Modelling process dynamics by the discovery of partial differential equations using a data-driven and hybrid modelling approach

by

Kiran Raviprakash

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

Master of Science

in

Process Control

Department of Chemical and Materials Engineering

University of Alberta

Abstract

The abundance of data and advances in data acquisition technologies have made data-driven approaches attractive to solve a multitude of problems. Differential equations deliver underlying models for most physical processes. Obtaining the fundamental physics underlying any data in the form of partial differential equations (PDEs) will facilitate modelling and prediction for systems where first principles modelling might not be feasible. Handling high dimensional spatiotemporal data holds high priority as it forms the underlying basis of many canonical models. Data-driven discovery has been addressed in the literature using Gaussian processes, artificial neural networks, and more recently, sparse regression techniques. Sparse optimization methods are used in multiple domains such as compressive sensing, scientific computing and to learn important features from data sets as they promote parsimony. Although there are multiple works done to discover PDEs using the sparse regression approach, there is no study about the optimal methods that can be utilized for a multitude of systems. We have carried out a detailed study of the sparse regression framework by inferring the best gradient estimation method and the optimal sparsity regularization method for different noise levels. These inferences have provided knowledge about handling uncertainties in the sparse regression framework.

We have utilized these inferences and extended the work to discover a system of PDEs using data-driven and hybrid modelling approaches. The hybrid modelling approach was implemented to utilize the process knowledge to discover the system of PDEs. Through this framework, any partial information about the process can be incorporated into the PDE discovery framework in the form of mathematical constraints. The hybrid modelling

approach improves the model accuracy due to the prior physical knowledge incorporated and also reduces the computational cost.

Petroleum reservoirs are large scale distributed parameter processes from a systems and control theoretic perspective. We have developed an algorithm to discover parametric PDEs to explain the temperature dynamics of a steam-assisted gravity drainage(SAGD) process in an oil reservoir. The data required for the discovery was collected from a first-principles based commercial reservoir simulator and geomechanical simulator CMG-STARS sequentially coupled with FLAC3D. An ensemble of multiple realizations of temperature and permeability was generated which spanned across the spatial domain of the reservoir. A hybrid model was developed by incorporating the permeability values in the discovery framework and has higher accuracy compared to the data-driven model. PDEs were discovered for each realization and then integrated to form a spatially varying parametric PDE explaining the temperature dynamics in an oil reservoir.

Preface

This thesis is an original work conducted by Kiran Raviprakash. The materials presented in this thesis are outcomes of the research projects conducted under the supervision of Dr Vinay Prasad and Dr Biao Huang.

Some parts of Chapter 2 of this thesis were presented as an on-demand oral presentation at the Canadian Society for Chemical Engineering's 70th edition of the Canadian Chemical Engineering Conference (CCEC-2020).

A journal article is being prepared based on Chapter 2 and Chapter 3 of this thesis.- A hybrid modelling approach to discover a system of PDEs. K. Raviprakash, V. Prasad, B. Huang.

A journal article is being prepared based on Chapter 4 of this thesis.- Modelling temperature dynamics of the SAGD process in an oil reservoir. K. Raviprakash, A. Ganesh, V. Prasad, B. Huang.

I was responsible for the idea generation, simulations, and manuscript preparation of the thesis and the works listed above. Dr Vinay Prasad and Dr Biao Huang provided invaluable advice and critical feedback on the technical content and manuscript corrections for all the works. Dr Ajay Ganesh was responsible for the data generated using CMG-STARS and FLAC3D for Chapter 4 and also contributed to the conceptualization in Chapter 4.

Dedicated to my parents

Acknowledgements

First and foremost, I would like to wholeheartedly thank and express my sincere gratitude to Dr Vinay Prasad and Dr Biao Huang for accepting me as a masters student under their supervision. This work would not have been possible without their guidance, support, motivation, and patience. I would like to thank Dr Prasad for being extremely supportive during the whole program, having his door always open for numerous discussions, and for the patience in reviewing my results and drafts. I would like to thank Dr Huang for his invaluable guidance at all stages of the program, and also his support in documentation of the results obtained and keeping up deadlines. I would like to thank Dr Prasad and Dr Huang for the generous financial support throughout my graduate studies and also the support they provided while working from home during the pandemic.

I would like to thank Dr Prasad for giving me an opportunity to serve as teaching assistant for process data analysis and advanced process data analytics courses. It helped me gain confidence and improved my understanding of data analytics. I would like to thank Dr Ajay Ganesh for his guidance and support, especially in developing the contents of chapter 4 of this thesis. I would like to thank Dr Santhosh Kumar Varanasi for his invaluable support in helping me understand various concepts of optimization and control throughout my masters journey. Also, I would like to thank Mr Gokul Subraveti for assisting me in the development of case studies in chapters 2 and 3 of this thesis.

I would like to thank all the research group members of Dr Prasad and the CPC group under Dr Huang. The weekly group meetings and presentations have helped me immensely to improve my knowledge in the field of process control and data science. I thank Dr Sudhir Ranganath and Mr Abhi Kinlekar for motivating me to take up higher studies.

I would like to thank all my dear friends including, but not limited to, Sri, Hesham, Sunad, Karthik, Priyanka, Maithri, and KP from India; Vamsi, Alireza, Sarupa, Meghana, Yousef, Chaitanya, Manju, Deepak, Vinay, Rohit, Rajesh, Sai, Siddarth and many others from Edmonton for everything. Finally, I would like to express my gratitude towards my parents for their encouragement and support throughout my graduate studies.

Contents

List of tables

List of figures

Chapter 1

Introduction

1.1 Motivation and problem statement

The application of data analytics in process control and optimization studies is gaining significant attention due to immense improvement in data collection and storage. The application of machine learning in various fields of engineering emphasises the importance of data storage. Data is utilized to build models that define the propagation of any process and can be used for the prediction of future states. The modelling and prediction framework has been part of process control for many decades. System identification deals with building data-driven models for given process inputs and outputs. Various model structures with different parametrizations and assumptions are assumed and the obtained models are validated. The data-driven models emerged due to complex processes for which the first-principle models were intractable or computationally expensive. The most recent application of the data-driven models can be seen in multiple papers published about modelling the spread of the pandemic in specific regions by considering different factors and making certain assumptions[1][2]. However, the data-driven models have their advantages and limitations. Obtaining the physical interpretability of data-driven models is difficult if the process information is unknown. Various approaches are available in the literature, but are process specific and utilize different statistical techniques to obtain reduced order models or proxy models.

Modelling physical processes or investigation of physical problems has attracted a lot of interest. Many complex processes, such as climate studies, neurosciences, epidemiology are the main areas of study as obtaining a physics-based model for such processes is not feasible. Similarly, in the engineering domain, studying the evolution of parameters of a complex process such as oil reservoirs, blast furnaces, etc. is a challenging task due to the

unavailability of data throughout the spatial and temporal domain of the process. The implementation of data-driven modelling approaches for such processes is not feasible as the generated models will not sufficiently explain the process variability and physical interpretation of the model will be impossible. Differential equations are generally utilised to explain the variability of a particular variable over time or space. Ordinary differential equations (ODEs) model the variation of the process corresponding to time i.e., the evolution of any parameter over different time instants can be modelled using an ODE. Partial differential equations (PDEs) are utilized to model any process that propagates across time and space. Many physical and chemical engineering processes involve spatial variability and evolve over time.

Inferring the model of any parameter using data-driven approaches from processes in which first principles modelling is infeasible is the discovery of differential equations from data. It can also be termed as model identification of dynamical systems. The obtained model has to explain the process variability and also should be parsimonious. For example, developing a model that explains the dynamics sufficiently but having a large number of parameters is not ideal. A similar ideology is utilized in the classic system identification framework as well, where the model selection is based on a trade-off between the number of parameters of the model and model fit percentage. The problem statement can be formulated as the data-driven discovery of parsimonious interpretable differential equations that govern the process under study.

The discovery of physical laws from experimental data in the form of ODEs using symbolic regression was one of the important breakthroughs in the data-driven model discovery domain [3]. Identification of a parsimonious model of an ODE from a predefined candidate library and sparse regression framework was proposed in the sparse identification of nonlinear dynamics(SINDy) [4]. This work led to the emergence of a new area of model identification using the sparse regression framework. This work was extended to the identification of PDEs in the PDE-FIND algorithm [5]. The SINDy and the PDE-FIND algorithms led to various other works in model identification to overcome its limitations. Schaeffer et al. studied the implementation of L1 norm to the PDE discovery[6] instead of the STRidge as used in PDE_FIND. Various other works extended the SINDy algorithm to

stochastic dynamical systems[7], the discovery of biophysical processes, and also to model predictive control [8]. Various other methods were also employed to build ODE and PDE models from data. Bayesian regression[9], the application of convolutional neural networks using constrained filters [10][11], using a feed-forward neural network as a function approximator [12], the application of genetic algorithms[13], and the integration of deep learning and genetic algorithms[14] are some of the noted works in the discovery of differential equations from data. The demonstrations shown in all the works are process-specific and there is no inference provided about the optimal conditions of the algorithm to discover PDEs from any system. As the sparse regression framework is extensively used in the model discovery domain, we studied the impact of different parameters of the algorithm on the model accuracy for different noise levels. We employed the application of different gradient estimation methods and different sparsity regularization techniques and inferred the optimal and consistent methods that will yield a higher accuracy model and is applicable across different systems.

As mentioned earlier, the physical interpretability of data-driven models remains unanswered. Various works in the literature consider the problem of estimating the parameters of a model for a known structure using data. Although these methods are data-driven, the structure of the ODE/PDE is considered to be known and only the coefficients or parameters of the equations are estimated using data-driven methods. The concept of sparse regression is not required in such methods as the number of terms and parameters is known before the discovery. Such methods can be termed grey-box models as only the parameters are estimated for known structures of the PDE. Physics informed neural networks (PINNs) framework proposed a method for forward and inverse solutions of the PDE [15]. Hidden physics models[16] and machine learning of linear differential equations using Gaussian processes [17] proposed by Raissi et al. utilize Gaussian processes to obtain the underlying governing equations.

The methods in the literature are either completely data-driven or assume complete knowledge of the structure of the equation. However, in reality, we end up in situations where data is available and there is some information about the process, but not enough to decide the complete structure of the equation. Therefore, we propose an intermediate

approach that utilizes the data and also incorporates the known information before the PDE discovery. This obtained model will be able to provide physical insights about the process and also explain the process variability. We demonstrate this approach on a system of PDEs.

The form of PDEs generally considered for model discovery contains constant coefficients of the PDE. In reality, there are numerous cases where the coefficients of the PDE are functions of time or space, i.e., the dynamics of the process cannot be explained using constant coefficients. Each term will have a vector of coefficients varying in time or space, depending on the process. The PDE-FIND algorithm was extended with the assumption of parametric PDE by utilizing the group sparsity approach. Regression was performed at each time step if the dependency is on time or spatial location if the dependency is on spatial points [18]. A deep learning framework integrated with a genetic algorithm utilizing a dual neural network framework was employed to discover parametric PDEs[19]. Although these two data-driven methods can discover the PDE in its parametric form, a physics-based approach integrated with the data-driven method will yield interpretable results. We consider the temperature dynamics in an oil reservoir as the subject of study and propose an algorithm to discover the governing parametric PDE, with the spatial dependency of the coefficients. A theoretical perspective of an oil reservoir with its states and parameters is shown in figure 2.1 to provide basic insights into the process[20]. We achieve this by generating an ensemble of realizations across the reservoir using the properties of the variogram and incorporating the information of petrophysical parameters into the discovery algorithm.

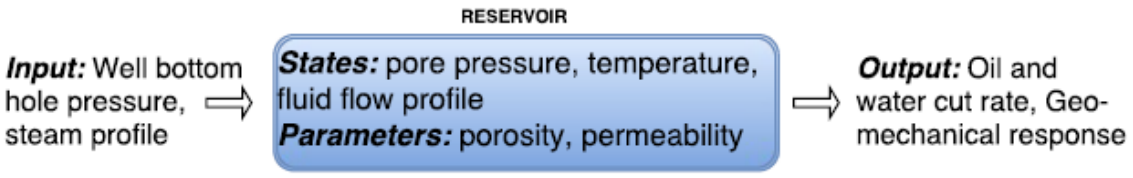

Figure 1.1: Theoretical perspective of an oil reservoir

1.2 Thesis organization and contributions

The remainder of the thesis is organized as follows.

In Chapter 2, the data-driven discovery of partial differential equations for noisy data is performed. The work is based on and an extension of the work presented in [5]. The main contributions of this chapter include a detailed study of the impact of different gradient estimation methods and sparsity regularizations on the model accuracy in the sparse regression framework of the PDE discovery. The best gradient estimation method and the optimal sparsity regularization method that can be utilized to obtain the best PDE models for different levels of noise are developed. We also demonstrate an SVD based denoising framework and compare the model accuracies between with and without denoising the data for different noise levels.

In Chapter 3, we propose a hybrid modelling approach to discover a system of PDEs from spatiotemporal data. The main contributions of this work include the development of a hybrid modelling framework, that incorporates any known process information in the form of mathematical constraints in the PDE discovery framework. The proposed method is an intermediate approach between data-driven and physics informed discovery methods that utilize both data and partially known process knowledge to obtain a parsimonious system of PDEs. We demonstrate the approach on three different complex systems of PDEs. Hybrid modelling for each case study utilizes different forms of process knowledge incorporation. We compare the data-driven and hybrid models for all the case studies.

In Chapter 4, we propose a methodology to model the temperature dynamics of the SAGD process in an oil reservoir. The main contributions of this work include the discovery of a parametric form of PDE to model the temperature dynamics by creating an ensemble of realizations for temperature and permeability data using the properties of the variogram. We utilize data-driven and hybrid modelling approaches to obtain the parametric PDE and compare the model accuracies.

Finally, the conclusions and possible future extensions of this thesis are presented in Chapter 5.

Chapter 2

Data-driven discovery of partial differential equations for noisy data

2.1 Introduction

The abundance of data and advances in data acquisition technologies have made data-driven approaches, an attractive way to solve a multitude of problems and to be applied in different domains. It is estimated that a huge proportion of the available data is generated in the last decade due to the significant plummeting cost of data acquisition through sensors, computational resources, and storage. Less explored is the field of developing physical models that govern the available data exhibiting spatiotemporal behaviour. Partial differential equations(PDEs) facilitate modelling of spatiotemporal data for many processes in science, for example Navier-Stokes' equation in fluid dynamics, Euler's equation in gas dynamics, Maxwell's equation for electromagnetism and in many chemical engineering processes such as heat transfer, mass transfer and adsorption. PDEs also model complex social systems such as population dynamics, financial markets, and epidemic models.

The first-principle based models rely on the inherent physical property and behaviour of the system under study and are available for many processes in science and engineering. Nevertheless, there are complicated systems(for example, climate study, neuroscience, and power grids) for which obtaining the first-principle based models is challenging because of the complexity in the translation of physical properties into variables and the selection of defining variables itself can be a difficult task. Data-driven modelling approaches to encode the governing physical laws in the form of PDEs and using them to make predictions of physical quantities outside the range of the measurement data are gaining significant attention. This broadens the domain of process modelling by eliminating the limitation of a mandatory first-principles model.

Machine learning provides numerous tools to investigate large datasets and to develop models. The discovery of physical laws from experimental data in the form of ordinary differential equations(ODEs) that govern the structure of the phenomena using symbolic regression are proposed by Lipson in [3] and [21]. As the development of parsimonious and interpretable models is of interest, sparse regression framework is most frequently utilized in several works related to data-driven modelling. Sparse regression identifies a few terms that constitute the governing equation from a large candidate library and promotes parsimony. Sparse identification of nonlinear dynamics (SINDy) [22] uses sparse regression to discover ODEs from data and the proposed algorithm in [4] is an extension of SINDy framework which uses an autoencoder framework to discover a coordinate transformation in a reduced space where the dynamics is sparsely represented. The PDE-FIND [5] algorithm by Rudy, et al. and PDE discovery using L1 regularization [6] by Schaeffer, et al. utilize sparse regression framework to discover PDEs from data. Investigation on data-driven discovery using sparse regression has increased significantly since the introduction of PDE FIND [5]. Dynamical system identification using group sparsity[23] demonstrates the discovery of ordinary differential equations from a family of datasets sharing the same physical laws but differs in bifurcation parameters, discovery of stochastic dynamical systems [7] extends the SINDy formulation to stochastic dynamical systems to model biophysical processes, discovery of dynamical systems using AIC criterion[24] concentrates on the model selection for SINDy algorithm using Akaike information criterion, discovery of PDEs is complex datasets [25] utilizes deep learning techniques for identification of governing equations. The identification of nonlinear dynamics with model predictive control [8] is an extension of the SINDy incorporating the actuation effects with a demonstration of enhanced MPC performance, Weak-SINDY uses the Fourier based implementation exploits separability of test functions for efficient mode [26], and deep learning based data driven discovery of PDE [27] integrates deep learning and sparse regression framework to discover PDEs.

