dongxiao Zhang. Discovery of partial differential equations from highly noisy and sparse data with physics-informed information criterion. *Research*, 6:0147, 2023.

[15] Hao Xu Hao Xu, Haibin Chang Haibin Chang, and Dongxiao Zhang Dongxiao Zhang. Dl-pde: Deep-learning based data-driven discovery of partial differential equations from discrete and noisy data. *Communications in Computational Physics*, 29(3):698–728, January 2021.

[16] Gert-Jan Both, Subham Choudhury, Pierre Sens, and Remy Kusters. Deepmod: Deep learning for model discovery in noisy data. *Journal of Computational Physics*, 428:109985, March 2021.

[17] Yuntian Chen, Yingtao Luo, Qiang Liu, Hao Xu, and Dongxiao Zhang. Symbolic genetic algorithm for discovering open-form partial differential equations (sga-pde). *Phys. Rev. Res.*, 4:023174, Jun 2022.

[18] Hao Xu, Haibin Chang, and Dongxiao Zhang. Dlga-pde: Discovery of pdes with incomplete candidate library via combination of deep learning and genetic algorithm. *Journal of Computational Physics*, 418:109584, October 2020.

[19] Daniel A. Messenger and David M. Bortz. Weak sindy: Galerkin-based data-driven model selection. *Multiscale Modeling amp; Simulation*, 19(3):1474–1497, January 2021.

[20] U. Fasel, J. N. Kutz, B. W. Brunton, and S. L. Brunton. Ensemble-sindy: Robust sparse model discovery in the low-data, high-noise limit, with active learning and control. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 478(2260), April 2022.

[21] L Gao and J Nathan Kutz. Bayesian autoencoders for data-driven discovery of coordinates, governing equations and fundamental constants. *arXiv preprint arXiv:2211.10575*, 2022.

[22] Alan Kaptanoglu, Brian de Silva, Urban Fasel, Kadierdan Kaheman, Andy Goldschmidt, Jared Callaham, Charles Delahunt, Zachary Nicolaou, Kathleen Champion, Jean-Christophe Loiseau, J. Kutz, and Steven Brunton. Pysindy: A comprehensive

python package for robust sparse system identification. *Journal of Open Source Software*, 7:3994, 1 2022.

[23] Brian de Silva, Kathleen Champion, Markus Quade, Jean-Christophe Loiseau, J. Kutz, and Steven Brunton. Pysindy: A python package for the sparse identification of nonlinear dynamical systems from data. *Journal of Open Source Software*, 5:2104, 5 2020.

[24] Daniel A. Messenger and David M. Bortz. Weak sindy for partial differential equations. *Journal of Computational Physics*, 443:110525, 2021.

[25] A. Y. Aravkin, J. V. Burke, L. Ljung, A. Lozano, and G. Pillonetto. Generalized kalman smoothing: Modeling and algorithms, 2016.

[26] F. van Breugel, J. Nathan Kutz, and B. W. Brunton. Numerical differentiation of noisy data: A unifying multi-objective optimization framework. *IEEE Access*, 2020.

[27] Peng Zheng, Travis Askham, Steven L. Brunton, J. Nathan Kutz, and Aleksandr Y. Aravkin. A unified framework for sparse relaxed regularized regression: Sr3, 2018.

[28] Dimitris Bertsimas and Wes Gurnee. Learning sparse nonlinear dynamics via mixed-integer optimization, 2022.

[29] Lorenzo Boninsegna, Feliks Nüske, and Cecilia Clementi. Sparse learning of stochastic dynamical equations. *The Journal of Chemical Physics*, 148(24).

[30] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259–268, 1992.

[31] Rick Chartrand, L Marin, and D Xiao. Numerical differentiation of noisy, nonsmooth data. *International Scholarly Research Network ISRN Applied Mathematics*, 2011:11, 2011.

[32] Jacob Stevens-Haas. mitosis, April 2024.

[33] Jacob Stevens-Haas and Yash Bhangale. pysindy-experiments, April 2024.