Other than sparse regression, neural networks and the Gaussian processes are also used in the discovery of governing equations. Physics informed neural networks(PINN) [15] provides a method for forward and inverse solutions of PDE utilizing automatic differentiation. Hidden physics models [16] and machine learning of linear differential

equations using Gaussian processes [17] proposed by Raissi et al. utilize Gaussian processes to obtain the underlying governing equations. The neural network and Gaussian based methods [15]–[17] require fewer data and are relatively less sensitive to noise, but the limitation is that they only estimate the coefficients of the known PDE structure and not the structure itself. A convolutional neural network-based approach using constrained filters was proposed by Long et.al in [10] and [11]; the PDE structure and coefficients were estimated but the results did not ensure parsimony.

The challenges in most of the above methods are the incomplete candidate library and lack of handling noisy data. DLGA-PDE algorithm proposed by Xu et al.[14] uses a combination of deep learning and genetic algorithms to perform an evolutionary search based PDE structure discovery and regression to estimate its coefficients. Although the issue of the incomplete library is addressed for a few examples, application to industrial data or any other complex processes might not be suitable due to the possibility of genetic algorithm producing a local optimal solution and the possibility of losing the correct terms and coefficients over the generations. The issue of noisy data is generally addressed by using finite difference approximations or cubic splines for interpolation, followed by partial differentiation of the fitted splines. DeepMoD, a deep learning approach, [12] employs a feed-forward neural network as a function approximator and also performs sparse regression within the neural network imposing lasso regularization. Moderate levels of noise and handled well and accurate PDEs are obtained in DeepMoD; however, for higher levels of noise, lasso regularization might not yield good results. The use of polynomial interpolation for gradient estimation is suggested by Rudy et al. in [5] but there are no comparative results displayed. Zhang et al. proposed the sparse Bayesian regression[9] which allows for error bars for each candidate term of the PDE and the idea of transformation of data from temporal to the spectral domain was proposed by Schaeffer in [6]. Both these methods were robust in applicability, but the noise was added only to the temporal derivative and not the data itself. Lagergren et al. proposed a single hidden layer artificial neural network approach for denoising followed by sparse regression for the PDE discovery[28], but the application was limited to first and second-order PDEs; the only process models considered were biological transport models.

The objective of the data-driven discovery of PDE is to obtain parsimonious, interpretable equations that explain the variability in the data and can be used to make predictions with high accuracy. Each process has its properties and irregularities, which makes it difficult to develop a generalized algorithm for all the processes. For example, the PDE model discovered for some processes after denoising the data might not explain the physics of the process due to loss in important information. Learning from all the above-mentioned methods, we can infer that PDE discovery comprises three main components: gradient estimation, denoising, and regularizations imposed on the regression. In this work, we perform a detailed comparative study of all three components for various systems. Different gradient estimation methods such as finite differences, polynomial interpolation, spectral methods, and neural network based automatic differentiation are utilized to estimate gradients to generate the candidate library prior to sparse regression for different physical and chemical engineering processes. The influence of these different gradient estimation methods on PDE discovery at different levels of noise in the input data is studied. Regularization imposed on the regression is another important factor as it promotes parsimony in the discovered model. Furthermore, the impact of various regularizations, such as Lasso, Ridge, and sequential thresholding on the PDE discovery is investigated. Finally, the impact of denoising data before the PDE discovery on the obtained model adequacy is analysed. Model selection is necessary due to hyperparameters in the PDE discovery algorithm. We achieve this through the Akaike information criterion of known systems for the obtained models and infer the most generalizable values of hyperparameters to facilitate application to systems where the ground truth is unknown. PDEs of different orders from various processes are used as case studies to demonstrate the comparative study. Burgers equation, Korteweg–De Vries (KdV) equation, Kuramoto Sivashinsky equation, and PDE from chromatographic studies are the systems under study. Through this study, we can infer the optimal method of gradient estimation, regularization imposed on regression and the impact of denoising data for PDE discovery through sparse regression.

2.2 Data-driven discovery of partial differential equations

2.2.1 Data generation

Spatiotemporal data is the input for the PDE discovery algorithm and all related studies that follow. To validate the method, we generate data by solving a set of known PDEs of different orders from various systems and use this as an input for the PDE discovery. This facilitates us to investigate the robustness of the method involved and facilitates its extension to processes where the ground truth is unknown.

The numerical solutions for the PDEs with pre-defined initial and boundary conditions were obtained using spectral differentiation i.e., fast Fourier transform and inverse fast Fourier transform (*fft* and *ifft* in MATLAB) integrated with Runge-Kutta-45 ODE- *ode45* solver in MATLAB. The temporal discretization, spatial discretization, initial condition, and boundary conditions are different for each system; values are given in the results section. Various levels of Gaussian noise(1% to 20%) are added to the datasets to identify the PDE under influence of noise.

2.2.2 Construction of candidate library

In this work, we discover underlying governing equations for a dataset that is assumed to be the solution to a PDE of the form

$$u_t = F(u, u_x, u_{xx}, u_{xxx}, \ldots\ldots uu_{xx}, u^2u_{xx} \ldots) \tag{2.1}$$

u in (2.1) refers to the spatiotemporal data from which the PDE has to be discovered. The subscripts refer to partial derivatives; u_t refers to the partial derivative of the data with respect to time. u_x refers to the first partial derivative of the data with respect to spatial component and u_{xx}, u_{xxx} ... refers to higher-order partial derivatives with respect to the spatial component respectively. We consider only one spatial dimension in this study. For systems involving multiple spatial dimensions, the partial derivative with respect to all the spatial dimensions will be considered on the right-hand side of (2.1).

The assumption is that the function F consists of only a few terms and is a sum of spatial derivatives, polynomials, non-linear combinations of derivatives and polynomials of the input data u and it is sufficient to explain the process dynamics of the input data. The assumption is based on the structure of the PDEs that are established and utilized in engineering and sciences [29]. This assumption makes F sparse relative to many terms in the candidate library. Our objective is to find the function F by constructing the candidate library followed by sparse regression.

$$U_t = \Theta(U)\xi \tag{2.2}$$

A large library of possible candidate terms that may be a part of F is generated as shown in (2.2). The left-hand side of the equation, U_t, will be a column vector with derivatives of the data with respect to time and the right-hand side is the candidate library matrix Θ containing each possible term in a different column with values at each time step and spatial location as shown in (2.3). ξ is the sparse vector of coefficients. Since Θ contains significantly more terms than the terms necessary to describe the PDE and the terms representing the dynamics are within the span of Θ, most values in ξ will be zero leading to a sparse solution.

$$\Theta(U) = [1 \;\; U \;\; U^2 \;\; U^3 \ldots. \;\; U_x \;\; U_{xx} \;\; U_{xxx} \ldots UU_{xx} \;\; U^2U_{xx} \ldots] \tag{2.3}$$

For example, if we have the data of size $n \times m$, where n represents the number of spatial measurements, m representes the number of time points, and we use d number of candidate terms in the library, then Θ will have $n.m \times d$ entries, and left-hand side will have $n.m \times 1$ entries as shown in (2.4). The library is constructed after estimating the temporal and spatial derivatives of the data at each point in the grid.

$$\begin{bmatrix} u_t(x_0,t_0) \\ u_t(x_1,t_0) \\ \vdots \\ u_t(x_n,t_0) \\ u_t(x_0,t_1) \\ \vdots \\ \vdots \\ u_t(x_{n-1},t_m) \\ u_t(x_n,t_m) \end{bmatrix} = \begin{bmatrix} 1 & u(x_0,t_0) & u_x(x_0,t_0) & \dots & u^3u_{xx}(x_0,t_0) \\ 1 & u(x_1,t_0) & u_x(x_1,t_0) & \dots & u^3u_{xx}(x_1,t_0) \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & u(x_n,t_0) & u_x(x_n,t_0) & \dots & u^3u_{xx}(x_n,t_0) \\ 1 & u(x_0,t_1) & u_x(x_0,t_1) & \dots & u^3u_{xx}(x_0,t_1) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & u(x_{n-1},t_m) & u_x(x_{n-1},t_m) & \dots & u^3u_{xx}(x_{n-1},t_m) \\ 1 & u(x_n,t_m) & u_x(x_n,t_m) & \dots & u^3u_{xx}(x_n,t_m) \end{bmatrix} \begin{pmatrix} \\ \\ \\ \\ \xi \\ \\ \\ \\ \\ \end{pmatrix} \quad (2.4)$$

2.2.3 Gradient Estimation

Evaluation of the temporal and spatial derivatives is the most vital task in PDE discovery through a sparse optimization approach. The adequacy of the obtained model is highly dependant on the accurate gradient estimations. As our objective is to develop a method that can be utilized for PDE discovery for industrial and experimental processes, consideration of noise in the data becomes mandatory. Differentiating noisy data is challenging because of the amplification of noise when the data is differentiated using numerical methods. As we are modelling an unknown system, we are not aware of the highest order of the spatial derivative or polynomial involved in the PDE. Hence, we consider the spatial derivatives up to fourth or fifth-order in the candidate library depending on the complexity of the system to overcome the limitation of an incomplete library. As the order of the derivative increases, effects of noise dominate the numerical derivatives leading to inaccurate values in the candidate library. An extensive study investigating the impact of different gradient estimation methods on PDE discovery for different processes was performed and we selected the best possible method in them that is least sensitive to noise. Finite differences, polynomial interpolation, spectral methods using the fast Fourier transform, and automatic differentiation using neural networks were utilized to estimate the gradients which were compared for different noise levels.

2.2.3.1 Finite differences

The method of finite differences is the most common numerical technique used for gradient estimation. The central difference method is used on the interior points of the grid and the forward difference is applied at the boundaries to find the first-order derivatives [30]. The same rules are applied to estimate higher-order derivatives, but the corresponding previous order derivative values are considered for calculation. The accuracy of the finite difference method may be good for noiseless data but not preferred if the data is noisy. For example, if we have a grid spacing of order h and noise with amplitude of order ϵ, then derivative of order d will have a magnitude of noise approximately of the order ϵ/h^d [5].

2.2.3.2 Polynomial interpolation

Polynomial interpolation is widely used for gradient estimation of noisy data [31]. The derivative of all the data points was approximated by fitting a polynomial of degree p to more than p data points and estimating the derivatives of the polynomial instead of calculating the data derivatives directly. The number of points used for the data fitting and the degree of the polynomial used played a crucial role in the accuracy of the gradients. Also, points close to the boundaries were excluded as it is difficult to differentiate those points using polynomial interpolation. We used the Chebyshev polynomial for fitting and the degree of the polynomial varied for each process. The best result obtained for all the processes is shown in the results section. A predefined function in the NumPy package of Python language was used for fitting the Chebyshev polynomial and estimating the derivatives. A more principled approach for polynomial differentiation is given in [32]

2.2.3.3 Spectral differentiation

Spectral methods involve the transformation of the data into the frequency domain, calculating the derivatives, and transforming the derivatives back to the time domain.

This is performed using the properties of the discrete Fourier transform algorithm(DFT). We transform the data into the frequency domain, and multiply it by $(ik)^d$, where k in the wavenumber and d is the order of derivative to be calculated. Wavenumbers are set based on the number of points and boundary conditions. Multiplication by $(ik)^d$ in the frequency domain is equivalent to taking the derivatives in the time domain [33]. The derivatives are transformed back into the time domain to form the candidate library. The limitation of this method is the data has to be on a periodic domain. We cannot always assume a periodic spatial domain and the Fourier transform cannot be used to obtain the temporal gradients. Finite differences or any other method has to be utilized to obtain the derivatives with respect to time. Spectral differentiation was implemented using predefined functions using the SciPy package in Python- *fft* to transform to the spectral domain, *ifft* to transform to the temporal domain. A command to perform the same operation is also directly available in MATLAB.

2.2.3.4 Automatic differentiation

Neural networks are universal function approximators and are used in various applications of machine learning. Physics informed neural networks [15] introduced utilize neural networks for forward and inverse solutions of PDE followed by the DL-PDE approach.[13]. Gradient estimation through automatic differentiation is executed with the help of neural networks. A feed-forward fully connected neural network was utilized and its structure is shown in figure 2.1.

A neural network consists of an input layer, hidden layers, and an output layer. Two adjoining layers are connected as:

$$z_l = \sigma(W_l z_{l-1} + b_l), l = 1.2, \dots L - 1 \quad (2.5)$$

where l denotes the index of the layer, W denotes the weight matrix, b denotes the bias vector and σ denotes the activation function. The relationship between the input vector z_0, and output z_L can be expressed as:

$$z_L = NN(z_0; \theta) \tag{2.6}$$

θ denotes the set of all weights and biases, that are learnt during the neural network training.

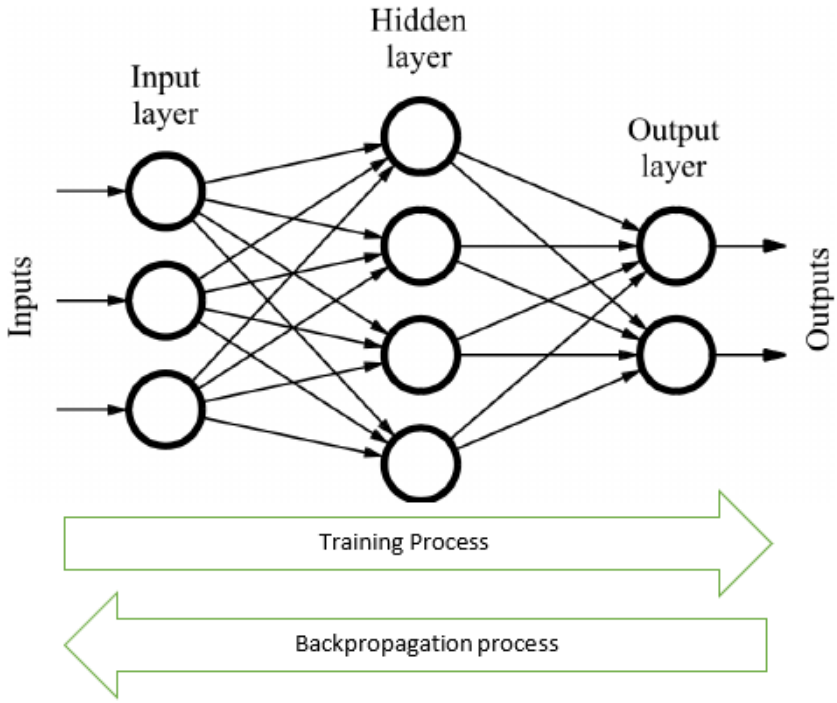

Figure 2.1: Schematic representation of feed-forward fully connected neural network

The first step in obtaining the gradients is training the neural network. The input comprises the spatial and temporal components (x, t) and the output prediction is $u(x, t)$, where u denotes the value at that particular spatial and time stamp, respectively. The loss function of the neural network is defined as:

$$Loss = \sum_{i=1}^{N} [u(x_i, t_i) - NN(x_i, t_i; \theta)]^2 \tag{2.7}$$

where N denotes the total number of points in the data, $u(x_i, t_i)$ represents the input spatiotemporal data and $NN(x_i, t_i, \theta)$ represents the predicted output of the neural network. Adam optimizer is used to minimize the loss function.[34]

After the completion of training the neural network, gradient estimation using automatic differentiation is facilitated by the backpropagation property of neural networks. Automatic differentiation has the advantage of lesser sensitivity to noise compared to numerical methods of differentiation.

2.2.4 Sparse regression

The candidate library Θ as shown in equation (2.4) is assumed to be sufficiently rich spanning across all the possible terms in the PDE. The obtained PDE may be written as a weighted sum of the candidate terms. Each row in (2.4) can be represented as:

$$u_t(x) = \sum_j \Theta\big(u(x,t)\big)\xi_j \tag{2.8}$$

where j represents a particular point on the grid.

We can directly solve the least-squares problem for ξ and obtain the corresponding parameters for each term in Θ. This rules out our objective of obtaining parsimonious PDE models since all the terms in Θ will have a non-zero coefficient. Also, for regression problems as defined in (2.8), least-squares estimation might not be reliable as errors in the gradient computation will be amplified when Θ is inverted. The gradient computation error, the Θ inversion numerical error, and the measurement error have a significant impact on the least-squares estimate. Hence, we use sparse regression as an alternative.

$$\hat{\xi} = argmin_{\hat{\xi}} \big|\big|\Theta\xi - U_t\big|\big|_2^2 + \lambda\big|\big|\xi\big|\big|_0 \tag{2.9}$$

Equation (2.9) assures that only limited terms whose effect on the error $\big|\big|\Theta\hat{\xi} - U_t\big|\big|$ outweigh their addition to $\Big|\Big|\hat{\xi}\Big|\Big|_0$ appear in the PDE. l^0 is imposed in (2.9) which makes the problem *np*-hard. Convex relation of the l^0 norm has to be performed to approximate the solution to (2.9).

$$\hat{\xi} = argmin_{\hat{\xi}} \big|\big|\Theta\xi - U_t\big|\big|_2^2 + \lambda\big|\big|\xi\big|\big|_1 \tag{2.10}$$

Equation (2.10) represents the convex relaxation of the l^0 optimization problem in (2.9). This is also called as l^1norm imposition or least absolute shrinkage and selection operator(LASSO). As mentioned in section 2.1, we have studied the impact of various sparsity regularizations on the PDE discovery.

The method of sequential thresholding coupled with ridge regression is proposed by Rudy et al. in [5]. In sequential thresholding, a hard threshold is placed on the regression coefficients once the predictor is obtained, and this process is repeated recursively on the remaining nonzero coefficients. Ridge regression is an imposition of the l^2 norm to the least-squares problem.

$$\hat{\xi} = argmin_{\hat{\xi}} \left|\left|\Theta\xi - U_t\right|\right|_2^2 + \lambda \left|\left|\xi\right|\right|_2^2 \tag{2.11}$$

(2.11) represents the ridge regression. Combining this with the sequential thresholding mechanism, the resulting algorithm is called Sequential threshold ridge regression or STRidge [5]. For $\lambda = 0$, (2.11) reduces to sequential threshold least squares problem.

The sparsity of the PDE obtained through STRidge is dependant on the threshold tolerance. Hence, a separate method is developed to estimate the best tolerance level. The detailed algorithm of STRidge [5] is shown below.

Table 2.1: Sequential threshold ridge regression algorithm

Algorithm: STRidge ($\boldsymbol{\Theta, U_t, \lambda, iters}$)	
$\hat{\xi} = argmin_{\hat{\xi}} \left\|\left\|\Theta\xi - U_t\right\|\right\|_2^2 + \lambda \left\|\left\|\xi\right\|\right\|_2^2$	#ridge regression
$bigcoeffs = \{j: \left\|\hat{\xi}_j\right\| \geq tol\}$	#selecting large coefficients
$\hat{\xi}[\sim bigcoeffs] = 0$	#apply a hard threshold
$\hat{\xi}[\sim bigcoeffs] = STRidge(\Theta[:, bifcoeffs], U_t, tol, iters - 1)$	#recursive call
return $\hat{\xi}$	

Table 2.2: Best tolerance estimation for STRidge

Algorithm : TrainSTRidge($\Theta, U_t, \lambda, d_{tol}, num_{points}, tol_{iters}, STR_{iters}$)

#Split the data into training and testing sets

$\Theta \rightarrow [\Theta^{\text{train}}, \Theta^{test}]$

$U_t = [U_t^{train}, U_t^{test}]$ } 80/20 split

#Set an appropriate l^0 penalty. (Selected based on empirical evidence)

$\eta = 10^{-3}\kappa(\Theta)$

#Obtaining a baseline predictor

$\xi_{best} = \left(\Theta^{train}\right)^{-1} U_t^{train}$

$error_{best} = \left|\left|\Theta^{test}\ \xi_{best} - U_t^{test}\right|\right|_2^2 + \eta\left|\left|\xi_{best}\right|\right|_0$

#Searching through values of tolerance to find the best predictor

$tol = d_{tol}$

$for\ iter = 1, \ldots., tol_{iters}$:

$\xi = STRidge\left(\Theta^{train}, U_t^{train}, \lambda, tol, STR_{iters}\right)$ #Train and evaluate performance

$error = \left|\left|\Theta^{test}\xi - U_t^{test}\right|\right|_2^2 + \eta\left|\left|\xi\right|\right|_0$

#Is the error still dropping?

$if\ error \leq error_{best}$:

$error_{best} = error$

$\xi_{best} = \xi$

$tol = tol + d_{tol}$

#Or is the tolerance too high?

$else$:

$tol = \max([0, tol - 2d_{tol}])$

$d_{tol} = \dfrac{2d_{tol}}{tol_{iters} - iter}$

$tol = tol + d_{tol}$

$SS_E = \sum[\Theta^{test}\xi - U_t^{test}]^2$;

$S_{yy} = \sum[U_t^{test} - mean(U_t^{test})]^2$

$R^2 = 1 - \frac{SS_E}{S_{yy}}$ #Model adequacy

$rmse = \sqrt{\frac{SS_E}{num_{points}}}$ #Mean squared error

$return\ \xi_{best}, R^2, rmse$

Function arguments given to the STRidge algorithm are $\Theta, U_t, \lambda, iterations$. Along with these, d_{tol} which deciding the step size while looking for optimal tolerance, tol_{iters} indicating the number of recursions to obtain the best tolerance and num_{points} denoting the grid size from the data to calculate the mean square error for model comparison are passed into the TrainSTRidge algorithm.

The elastic-net algorithm that is a linear combination of the LASSO and ridge regression, was also utilized for the PDE discovery.

$$\hat{\xi} = argmin_{\xi} \left|\left|\Theta\xi - U_t\right|\right|_2^2 + \lambda_1 \left|\left|\xi\right|\right|_1 + \lambda_2 \left|\left|\xi\right|\right|_2^2 \quad (2.12)$$

Elastic net is considered to overcome the limitation of the LASSO of underperforming when the variables are highly correlated [35]. The LASSO and ridge regression are special cases of the elastic net. If $\lambda_1 = 0$, then (2.12) will be ridge regression. If $\lambda_2 = 0$, then (2.12) is LASSO.

We compared the LASSO, STRidge, and elastic nets based on the accuracy of the PDE discovered for various systems. The LASSO performed poorly as the columns in Θ are highly correlated. Elastic nets outperformed LASSO but did not yield good results for PDEs containing higher-order derivatives. Although elastic nets provided better results for a few systems, STRidge was able to produce consistently good results for all the systems under study. Detailed analysis of all the systems is given in the results section.

2.2.5 Model selection

Many models can be obtained due to the variation of hyperparameters in the algorithm. There is the possibility that the obtained models are significantly different from each other, which makes it challenging to make a selection. A model selection framework can overcome the disadvantage of the standard method that limits the number of models obtained by reducing the range of hyperparameters. Mangan et al. proposed a framework that utilized the Akaike information criteria(AIC) to hierarchically rank models, enabling

the selection of the model explaining the highest variability in the data [24]. We utilize the same framework to rank the obtained PDE models, developing a more principled approach to obtain a better model. We try to minimize the range of hyperparameters, generalizing the method for the data where the underlying equation is unknown by comparing the obtained models across systems. This also helps to reduce the computational cost for high dimensional systems.

To obtain different models from the sparse regression framework using STRidge, the tolerance value and number of iterations were varied to obtain models of different sparsity levels. Sparsity is altered by decreasing the tolerance values up to 0.1 and increasing the number of iterations to 50. We evaluate all the obtained models using the AIC framework and rank them by the AIC score, and the model with the lowest AIC value is selected as the best model. A demonstration is provided for the different models obtained for the Burgers' equation in the results section. After a detailed study with various systems, we suggest performing the STRidge with 25 iterations to obtain an optimum value for most cases. It was difficult to obtain a definite value for the tolerance level but it was evident that a tolerance level of less than 1 was unnecessary as it did not have a significant impact on the result and would increase the computational cost. The detailed algorithm for model selection is shown below.

Table 2.3: Model selection using AIC criterion

Algorithm – PDE discovery and model selection
Estimate time derivatives and spatial derivatives from data (Y)
Generate the candidate library Θ
For different tolerance levels(d_{tol}) and the number of iterations($iters$):
Model discovery by sparse regression
Simulate the obtained model (X')
Compute AIC between the Y and X'
Sort the models and rank them based on the AIC score
Return the model with the least AIC score.

2.2.6 Denoising data

A practical challenge in the discovery of dynamical systems is the presence of noise in the data. As gradient estimation is a crucial step in the PDE discovery, it is a challenging task to model the PDEs from noisy data as the effect of noise gets amplified during derivative calculation and continues to increase as the order of the derivative increases. Therefore, it is necessary to explore the denoising methods and their influence on the PDE discovery.

We first utilized the proper orthogonal decomposition(POD), which is an equation-based framework to denoise the data using a reduced-order modelling approach [36]. The POD is an extension of the singular value decomposition(SVD) algorithm applied to PDEs. It is a dimensionality reduction technique available to study complex, spatiotemporal systems. POD provides a low-rank representation of the dynamics by the selection of the most dominant modes from the left singular matrix of the SVD of the snapshot matrix developed from the data. We exploit this property of the POD, utilize it to extract the most dominant modes and use them to reconstruct the snapshot matrix. The modes significantly contributing to the process dynamics are extracted and the noise factor is removed through this method. A schematic representation of the POD-denoising algorithm is shown below.

Table 2.4: POD based denoising (model-based framework)

Algorithm: POD based denoising
$X = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots \\ u_1 & u_2 & \dots & u_m \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}$ #Build a snapshot matrix X, rank m
$u_k = [u(x_1, t_k) \;\; u(x_2, t_k) \;\; \dots u(x_n, t_k)]^T$ #Constructed by sampling at a time t_k
$X = U\Sigma V^*$ #Perform SVD on the snapshot matrix
$\tilde{X} = \widetilde{U}\tilde{\Sigma}\widetilde{V^*}$
$\widetilde{U} = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots \\ \psi_1 & \psi_2 & \dots & \psi_r \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}$ #Selected optimal basis modes of rank r, r<<m
$u(t) \approx \psi a(t)$ #$a(t)$ is the low-rank representation and ψ is the set of optimal basis modes.
Reconstruction of the snapshot matrix from $a(t)$ and optimal basis modes.

We added different levels of artificial noise to the generated data for studying the effect of denoising on the PDE discovery. We concluded that denoising aided the discovery of PDE models with higher accuracy compared to that of without denoising case. However, for up to moderate levels of noise, few gradient estimation methods yielded PDE models with the same accuracy without apriori data denoising. For higher noise levels, it is suitable to integrate denoising with the gradient estimation method least sensitive to noise.

As mentioned earlier, the POD algorithm is an equation-based approach and cannot be used directly for data-driven purposes. The POD implementation was performed only to understand the impact of denoising on PDE discovery. To perform denoising for data-driven systems, there are a few established methods in the literature. Denoising using single hidden layer artificial neural networks is proposed in [28] but it is validated only for biological transport systems with 2^{nd} order PDEs. We utilize a framework proposed by Epps and Krivitsky for denoising the data [37]. The noise filtering and reconstruction framework uses the SVD algorithm, calculates the root mean square error(rmse) of the SVD modes and filters the noise using only the SVD modes that have low enough rmse. This method was selected for denoising as we had validated the impact of denoising using POD, which also utilizes the SVD algorithm and Galerkin projection for reconstruction. The difference between POD and this method lies in the selection of the optimal bases modes and procedures used for reconstruction. A brief description of the steps involved in data-driven denoising is mentioned below.

Table 2.5: SVD based denoising (data-driven)

1. Perform SVD on the noisy data matrix.
2. Estimate the measurement error and the spatial correlation parameter by fitting a Marchenko–Pastur distribution to the tail of the noisy singular values[37]
3. Estimate the root mean square error of the modes[38]
4. Estimate the rank for minimum-loss reconstruction[37]
5. Reconstruct an estimate of the clean singular values
6. Reconstruct an estimate of the clean data using clean singular values.

Detailed analysis of the impact of denoising on PDE discovery using different gradient estimation methods is given in the results section.

2.3 Results and discussions

In this section, we demonstrate the methods explained in section 2.2 using classical physical processes which are described by Burgers' equation, Korteweg–De Vries (KdV) equation, Kuramoto Sivashinsky equation, and a chromatographic process. All the PDEs in this section are represented using the format in equation (2.1).

2.3.1 Burgers' equation

The 1-D Burgers' equation is a second-order non-linear partial differential equation simulating the propagation and reflection of shock waves [39]. It is derived from the Navier Stokes equation for the velocity field. It is used in various applications of fluid dynamics, nonlinear acoustics, and applied mathematics. The 1-D Burgers' equation is given by,

$$u_t = -uu_x + au_{xx} \tag{2.13}$$

where a is the diffusion coefficient, considered 0.10 in this case for data generation. The u_{xx} term in (2.13) prevents the shock formation, unlike the inviscid Burgers' equation ($u_t + uu_x = 0$).

Data is generated by simulating (2.13) with initial condition $u(0,x) = -\sin\left(\frac{\pi x}{8}\right), x\epsilon[-8,8]$ and periodic boundary conditions with timestamps from 0 to 10. The number of spatial and temporal points after uniform discretization is 256 and 101 respectively. We consider the highest order of the polynomial and spatial derivative in the candidate library to be 4.

For polynomial interpolation, 4th order Chebyshev polynomial is used and 10 points are used for fitting. We used a higher-order polynomial to generalize the method for all systems. A 3rd order polynomial will also yield the same results for Burgers' equation. To facilitate automatic differentiation, a nine-layer deep neural network with 20 neurons per hidden layer was utilized; tanh(x) was used as the activation function. The PDE discovered using different gradient estimation methods for different levels of noise is shown in tables 2.6 to 2.9.

To demonstrate the model selection algorithm, we altered the sparsity by changing the tolerance values up to 0.1 and increased the number of iterations to 50. The different models obtained and their respective AIC values are shown in table 2.10.

Table 2.6: PDE discovered using finite differences

Gradient estimation method: Finite differences							
Correct PDE : $u_t = -uu_x + 0.1u_{xx}$							
	u	u^2	u_x	uu_x	u_{xx}	u^2u_{xx}	u^3u_{xxx}
Clean				-1.002	0.102		
1%				-0.985	0.115		
2%				-0.974	0.119		
5%			-0.008	-0.921	0.104		
10%			-0.015	-1.354	0.068	-0.125	-0.014
20%	0.257	-0.357	-0.142	-0.912	0.027	-0.258	

Noise level

Table 2.7: PDE discovered using polynomial interpolation

Gradient estimation method: Polynomial interpolation							
Correct PDE : $u_t = -uu_x + 0.1u_{xx}$							
	u	u^2	u_x	uu_x	u_{xx}	u^2u_{xx}	u^3u_{xxx}
Clean				-1.003	0.101		
1%				-0.998	0.108		
2%				-0.986	0.096		
5%				-0.971	0.115		
10%			-0.009	-1.125	0.074		
20%		-0.147	-0.0078	-1.944	0.038	-0.0987	0.00015

Noise level

Table 2.8: PDE discovery using spectral methods

	Gradient estimation method: Spectral differentiation						
	Correct PDE : $u_t = -uu_x + 0.1u_{xx}$						
	u	u^2	u_x	uu_x	u_{xx}	u^2u_{xx}	u^3u_{xxx}
Clean				-0.989	0.107		
1%				-1.214	0.119		
2%				-0.841	0.072		
5%				-0.789	0.138		0.0087
10%			-0.0145	-0.758	0.074		0.0076
20%	-0.089	-0.254	-0.0014	1.358	0.041	-0.087	0.00023

Noise level

Table 2.9: PDE discovery using automatic differentiation

	Gradient estimation method: Automatic differentiation						
	Correct PDE : $u_t = -uu_x + 0.1u_{xx}$						
	u	u^2	u_x	uu_x	u_{xx}	u^2u_{xx}	u^3u_{xxx}
Clean				-1.002	0.107		
1%				-1.003	0.119		
2%				-0.998	0.072		
5%				-0.974	0.115		
10%				-0.854	0.063		
20%			-0.0157	1.415	0.071	-0.0041	

Noise level

STRidge algorithm was used in studying the impact of gradient estimation methods on PDE discovery for different levels of noise based on the literature survey [5], [13], [28]. As we

can observe from tables 2.6 to 2.9, it is evident that automatic differentiation outperforms the other three methods in presence of noise. Polynomial interpolation is less sensitive for lower and moderate noise levels, but coefficients start to diverge and wrong terms appear for relatively higher noise levels. We utilized two model adequacy metrics to assess the performance of the obtained models: the coefficient of multiple determination(R^2) and root mean square error(RMSE). R^2 is a measure of amount of variability of the data explained by the model. Root mean square error is the square root of the variance of the residuals. It indicates the absolute fit of the model to the data. The lower the RMSE, the better the model fits the data. R^2 is a relative measure of fit, whereas RMSE is an absolute measure of the fit.

$$SS_E = \sum_{i=1}^{n} [\Theta^{test}\xi - U_t^{test}]^2 \; ; \; \Theta^{test}, \xi, U_t^{test} from\ table\ 2.2 \tag{2.14}$$

$$S_{yy} = \sum_{i=1}^{n} [U_t^{test} - mean(U_t^{test})]^2 \tag{2.15}$$

$$R^2 = 1 - \frac{SS_E}{S_{yy}}, = \frac{Explained\ variability}{total\ variability} \tag{2.16}$$

$$RMSE = \sqrt{\frac{SS_E}{num_{points}}} \tag{2.17}$$

2.3.1.1 Comparison of obtained PDE models with and without data denoising

The model adequacy report for the study shown in Tables 2.6 to 2.9 is shown in figures 2.2 and 2.3. The same procedure for PDE model discovery using different gradient estimation methods was followed after denoising the data using the method explained in section 2.2.6. Figures 2.4 and 2.5 present the model adequacy report of PDE discovery after denoising. The results clearly depict that for higher levels of noise, denoising data is preferable but for

lower and moderate levels of noise, using automatic differentiation will provide parsimonious interpretable results. Also, for systems where there is a possibility of information loss by denoising, automatic differentiation is preferable over other methods as it is able to provide good results even in the presence of noise as shown in figures 2.2 and 2.3.

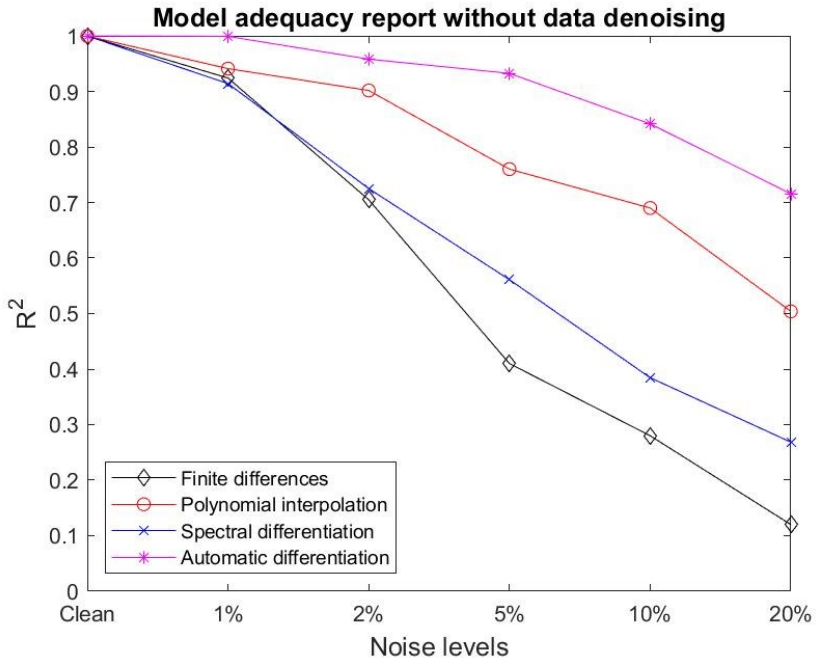

Figure 2.2: Model adequacy report (R^2) for different gradient estimation methods without denoising(Burgers' equation)

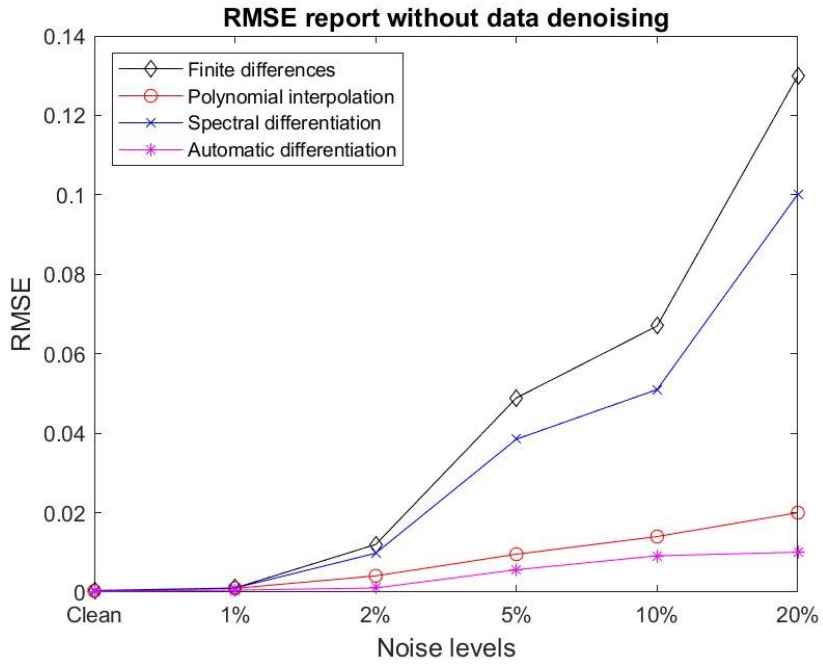

Figure 2.3: RMSE report for different gradient estimation methods without denoising(Burgers' equation)

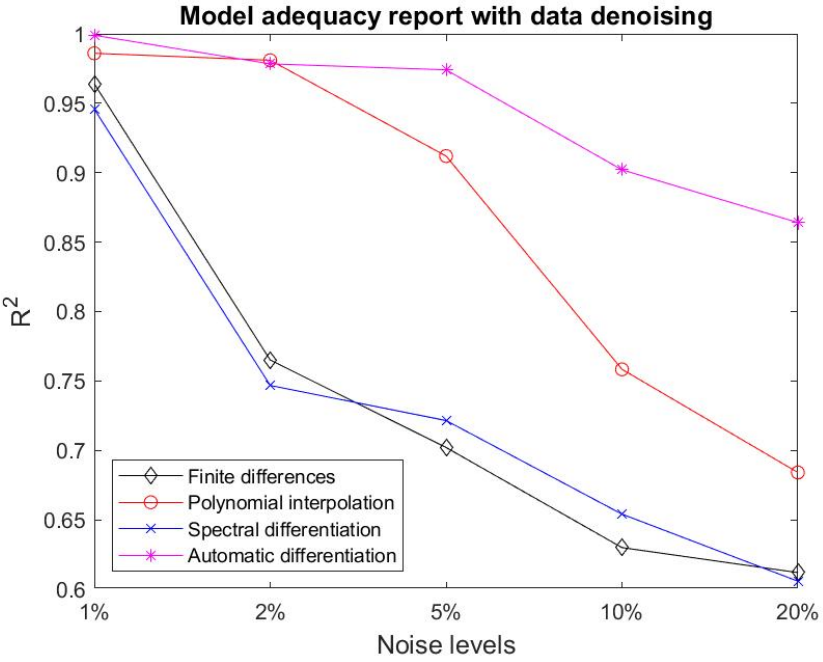

Figure 2.4: Model adequacy report(R^2) for different gradient estimation methods with denoising(Burgers' equation)

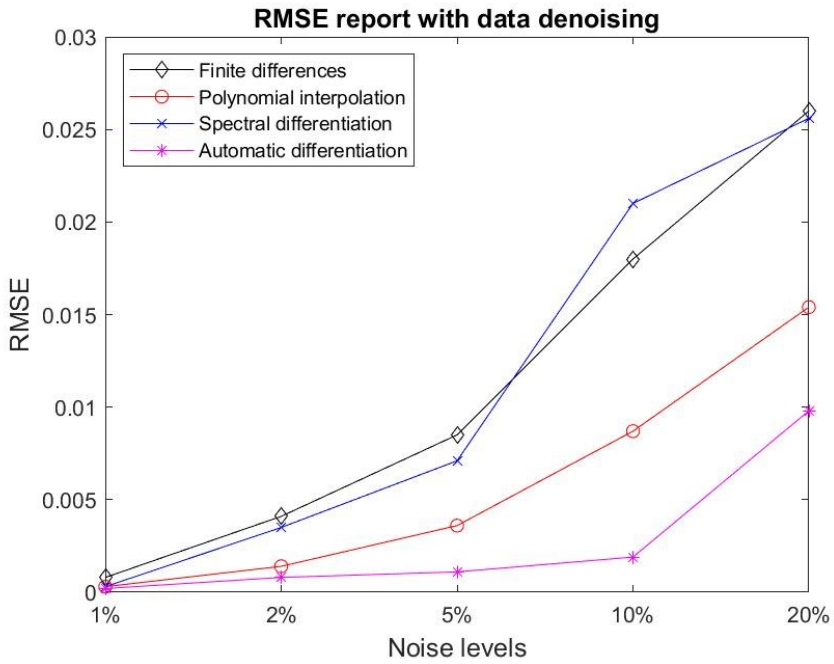

Figure 2.5: RMSE report for different gradient estimation methods with denoising(Burgers' equation)

Comparing the above figures, we can infer the following:

1. Automatic differentiation aids in obtaining the best PDE model compared to the other method of gradient estimation for noisy data.
2. Denoising has a significant impact in improving the PDE discovery if any method other than automatic differentiation is used for gradient estimation.
3. For higher levels of noise, denoising may be preferred even if automatic differentiation is utilized for PDE discovery.

2.3.1.2 Comparison of obtained PDE models for different sparsity regularizations imposed.

As explained in section 2.2.4, different sparsity regularizations were imposed to the sparse regression framework and obtained models were analysed to determine the most applicable method to the PDE discovery framework. LASSO, STRidge, and elastic nets were utilized to discover the PDE models. Based on the analysis in section 2.3.1.1, automatic differentiation was used for estimating gradients and denoising was performed for data having noise >10%. Since no significant difference was observed in the R^2 *and* $RMSE$

values across the obtained models for different noise levels, average values of both the metrics are plotted and presented in figures 2.6 and 2.7.

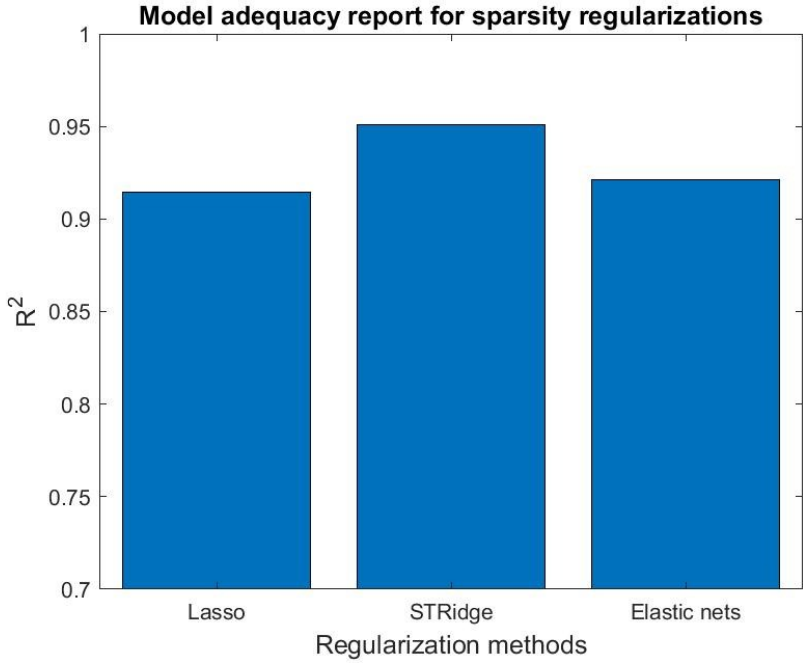

Figure 2.6: Model adequacy report (R^2) for different regularization methods(Burgers' equation)

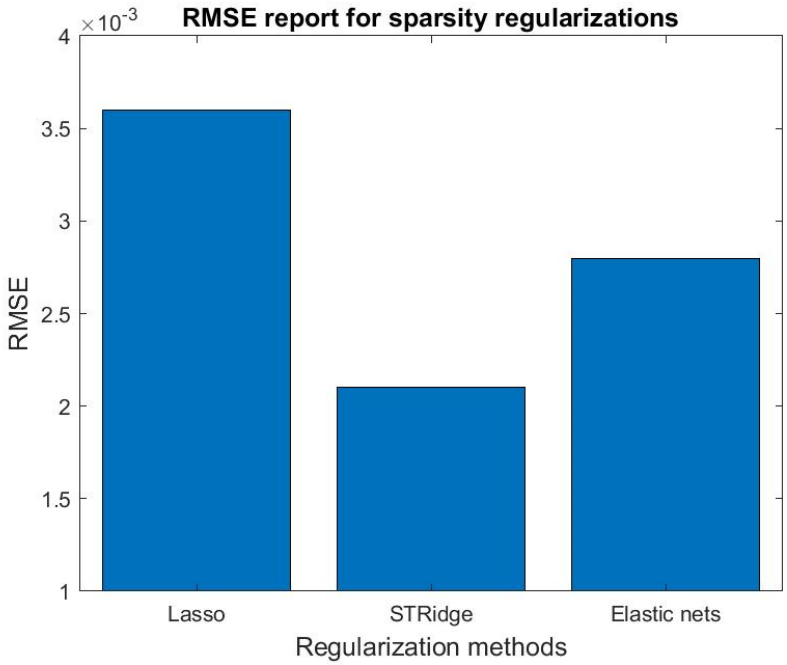

Figure 2.7: RMSE reports for different regularization methods(Burgers' equation)

From the above plots, we can infer that STRidge provides the best model compared to LASSO and elastic nets. The performance of LASSO is based on the correlation between columns of Θ. If the columns are highly correlated, as is true in our case, LASSO tends to perform poorly. Elastic nets overcome this limitation and may provide the best results for a few systems but cannot be considered as robust as STRidge. Model adequacy reports for other systems are presented in other sections 2.3.2 to 2.3.4.

2.3.1.3 Model selection

The tolerance values and number of iterations are varied in the STRidge algorithm to alter the sparsity of the obtained equation. A different model is obtained for different values of tolerance and iterations. The algorithm utilized is explained in section 2.3. We decrease the tolerance values up to 0.1 and increase the number of iterations up to 50. The AIC values were computed for each model and sorted. The model with the least AIC value was selected. The list of models obtained and their AIC values are shown in table 2.10.

Table 2.10: Discovered models and their AIC score(Burgers' equation)

Discovered PDE	AIC score
Correct PDE : $u_t = -uu_x + 0.1u_{xx}$	--
$u_t = +(0.027)u - (0.109)u^2 + (0.131)u^3 - (0.010)u_x - (1.010)uu_x + (0.535)u^2u_x - (0.656)u^3u_x + (0.067)u_{xx} + (0.248)uu_{xx} - (0.485)u^2u_{xx} + (0.343)u^3u_{xx} - (0.005)u_{xxx} + (0.029)uu_{xxx} - (0.020)u^2u_{xxx} - (0.027)u^3u_{xx}$	333

$ut = +(0.022)u - (0.084)u^2 + (0.100)u^3 - (1.160)uu_x + (0.630)u^2u_x - (0.677)u^3u_x + (0.070)u_{xx} + (0.237)uu_{xx} - (0.462)u^2u_{xx} + (0.317)u^3u_{xx} - (0.005)u_{xxx} + (0.037)uu_{xxx} - (0.048)u^2u_{xxx}$	354.6
$ut = -(1.074)uu_x + (0.386)u^2u_x - (0.446)u^3u_x + (0.085)u_{xx} + (0.125)uu_{xx} - (0.262)u^2u_{xx} + (0.186)u^3u_{xx} + (0.026)uu_{xxx} - (0.043)u^2u_{xxx}$	360.25
$ut = -(1.022)uu_x + (0.087)u_{xx} + (0.096)uu_{xx} - (0.097)u^2u_{xx}$	351.28
$ut = -(1.064)uu_x + (0.571)uu_{xx} - (0.593)u^2u_{xx}$	641
$ut = -(1.010)uu_x + (0.103)u_{xx}$	0

2.3.2 Korteweg–De Vries (KdV) equation

The KdV equation is a third-order nonlinear PDE. It is an asymptotic simplification of Euler equations used to model waves in shallow water [40]. It is an extension of the Burgers' equation with a dispersive term. It was discovered by Korteweg and de Vries when studying small amplitude and longwave motion in shallow water. The equation is given by:

$$u_t = -\alpha u u_x - u_{xxx} \tag{2.18}$$

where α is a constant and is assumed to be 6 in our case for data generation. Data is generated by simulating (2.18) with initial condition $u(0,x) = -\cos(\pi x), x\epsilon[-30,30]$ and periodic boundary conditions with timestamps from 0 to 20. The number of spatial and

temporal points after uniform discretization is 512 and 201 respectively. We consider the highest order of the polynomial and spatial derivative in the candidate library to be 4.

For polynomial interpolation, 4th order Chebyshev polynomial is used and 10 points are used for fitting. To facilitate automatic differentiation, a five-layer deep neural network with 50 neurons per hidden layer was utilized; sin(x) was used as the activation function as it was found to be more stable than tanh(x) for this case.

We performed PDE discovery using different gradient estimation methods for different noise levels. The process was repeated after denoising the data as well. A similar observation as in section 2.3.1 was obtained. Automatic differentiation outperformed the other three methods and was found to be the optimal method among the studied approaches. The PDE discovered from using automatic differentiation after denoising for different noise levels is presented in table 2.11. The methodology used is explained in section 2.3.1.

Table 2.11: PDE discovered for different noise levels(KdV equation)

Correct PDE : $u_t = -6uu_x - u_{xxx}$	
Noise level	Discovered PDE
Clean data	$u_t = -5.977uu_x - 1.005u_{xxx}$
1% noise	$u_t = -6.174uu_x - 0.974u_{xxx}$
2% noise	$u_t = -6.18uu_x - 1.124u_{xxx}$
5% noise	$u_t = -5.737uu_x - 0.976u_{xxx}$
10% noise	$u_t = -6.981uu_x - 1.547u_{xxx}$
20% noise	$u_t = -5.871uu_x - 0.874u_{xxx} + 0.002u^2u_x$

2.3.2.1 Comparison of obtained PDE models with and without data denoising

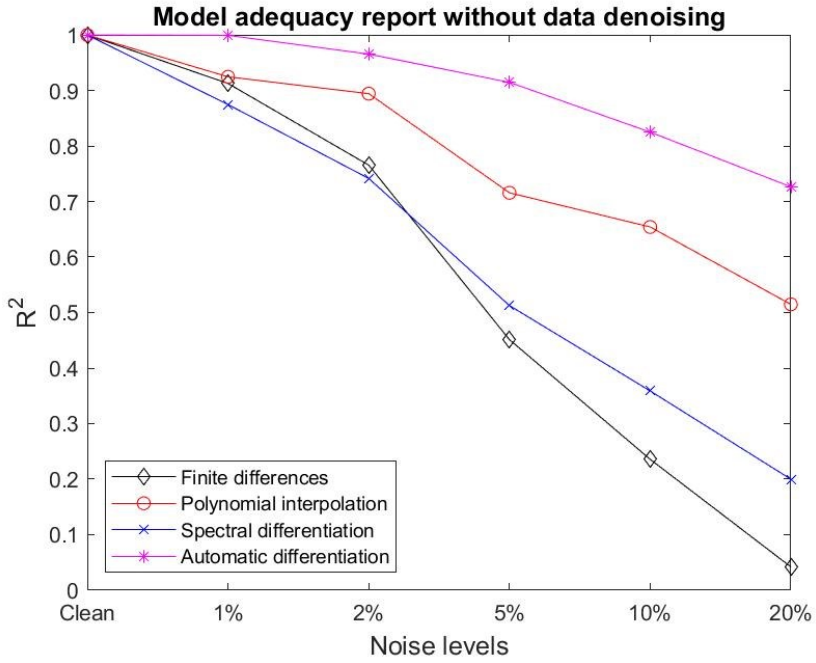

Figure 2.8: Model adequacy report(R^2) for different gradient estimation methods without denoising(KdV equation)

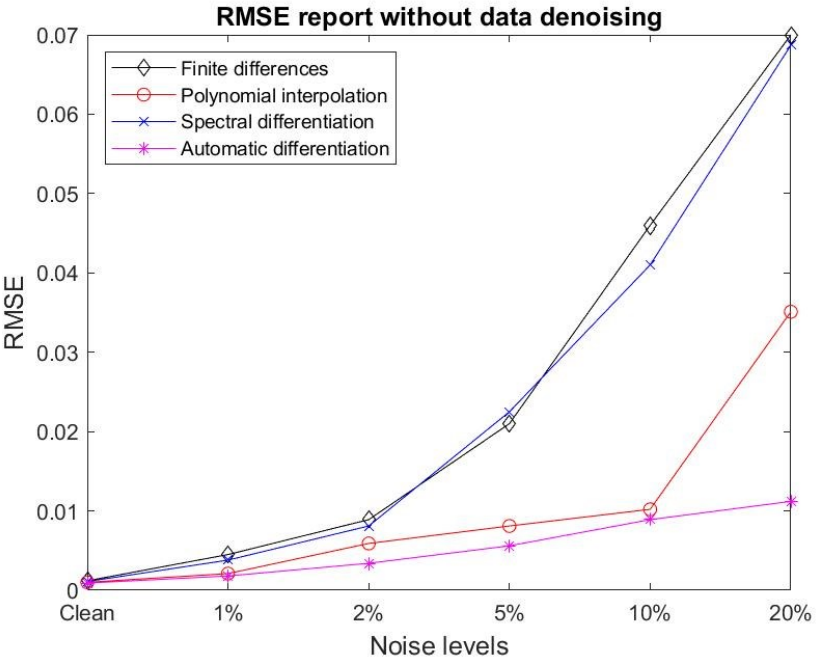

Figure 2.9: RMSE report for different gradient estimation methods without denoising(KdV equation)

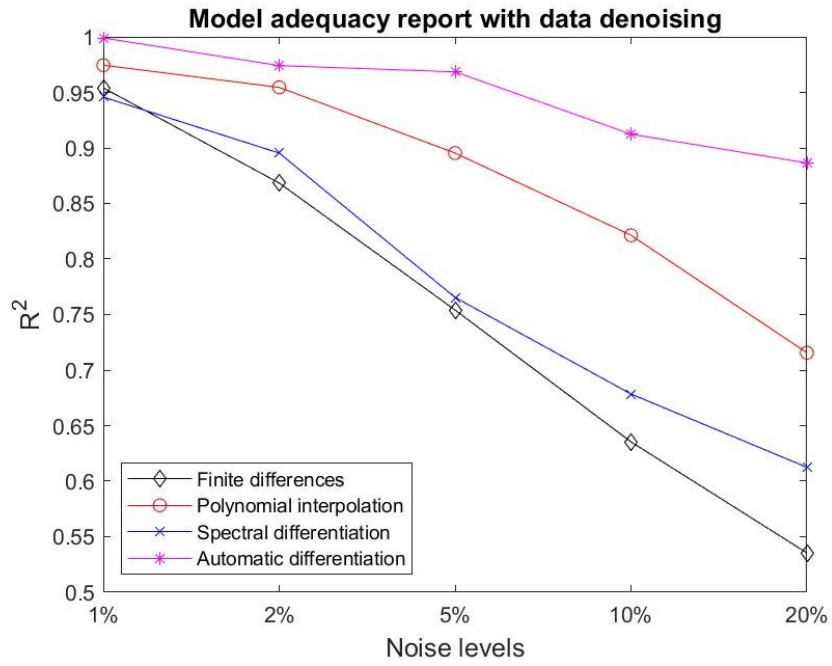

Figure 2.10: Model adequacy report(R^2) for different gradient estimation methods with denoising(KdV equation)

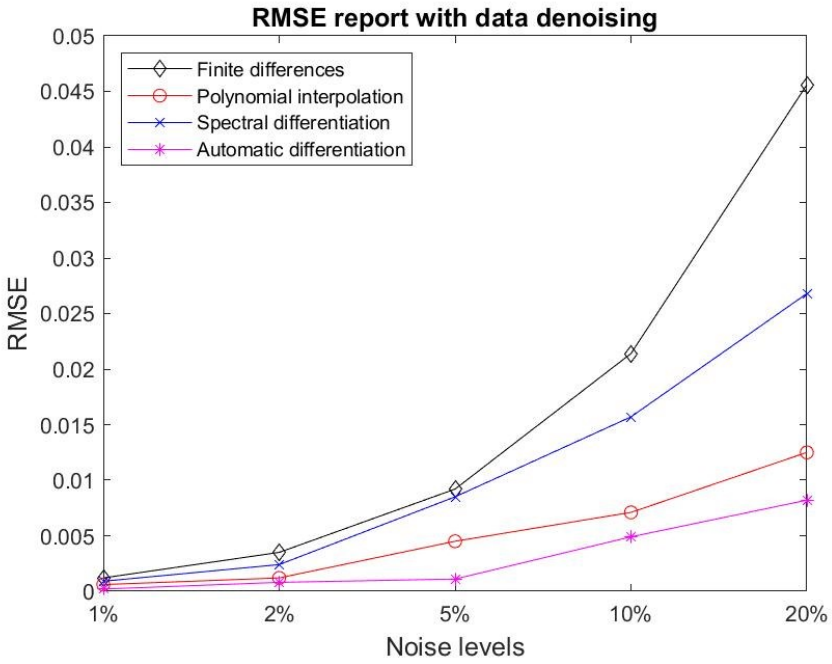

Figure 2.11: RMSE report for different gradient estimation methods with denoising(KdV equation)

Analysing figures 2.8 to 2.11, we can validate the results obtained for Burgers' equation. Automatic differentiation provides optimal results in the case of the KdV equation as well. Here, we discovered a third-order non-linear PDE through a sparse optimization approach using different gradient estimation methods. The complexity of PDE discovery increases as the order of the PDE increases since the possibility of numerical errors in gradient estimation and regression increases.

2.3.2.2 Comparison of obtained PDE models for different sparsity regularizations imposed.

As explained in section 2.2.4, different sparsity regularizations were imposed to the sparse regression framework and obtained models were analysed to infer the most applicable method to the PDE discovery framework. LASSO, STRidge, and elastic nets were utilized to discover the PDE models. Based on the analysis in section 2.3.2.1, automatic differentiation was used for estimating gradients and denoising was done for data having noise >10%. Since no significant difference was observed in the R^2 and $RMSE$ values across the obtained models for different noise levels, average values of both the metrics are plotted and presented in figures 2.12 and 2.13.

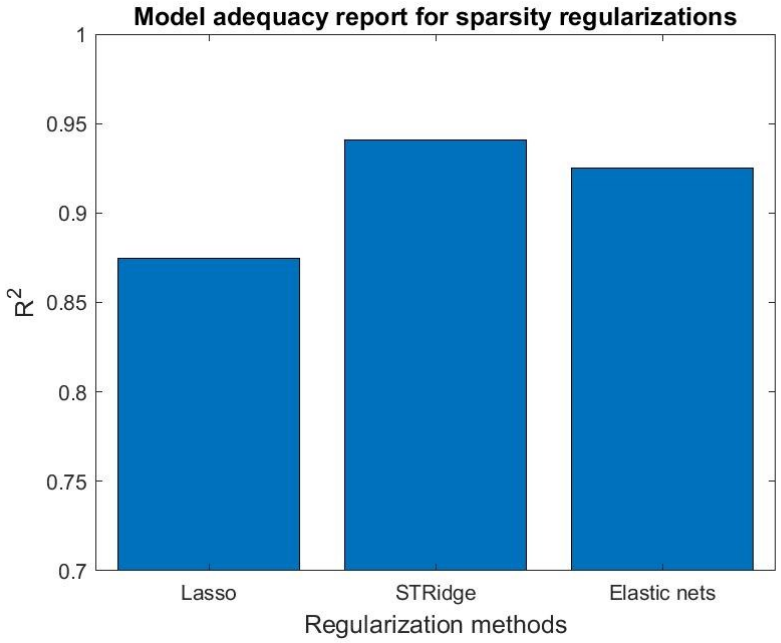

Figure 2.12: Model adequacy report (R^2) for different regularization methods(KdV equation

From figures 2.12 and 2.13, we can observe that elastic nets and the STRidge, both provide comparable results in the case of the KdV equation whereas LASSO does not perform well. The STRidge is the most consistent method when compared to LASSO and elastic nets. This analysis is revalidated in sections 2.3.3 and 2.3.4 where we study the Kuramoto Sivashinsky equation and a chromatographic system respectively

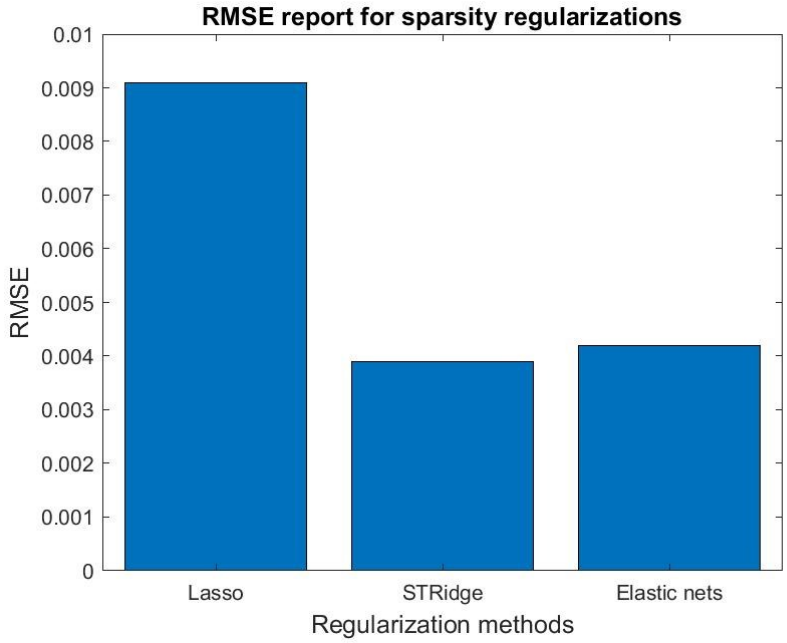

Figure 2.13: RMSE reports for different regularization methods(KdV equation)

2.3.2 Kuramoto Sivashinsky equation

The Kuramoto Sivashinsky(KS) equation is a fourth-order PDE. It is utilized in the context of multiple physical systems driven far from equilibrium by intrinsic instabilities [5]. It is considered to be a canonical model of pattern-forming systems with chaotic behaviour. KS equation provides a diffusive regularization similar to the Burgers' equation given by $u_t + uu_x = 0$. Stabilization is achieved by the fourth order diffusion term as the second order diffusion corresponds to backwards diffusion equation. The KS equation is given by,

$$u_t = -uu_x - u_{xx} - u_{xxxx} \tag{2.19}$$

Data is generated by simulating (2.19) using spectral methods. The number of spatial and temporal points after uniform discretization is 1024 and 251 respectively. We consider the highest order of the polynomial and spatial derivative in the candidate library to be 5 to prevent the problem of an incomplete library. PDEs governing systems having an order greater than 3 is unusual but not impossible. Generally, considering a candidate library of order 4 are sufficient for most of the systems but we can include higher orders if we have any prior information about the process or if we are aware of any known similar physical system with a governing PDE of higher order.

For polynomial interpolation, a 4th order Chebyshev polynomial is used and 10 points are used for fitting. To facilitate automatic differentiation, a nine-layer deep neural network with 20 neurons per hidden layer was utilized; tanh(x) was used as the activation function.

We performed PDE discovery using different gradient estimation methods for different noise levels. The process was repeated after denoising the data as well. Models obtained for different noise levels using the automatic differentiation after denoising the data is shown in table 2.12. The methodology used is explained in section 2.3.1.

Table 2.12: PDE discovered for different noise levels(KS equation)

Correct PDE : $u_t = -uu_x - u_{xx} - u_{xxxx}$	
Noise level	Discovered PDE
Clean data	$u_t = -0.989uu_x - 0.995u_{xx} - 0.998u_{xxxx}$
1% noise	$u_t = -0.745uu_x - 0.812u_{xx} - 0.761u_{xxxx}$
2% noise	$u_t = -0.712uu_x - 0.845u_{xx} - 0.709u_{xxxx}$
5% noise	$u_t = -1.210uu_x - 0.5874u_{xx} - 0.6415u_{xxxx}$
10% noise	$u_t = -0.451uu_x - 0.481u_{xx} - 0.492u_{xxxx}$
20% noise	$u_t = -0.381uu_x - 0.411u_{xx} - 1.234u_{xxxx} - 0.0087u^2u_{xx}$

2.3.3.1 Comparison of obtained PDE models with and without data denoising

The performances of finite differences, polynomial interpolation, and spectral differentiation are found to be worse for high levels of noise in the case of the KS equation compared to Burgers' and KdV equations due to the presence of a fourth-order derivative, and calculation of 5^{th} order derivative to construct the candidate library.

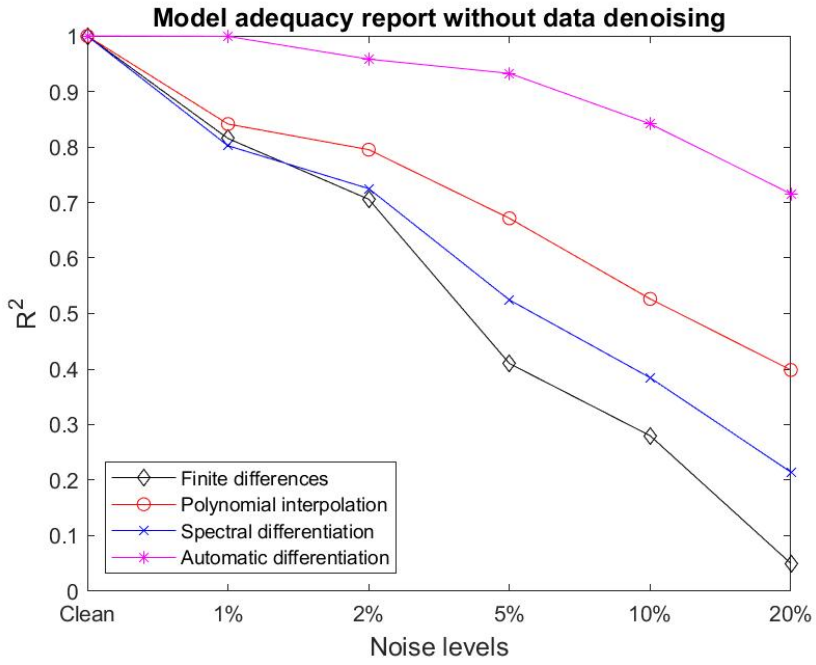

Figure 2.14: Model adequacy report(R^2) for different gradient estimation methods without denoising(KS equation)

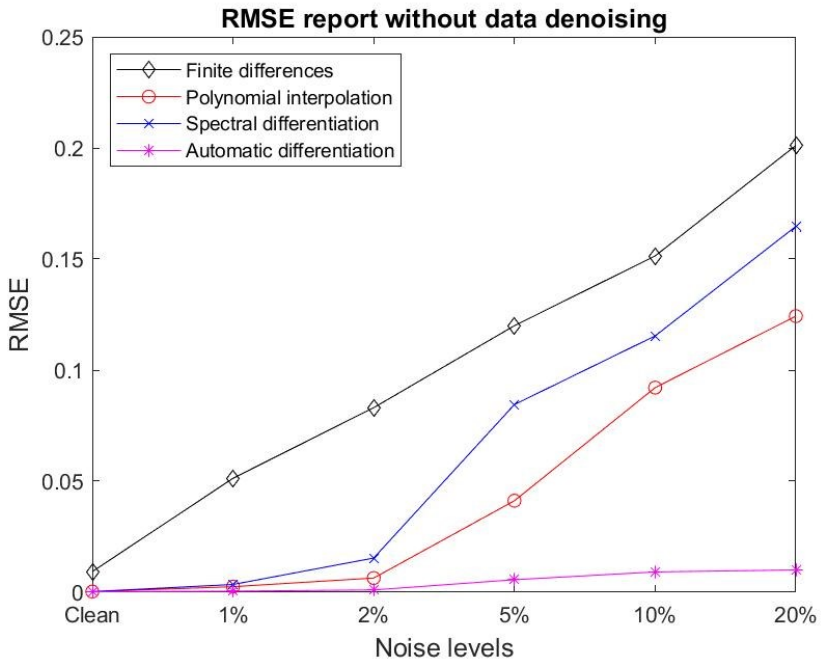

Figure 2.15: RMSE report for different gradient estimation methods without denoising(KS equation)

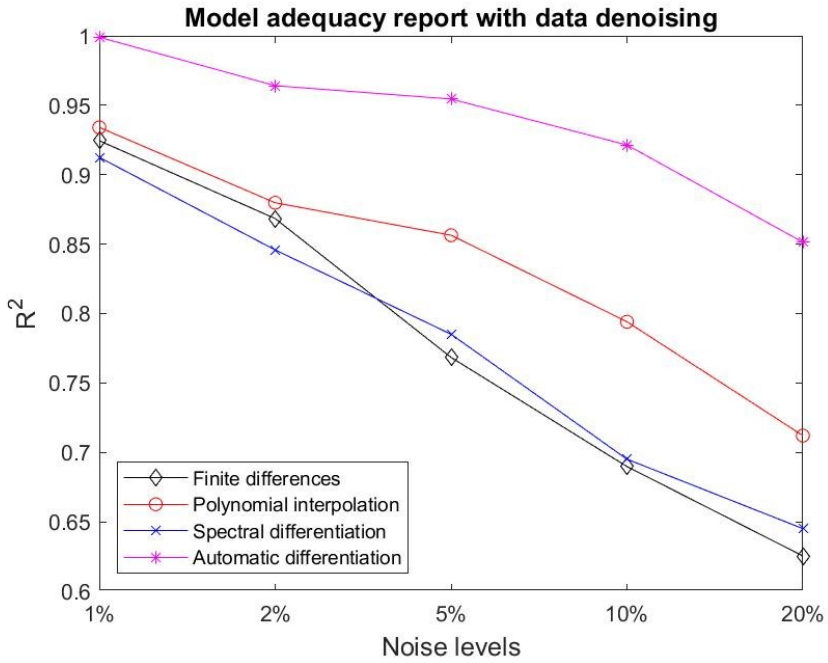

Figure 2.16: Model adequacy report(R^2) for different gradient estimation methods with denoising(KS equation)

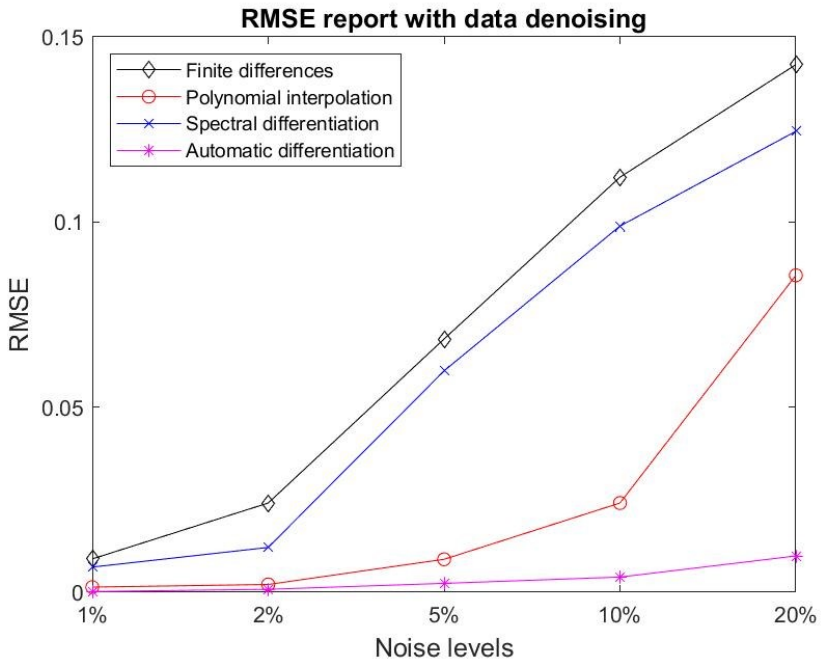

Figure 2.17: RMSE report for different gradient estimation methods with denoising(KS equation)

2.3.3.2 Comparison of obtained PDE models for different sparsity regularizations imposed.

In the case of the KS equation, only STRidge imposition led to PDE models with higher accuracy. Compared to STRidge, the elastic nets and LASSO failed to produce satisfactory results. Similar procedure as sections 2.3.1.2 and 2.3.2.2 is followed to obtain the average $R^2$and RMSE values.

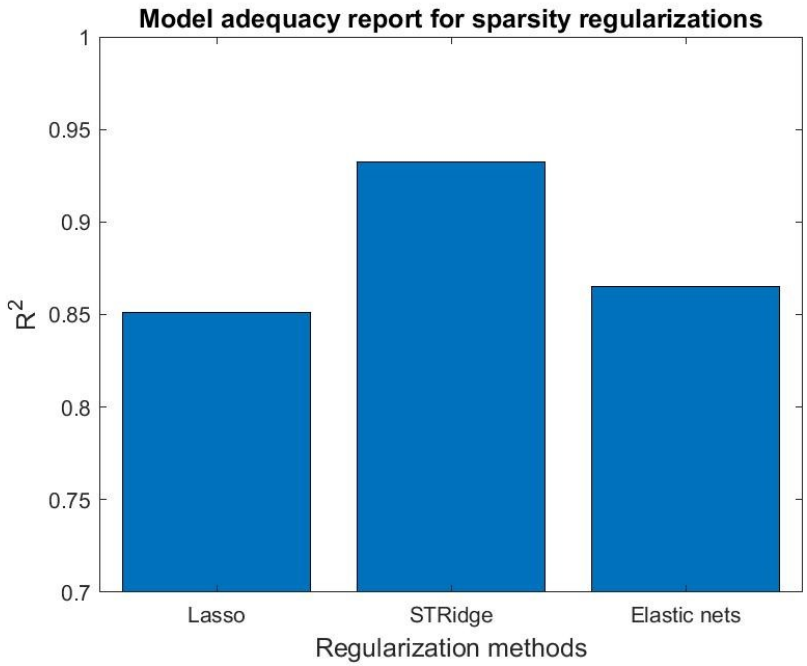

Figure 2.18: Model adequacy report(R^2) for different regularization methods(KS equation)

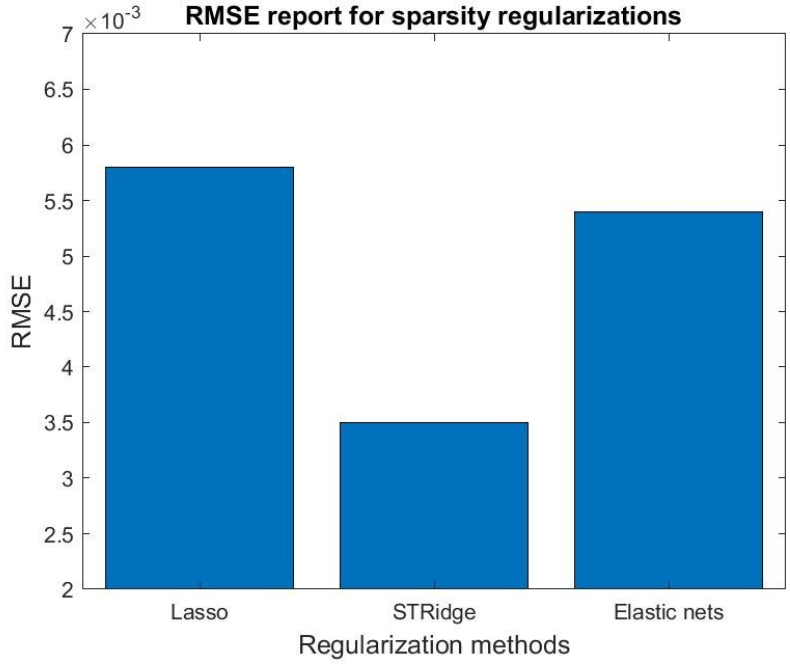

Figure 2.19: RMSE reports for different regularization methods(KS equation)

2.3.3 Chromatography

Finally, we tested the PDE discovery method along with investigating the optimal gradient estimation method, regularization imposition and impact of denoising on a chromatographic PDE. Chromatography is a technique used for the separation of a mixture.The separation is based on the differential partitioning between the mobile and stationary phases. Subtle differences in a compound's partition coefficient result in differential retention on the stationary phase and thus affect the separation.

We consider a solute transport equation for illustration. It is known as the equilibrium dispersive model and the adsorption equilibrium is described using a linear isotherm with the Henry's constant, H [41]. A single component solute propagation can be explained using this equation, 2.20.

The PDE for a particular single component separation is given by:

$$\left(1+\frac{1-\varepsilon}{\varepsilon}H\right) C_t + vC_x - DC_{xx} = 0 \tag{2.20}$$

where H is the Henry's constant, ϵ is the void fraction of the bed, and C is the solute concentration. For data generation, we consider $\epsilon = 0.4,\ D = 0.00087cm^2/s, H = 2.79$. Substituting these values and rearranging, we get

$$C_t = -0.0145C_x + 0.00012C_{xx} \tag{2.21}$$

Data is generated by simulating (2.21) with the initial condition $C(x, t = 0) = 0$, $x\epsilon[0,20], t\epsilon[0,2500]$ and boundary conditions $C(z = 0, t) = 4\ and\ C_x|_{z=L} = 0$. The number of spatial and temporal points after uniform discretization is 201 and 2501 respectively. We consider the highest order of the polynomial and spatial derivative in the candidate library to be 4.

For polynomial interpolation, 4th order Chebyshev polynomial is used and 10 points are used for fitting. To facilitate automatic differentiation, a nine-layer deep neural network with 20 neurons per hidden layer was utilized; tanh(x) was used as the activation function. We performed the same detailed studies on this data as in sections 2.3.1 to 2.3.3. Results

corresponding to discovered PDE for automatic differentiation after denoising, model adequacy reports for different gradient estimation methods, and analysis of different regularizations imposed are presented in table 2.13 and figures 2.20-2.25, respectively.

Table 2.13: PDE discovered for different noise levels(Chromatography)

Correct PDE : $C_t = -0.0145C_x + 0.00012C_{xx}$	
Noise level	Discovered PDE
Clean data	$C_t = -0.01481C_x + 0.00014C_{xx}$
1% noise	$C_t = -0.0151C_x + 0.00015C_{xx}$
2% noise	$C_t = -0.0168C_x + 0.00022C_{xx}$
5% noise	$C_t = -0.0114C_x + 0.000262C_{xx}$
10% noise	$C_t = -0.0175C_x + 0.000090C_{xx}$
20% noise	$C_t = -0.0098C_x + 0.000451C_{xx}$

As it is a linear second-order PDE, utilizing denoising prior to the discovery and automatic differentiation for gradient estimation, we were able to obtain a good approximation of the PDE even for higher levels of noise.

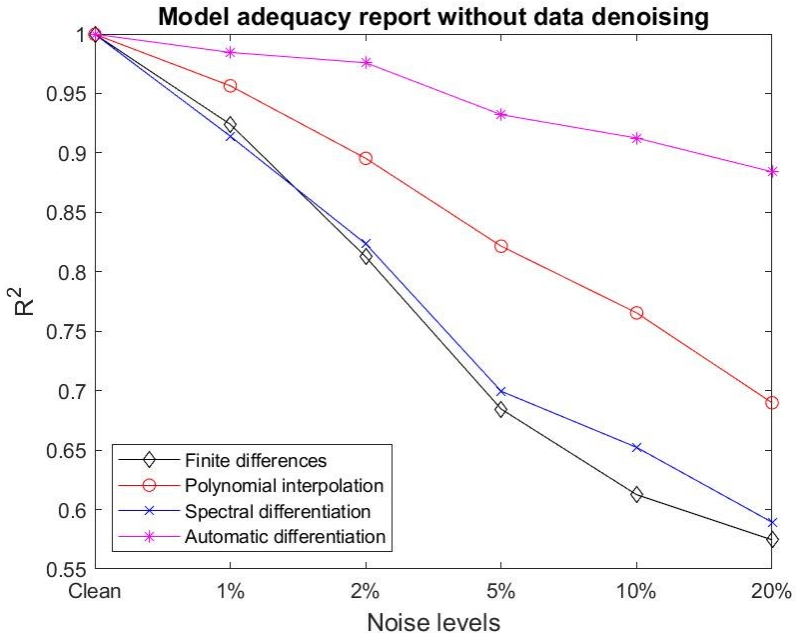

Figure 2.20: Model adequacy report(R^2) for different gradient estimation methods without denoising(Chromatography)

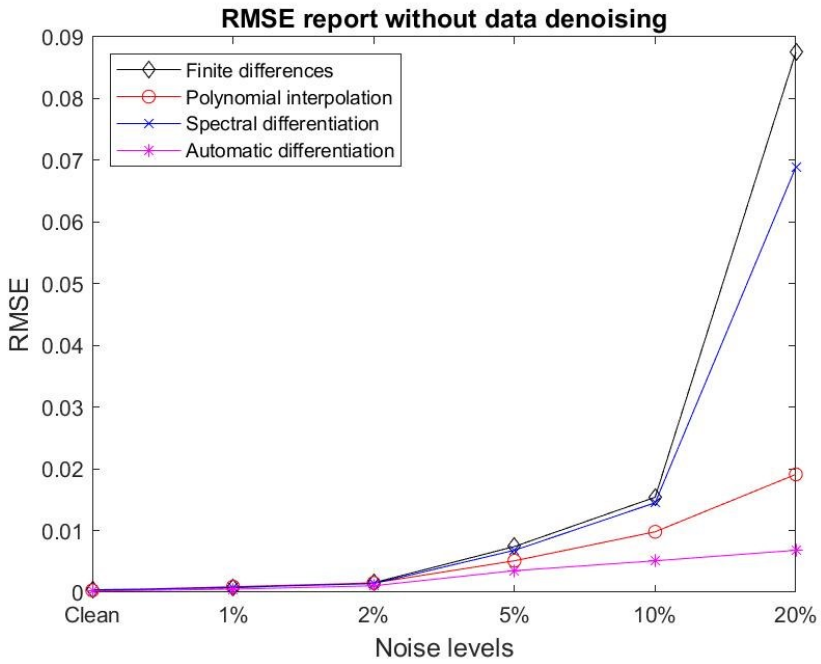

Figure 2.21: RMSE report for different gradient estimation methods without denoising(Chromatography)

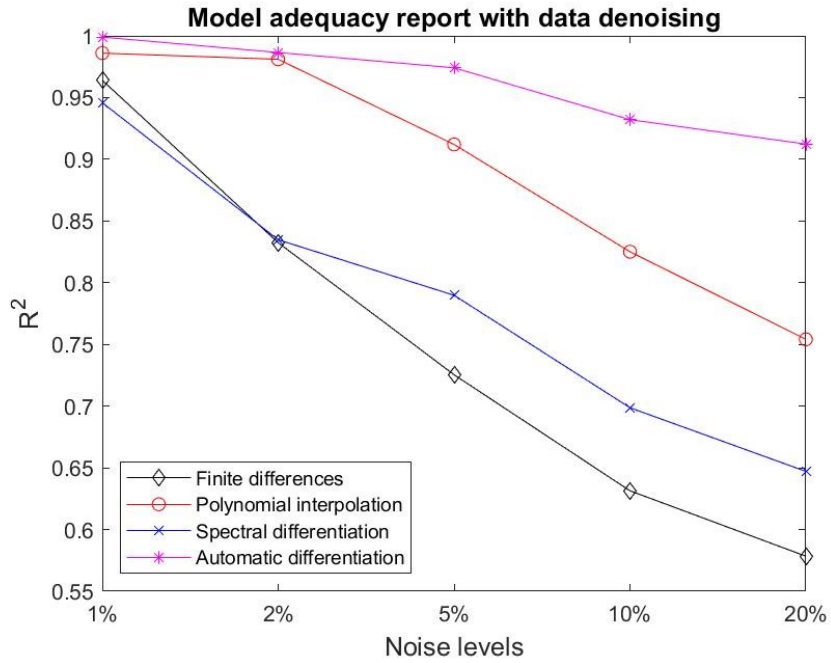

Figure 2.22: Model adequacy report(R^2) for different gradient estimation methods with denoising(Chromatography)

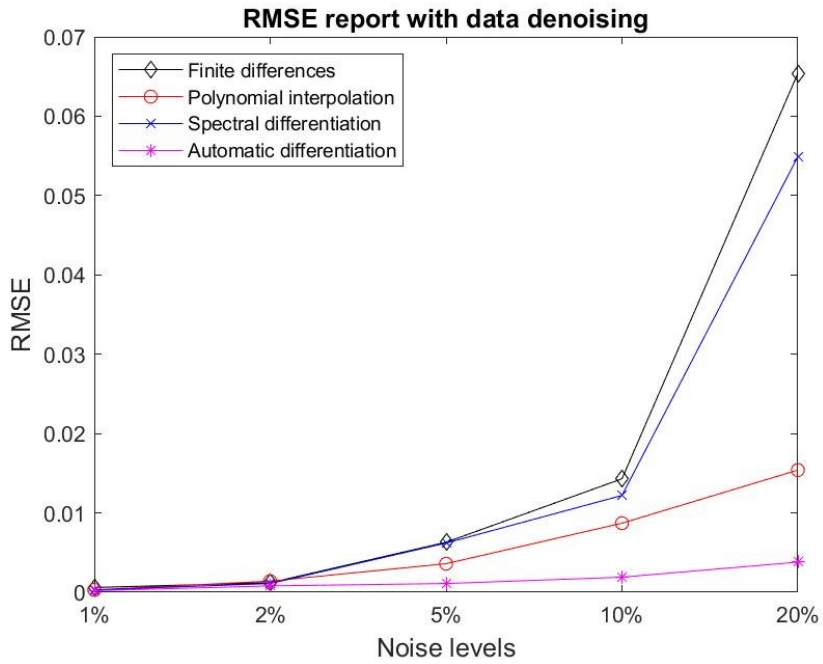

Figure 2.23: RMSE report for different gradient estimation methods with denoising(Chromatography)

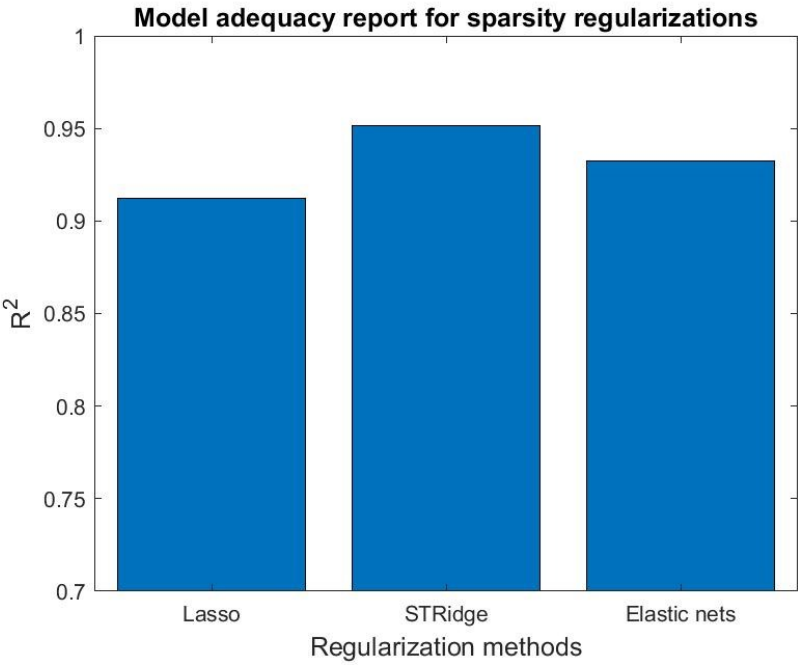

Figure 2.24: Model adequacy report(R^2) for different regularization methods(Chromatography)

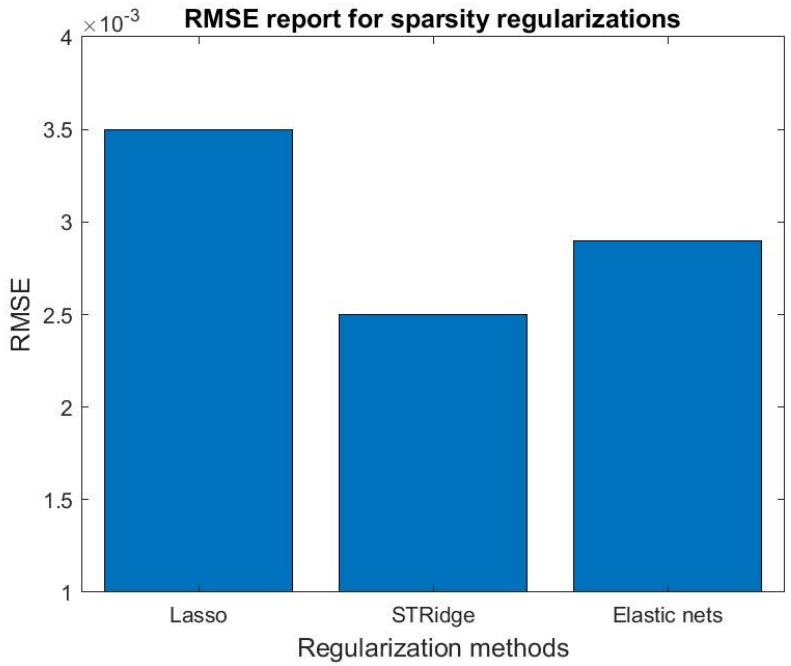

Figure 2.25: RMSE reports for different regularization methods(chromatography).

From figures 2.24 and 2.25, we can observe that all the regularizations provide comparable results, yet STRidge can be considered the best method based on the metrics and its consistency over all the different systems containing PDEs of different orders.

2.4 Conclusions

In this work, a detailed study of different uncertainties involved in the data-driven, sparse regression framework of PDE discovery is studied and the best possible approaches to mitigate the uncertainties are proposed. Gradient estimation and regression are the two major components of the PDE discovery algorithm. A numerical scheme such as finite differences, approximation strategies using polynomial interpolation, gradient estimation in the spectral domain using the fast Fourier transform, and a deep learning-based approach that utilizes the automatic differentiation are all implemented for various systems under study to infer the best gradient estimation method. To make inferences about practical scenarios, different levels of noise were added to the data and the obtained results were compared. Automatic differentiation was inferred to be the optimal gradient estimation

method compared to the other three methods mentioned. However, utilizing polynomial interpolation also yielded good models at lower noise levels. As we are interested in parsimonious solutions, the impact of the type of sparsity regularization used was also studied for different noise levels. LASSO, Sequential threshold ridge regression, and elastic nets were studied in this work. Although using elastic nets could provide good models for a few case studies, consistency across different systems was missing. Sequential threshold ridge regression, which utilizes the L2 norm, was able to generate higher accuracy models for all systems with different derivative orders. An SVD based denoising approach [37] was utilized to study the impact of denoising data prior to the PDE discovery. Denoising the data may be recommended for higher levels of noise due to the significant difference between the model accuracies obtained with and without denoising. However, for lower levels of noise, considering the trade-off between computational cost and difference in the obtained model accuracies, utilizing automatic differentiation or polynomial interpolation for gradient estimation will compensate for the denoising framework. The inferences obtained in this chapter will be utilized in chapter 3 for the development of hybrid models and in chapter 4 for the development of parametric PDE models for temperature dynamics in an oil reservoir. Solutions of PDEs of different orders, such as Burgers' equation(2^{nd} order), KdV equation(3^{rd} order), KS equation(4^{th} order) and a PDE from the chromatographic study were utilized to demonstrate the detailed study of all the different algorithms.

Chapter 3

A hybrid modelling approach to discover a system of partial differential equations

3.1 Introduction

Partial differential equations(PDEs) are utilized to explain numerous physical systems and conservation laws. A wide variety of problems in solid-state physics, fluid dynamics, electromagnetism, climate modelling, chemical engineering, etc. are formulated by PDEs. While some systems can be explained by a single PDE, some complex systems require multiple PDEs, also called a system of PDEs to formulate the dynamics. Burgers' equation, KdV equation, KS equation as shown in chapter 2 are some examples for individual PDEs. A popular example for a system of PDEs is the Navier-Stokes equations which describe the motion of viscous fluid substances. Mathematically, they express the conservation of momentum and conservation of mass for Newtonian fluids. They are accompanied by an equation of state which is related to pressure, temperature and density. Few other illustrations of a system of PDEs can be given by microscopic Maxwell's equations, resistive magnetohydrodynamics equations [42], Reaction-diffusion system [39], Keller Segel equations of chemotaxis [43], etc.

The general form of a linear system of PDEs can be illustrated as,

$$\begin{aligned} u_t &= L[u] + f_1(t)u + g_1(t)w \\ w_t &= L[w] + f_2(t)u + g2(t)w \end{aligned} \tag{3.1}$$

Where L can be any linear differential operator in the spatial coordinates $x_1, x_2, \ldots . x_n$ with any derivative order. f_1, g_1, f_2, g_2 can be any arbitrary functions or constant coefficients of u and w. u_t and w_t are the time derivates of u and w, respectively.

Grey box modelling in the computational modelling framework incorporates a prior theoretical structure of the model before estimation of the parameters. Grey box models are dissimilar to both black-box models where the model structure is assumed to be unknown and white-box models which are theoretical or first principle-based. Knowledge from the data and some prior information about the model structure is utilized to build the model. In the PDE discovery literature, notable work assuming the complete knowledge of the structure is presented in Physics informed networks(PINNs) by Raissi et. al.[15]. All other works consider a black box architecture, i.e., an unknown PDE structure. However, grey box modelling is extensively used in control of the chemical engineering processes. Identification of linearly parametrized dynamic regression model using physical knowledge and Bayesian techniques was first introduced by Tulleken [44]. Non-linear system identification by combining partial information about the process to a neural network to capture the dynamics of the process was proposed by Xiong et al [45]. The proposed method yielded in a reduction of the neural network size, the number of data points, and ensured faster convergence to the optimal value. As an extension of the control application stated above, Sohlberg designed a model predictive controller for a heating process based on the conservation of energy and measured data from the experiment [46]. These are some of the illustrations of the application of grey box modelling in model identification and control. A review of different branches of grey box modelling, such as constrained black box identification, semi-physical modelling, mechanistic modelling, hybrid modelling, distributed parameter modelling, and its application to various chemical engineering processes is presented by Sohlberg. et al[47].

The hybrid modelling framework is utilized mainly for the optimization and control of different physical and biological processes. Knowledge of the process from different levels is integrated to obtain interpretable models. The type of knowledge or prior information that is incorporated can be different for each process. Process knowledge may be inferred from the first principle model, or conservation laws governing the process. The prior

information can be utilized to determine the model structure, applied as a constraint to the optimization problem, or any other way aiding us to obtain a better model. The method of incorporating the physical knowledge and solving the optimization problem will be dependant on the model identification procedure followed and the properties of the process under study. Application of hybrid modelling was introduced to bioprocess optimization and control by Schubert et. al, by integrating mathematical models, neural nets, and incorporating fuzzy expert systems [48]. Recently, the study and application of hybrid modelling and optimization are gaining significant attention, mainly in chemical engineering and process systems engineering. Process optimization of distillation reflux condition by combining first principle models of distillation and statistical approaches like least squares regression applied on the measured data was presented by Taguchi et al.[49]. Model enhancement and development of a digital twin framework by combining mechanistic models and data-driven models for continuous pharmaceutical manufacturing was proposed by Bhalode et al.[50] demonstrating the advantages of hybrid modelling in process industries. A detailed review of hybrid semiparametric modelling and parameter identification techniques and their applications in the area of process monitoring, control, optimization, scale-up, and model reduction is done by Stosch et al.[51] displaying the importance and growing application of hybrid modelling in the process systems engineering discipline. Hybrid models are best suited to improving the accuracy of the data-driven model by incorporating any physical knowledge prior to the model identification. As it ensures improvement in model accuracy, a hybrid modelling framework is best suited for applied research and real-time applications.

Integrating physical knowledge with the data-driven approach while modelling the process dynamics generates a physical insight into the obtained model. Models with higher accuracy will lead to better predictions and optimal control strategies. In this work, we propose a hybrid modelling framework to discover PDEs by incorporating any available physical knowledge a priori to the PDE discovery framework. This aids in the reduction of uncertainties in the data-driven PDE discovery, such as incomplete candidate library and unknown order of the PDE. The system of PDEs generally describes a complex system for which the discovery of PDEs using data-driven approaches might lead to low accuracy models. Incorporating prior process knowledge and building a hybrid model for such

complex systems will yield better interpretable PDE models. The optimal gradient estimation method, sparsity regularization technique for the data-driven PDE discovery, has been inferred in chapter 2. We utilize these inferences and integrate them with process knowledge to build PDE models for various systems of PDEs. To conduct a detailed study and construct a robust approach, we demonstrate PDE discovery considering multiple different cases of incorporating the prior knowledge for a system of PDEs. The order of the PDE, coupled nonlinear terms of the system, and the uncoupled terms of the system is assumed to be known in each case; the co-efficient of the assumed part and structure of the remaining PDE is discovered and the obtained model is tested for accuracy. Another case of functional approximation for a complex section of the PDE is considered as prior information and discovery of such systems is also demonstrated. The improvement in model accuracy by utilizing the hybrid approach for noisy data is presented for various systems by comparing the data-driven models and hybrid models. Reaction-diffusion equation, Keller-Segel equations for chemotaxis and coupled PDE from chromatographic studies are considered as case studies to demonstrate the proposed approach. We validate the advantages of the hybrid modelling approach in the PDE discovery domain.

3.2 Hybrid modelling framework for discovery of a system of PDEs

The objective of the work is to discover a system of PDEs from data incorporating any known information about the process. The form of the system of PDEs assumed is described in equation (3.1). Modifications are done to the candidate library construction and sparse regression algorithm to facilitate the discovery of a system of PDEs and incorporate prior information. Inferences about gradient estimation and sparsity regularizations from chapter 2 are utilized to obtain a high accuracy model.

A set of a known system of PDEs with pre-defined initial and boundary conditions were solved using spectral differentiation combined with $ode45$ solver in MATLAB for validating the proposed method. The specific values corresponding to each case study is

explained in the results section. Various levels of Gaussian noise were added to the data to compare the data-driven and hybrid model accuracy at different levels of noise.

3.2.1 Candidate library

The system of PDEs may consist of $'n'$ number of individual PDEs, where $n > 1$. The number of PDEs required to describe a process will be dependant on the complexity of the process and the variables involved. To demonstrate our method, we consider a system of 2 PDEs; however, it can be extended to systems with more than 2 equations as well. The general form of a system of 2 PDEs for the PDE discovery can be expressed as,

$$\begin{aligned} u_t &= F(u, u_x, u_{xx}, \ldots\ldots uu_{xx}, u^2u_{xx} \ldots w, w_x, w_{xx}, ww_{xx}, \ldots uw, u_xw_x, u_xw_{xx} \ldots) \\ w_t &= F(w, w_x, w_{xx}, ww_{xx}, u, u_x, u_{xx}, \ldots\ldots uu_{xx}, u^2u_{xx} \ldots, \ldots uw, u_xw_x, u_xw_{xx} \ldots) \end{aligned} \quad (3.2)$$

u_t and w_t are time derivatives of the spatiotemporal data $u\ and\ w$ obtained from the solution of the PDE. The right-hand side of the equation constitutes a linear combination of spatial derivatives of different orders of $u\ and\ w$, polynomials of different orders of $u\ and\ w$, and their combinations. If multiple spatial dimensions are involved, derivatives with respect to all the spatial dimensions and corresponding combinations will also be a part of the candidate library.

The problem is formulated based on the assumption that the function F in (3.2) is parsimonious and consists of a few terms and is sufficient to explain the process dynamics.

$$\begin{aligned} U_t &= \Theta(U, W)\xi_1 \\ W_t &= \Theta(W, U)\xi_2 \end{aligned} \quad (3.3)$$

U_t and W_t are column vectors with time derivatives of the input data $u\ and\ w$ respectively. The candidate library Θ consists of all the possible terms of the PDE formed by the combination of spatial derivatives and polynomials of $u\ and\ w$ as described in (3.2). $\xi_1\ and\ \xi_2$ are the sparse vector of coefficients corresponding to $u\ and\ w$ respectively. Since Θ contains significantly more terms than the terms necessary to describe the PDE and

the terms representing the dynamics are within the span of Θ, most values in $\xi_1 and\ \xi_2$ will be zero leading to a sparse solution. Utilizing the inference from chapter 2, automatic differentiation is used to calculate the gradients involved in the candidate library.

From the structure of the candidate library shown in (3.2), and comparing it with the candidate library structure shown in (2.4), we can observe that number of terms involved in the candidate library in the case of a system of 2 PDEs is significantly higher, almost thrice the number of terms in the case of a single PDE. For a data-driven approach, the computational cost and the possibility of errors continue to increase as the system complexity increases. The obtained models become less reliable, especially in the presence of noise. Hence, to obtain reliable models with higher accuracy, the incorporation of prior process knowledge to the PDE discovery framework becomes essential.

3.2.2 Sparse regression

The objective of sparse regression is to obtain parsimonious models. From the candidate library pool, we need to identify the terms which are contributing to the PDE governing the system. From equation (3.3), we can perform least squares estimation and obtain coefficients for all the terms in Θ, but it is not practically reasonable to have a PDE with all the terms in the library. For example, if we had 50 terms in the library, our discovered PDE model would contain 50 terms and violates our objective of discovering parsimonious models. In this case, since Θ contains numerous candidate terms, the least-squares estimates are significantly affected by inversion numerical error. To promote parsimony and reduce the inversion error, sparse regression is utilized.

$$\begin{aligned} \widehat{\xi_1} &= argmin_{\hat{\xi}} \left|\left|\Theta\xi_1 - U_t\right|\right|_2^2 + \lambda \left|\left|\xi_1\right|\right|_0 \\ \widehat{\xi_2} &= argmin_{\hat{\xi}} \left|\left|\Theta\xi_2 - W_t\right|\right|_2^2 + \lambda \left|\left|\xi_2\right|\right|_0 \end{aligned} \tag{3.4}$$

Equation (3.4) assures that only limited terms whose effect on the error $\left|\left|\Theta\hat{\xi} - U_t\right|\right|$ outweigh their addition to $\left|\left|\xi\right|\right|_0$ appear in the PDE. l^0 is imposed in (3.4) which makes the problem *np*-hard.

There are multiple ways of convex relaxation of the l^0 norm, such as LASSO, Ridge, Elastic nets, etc. Detailed explanation and study of different sparsity regularization imposition on the PDE discovery have been presented in chapter 2, and STRidge was determined to be the optimal regularization method that yielded consistent results across various systems.

The method of sequential thresholding coupled with ridge regression is proposed by Rudy et al. in [5]. In sequential thresholding, a hard threshold is placed on the regression coefficients once the predictor is obtained, and this process is repeated recursively on the remaining nonzero coefficients. Ridge regression is an imposition of the l^2 norm to the least-squares problem.

$$\begin{aligned} \widehat{\xi_1} &= argmin_{\hat{\xi}} \left|\left|\Theta\xi_1 - W_t\right|\right|_2^2 + \lambda \left|\left|\xi_1\right|\right|_2^2 \\ \widehat{\xi_2} &= argmin_{\hat{\xi}} \left|\left|\Theta\xi_2 - W_t\right|\right|_2^2 + \lambda \left|\left|\xi_2\right|\right|_2^2 \end{aligned} \tag{3.5}$$

(3.5) represents the ridge regression. Combining this with the sequential thresholding mechanism, it is called Sequential threshold ridge regression or STRidge [5]. For $\lambda = 0$, (3.5) reduces to sequential threshold least squares problem.

As a system of PDEs is involved, sparse regression was performed recursively, changing the variable involved each time. For example, from equation (3.3) U_t was considered for the first iteration and corresponding ξ_1 was estimated. W_t was considered for the second iteration and corresponding ξ_2 was estimated. Θ remains constant for all the iterations. The number of recursions is equal to the number of equations in the PDE system or the number of measured variables under study. The system of PDEs governing the process is presented after all the recursions are completed. A brief description of the sparse regression algorithm for a system of 2 PDEs is presented below in table 3.1.

Table 3.1: Sparse regression for a system of PDEs

Algorithm: STRidge ($\boldsymbol{\Theta, U_t, W_t\ \lambda, iters}$)	
$\widehat{\xi_1} = argmin_{\xi}\left\|\|\Theta\xi_1 - U_t\|\right\|_2^2 + \lambda\left\|\|\xi_1\|\right\|_2^2$	#ridge regression
$bigcoeffs = \{j: \left\|\hat{\xi}_j\right\| \geq tol\}$	#selecting large coefficients
$\widehat{\xi_1}[\sim bigcoeffs] = 0$	#apply a hard threshold
$\widehat{\xi_1}[\sim bigcoeffs] = STRidge(\Theta[:, bifcoeffs], U_t, tol, iters - 1)$	#recursive call
#after ξ_1 is obtained, proceed to estimate ξ_2	
$\widehat{\xi_2} = argmin_{\xi}\left\|\|\Theta\widehat{\xi_2} - U_t\|\right\|_2^2 + \lambda\left\|\|\xi_2\|\right\|_2^2$	
$bigcoeffs = \{j: \left\|\hat{\xi}_j\right\| \geq tol\}$	
$\widehat{\xi_1}[\sim bigcoeffs] = 0$	
$\widehat{\xi_2}[\sim bigcoeffs] = STRidge(\Theta[:, bifcoeffs], W_t, tol, iters - 1)$	
return $\widehat{\xi_1}\ and\ \widehat{\xi_2}$	

The sparsity of the PDE obtained through STRidge is dependent on the threshold tolerance. Hence, a separate method is developed to estimate the best tolerance level. The algorithm for a system of 2 PDEs is explained below in table 3.2.

Table 3.2: Estimation of best tolerance for a system of PDEs

Algorithm : TrainSTRidge($\Theta, U_t, W_t\ \lambda, d_{tol}, num_{points}, tol_{iters}, STR_{iters}$)

#Split the data into training and testing sets

$\Theta \rightarrow [\Theta^{\text{train}}, \Theta^{test}]$

$U_t = [U_t^{train}, U_t^{test}]$

} 80/20 split

#Set an appropriate l^0 penalty. (Selected based on empirical evidence)

$\eta = 10^{-3}\kappa(\Theta)$

#Obtaining a baseline predictor

$\xi_{best} = \left(\Theta^{train}\right)^{-1} U_t^{train}$

$error_{best} = \left|\left|\Theta^{test}\ \xi_{best} - U_t^{test}\right|\right|_2^2 + \eta\left|\left|\xi_{best}\right|\right|_0$

#Searching through values of tolerance to find the best predictor

$tol = d_{tol}$

$for\ iter = 1, \ldots., tol_{iters}$:

$\xi = STRidge\left(\Theta^{train}, U_t^{train}, \lambda, tol, STR_{iters}\right)$ #Train and evaluate performance

$error = \left|\left|\Theta^{test}\xi - U_t^{test}\right|\right|_2^2 + \eta\left|\left|\xi\right|\right|_0$

#Is the error still dropping?

$if\ error \leq error_{best}$:

$error_{best} = error$

$\xi_{best} = \xi$

$tol = tol + d_{tol}$

#Or is the tolerance too high?

$else$:

$tol = \max([0, tol - 2d_{tol}])$

$d_{tol} = \dfrac{2d_{tol}}{tol_{iters} - iter}$

$tol = tol + d_{tol}$

$SS_E = \sum[\Theta^{test}\xi - U_t^{test}]^2$;

$S_{yy} = \sum[U_t^{test} - mean(U_t^{test})]^2$

$R^2 = 1 - \frac{SS_E}{S_{yy}}$ #Model adequacy

$rmse = \sqrt{\frac{SS_E}{num_{points}}}$ #Mean squared error

$return\ \xi_{best}, R^2, rmse$

$\#same\ procedure\ is\ repeated\ for\ next\ iteration\ by\ replacing\ U_t with\ W_t.\xi_1\ with$

3.2.3 Hybrid modelling approach

In earlier sections of this chapter, we have discussed the method to discover a system of PDEs through a data-driven approach. However, the motivation for the development of hybrid models has been discussed in section 3.1. From the candidate library construction and sparse regression algorithms, it is evident the computational complexity is significantly higher in the discovery of a system of PDEs. The hybrid modelling framework has been utilized to increase model accuracy in many other optimizations, identification and control domains[51]. Therefore, considering the uncertainties in PDE discovery and advantages of hybrid modelling, we incorporated the physical knowledge prior to the model discovery, to reduce the computational complexity yielding higher accuracy models.

The process information converted into mathematical form has to be incorporated in the model identification framework. In our case, candidate library construction and sparse regression are the two vital sections that decide the accuracy of the model. Any known process information has to be incorporated in one of these sections. If we consider the candidate library construction section, we have to reduce the number of terms in the candidate library either by elimination or by the selection of a few terms before the sparse regression. Another possible option is to convert the physical information into constraints in the sparse regression framework.

In this work, we concentrate on incorporating the process knowledge into the candidate library and reduce the number of terms before the sparse regression. It is not practically feasible to develop a common framework to incorporate physical knowledge of different systems due to the varying properties and behaviour of each system. However, we know that any physical knowledge when converted to mathematical form, will have a few similar

features. To develop a robust method applicable across processes, we directly consider the different mathematical forms of the physical knowledge that can be incorporated into the candidate library. Hybrid modelling specific to a process is explained in detail in chapter 4. We consider three different cases of inducing process knowledge into the candidate library.

a) Knowledge about the order of the PDE: Based on the conservation laws or governing laws of the process, sometimes we can assume the highest possible order of the derivative in the process. By considering this assumption, we can set the highest order of the spatial derivative in the candidate library and eliminate all higher-order derivatives and their combinations. For example, if we infer that the highest derivative order is 2, we need not calculate the 3^{rd} or higher-order derivatives for the PDE discovery. This significantly reduces the computational cost for gradient estimation and also reduces the error during sparse regression.

b) Knowledge of coupled terms in the PDE: Multiple variables involved in the system of PDEs are the reason for increased complexities. Most of the PDE systems contain coupled non-linear derivative terms which are challenging to identify using data-driven approaches due to high correlation in columns of Θ and numerical errors in gradient estimation. If we can obtain information about the possible coupled terms in the governing equation, we can modify the sparse regression framework to estimate the coefficients of these known coupled terms excluding them from the threshold values. Hence, we are forcing the regression framework to estimate the coefficients of known coupled terms integrated with sparse regression for the remaining terms. This approach increases the model accuracy as we are enforcing the selection of a known term. Also, as we know the coupled terms of the governing PDE, we can exclude all the other coupled terms that are generated due to the combination of derivatives and polynomials.

c) Knowledge of other terms in the PDE: We are aware of many complex systems of PDEs in chromatographic studies, adsorption studies where multiple component transfers are involved leading to complicated terms in the PDE. Such terms are

neither generated by a combination of derivatives and polynomials nor can be assumed without prior process knowledge. Identification of these terms may be important to describe the process dynamics using the obtained model. This aids in significantly improving the model accuracy.

All the abovesaid three cases are validated using various systems of PDEs; detailed results are given in the results section.

3.3 Results and discussions

The extension of sparse regression framework to a system of PDEs and the hybrid modelling approach explained in section 3.2 is validated using the solutions of various systems of PDEs from physics and engineering. As we are demonstrating the proposed hybrid approach to a complex system of PDEs, it can be applied to single PDE systems as well to improve the model accuracy subject to prior process knowledge. We utilize systems of 2 PDEs and multiple spatial dimensions to demonstrate the hybrid modelling approach. The reaction-diffusion equation, Keller-Segel equation for chemotaxis, and a system of PDEs from chromatographic studies are considered case studies.

3.3.1 Reaction-diffusion equation

The reaction-diffusion systems are extensively used to study pattern formations in mathematical physics [5]. Many patterns such as spiral waves, sawtooth behaviour can be observed and studied through reaction-diffusion systems. We consider a reaction-diffusion system that exhibits spiral waves on a 2D domain with periodic boundaries. It is commonly known as the $\lambda - \omega$ system[39].

$$\begin{aligned} u_t &= 0.1\nabla^2 u + \lambda u - \omega v \\ v_t &= 0.1\nabla^2 v + \omega u - \lambda v \\ \omega = -\beta(u^2 + v^2), &\lambda = 1 - (u^2 + v^2)\,, \beta = 1 \end{aligned} \tag{3.6}$$

After simplification,

$$u_t = 0.1u_{xx} + 0.1u_{yy} + u - uv^2 - u^3 + v^3 + u^2v$$
$$v_t = 0.1v_{xx} + 0.1v_{yy} + v - uv^2 - u^3 - v^3 - u^2v \quad (3.7)$$

We have two spatial dimensions $x, y \epsilon [-10,10]$, and timestamps from 0 to 10. The number of spatial and temporal points after uniform discretization is 256 and 201, respectively.

For this specific case, we compare the models obtained from the data-driven method with the hybrid model approach. For the data-driven method, similar to the case studies shown in section 2.3, we construct a library with the highest spatial derivative order and polynomial order as 4 and perform the sparse regression. For the hybrid modelling, we utilize the approach explained in 3.2.3(a). We assume that we know the highest order of the spatial derivative and the polynomial order in the PDE system. In this way, the physical knowledge about the system is utilized to constrain the candidate library and reduce the possible candidate terms significantly. Different levels of Gaussian noise are added to the generated data to compare the model accuracy of both methods at various noise levels. As the efficiency of the hybrid model is the main concern of study in this chapter, we do not utilize the denoising framework. However, from the inferences of data-driven discovery of PDEs in chapter 2, we know that denoising the data will yield higher accuracy models. Since the systems under study are complex and the denoising framework is not used, we add up to 5% Gaussian noise to the data. R^2 and $RMSE$(explained in table 3.2 and equations 2.14 - 2.17) are used as metrics to compare the models obtained by different methods.

The PDEs discovered from the data-driven approach for different noise levels are shown below in table 3.3.

Table 3.3: Reaction-diffusion PDEs discovered from the data-driven approach

Correct PDE :
$u_t = 0.1u_{xx} + 0.1u_{yy} + u - uv^2 - u^3 + v^3 + u^2v$
$v_t = 0.1v_{xx} + 0.1v_{yy} + v - uv^2 - u^3 - v^3 - u^2v$

Noise level	Discovered PDE
Clean data	$u_t = 0.1u_{xx} + 0.1u_{yy} + 1.001u - 1.021uv^2 - 0.951u^3 + 0.98v^3 + 1.01u^2v$ $v_t = 0.1v_{xx} + 0.1v_{yy} + 1.002v - 0.994uv^2 - 0.998u^3 - 1.00v^3 - 1.04u^2v$
1% noise	$u_t = 0.25u_{xx} + 0.14u_{yy} + 1.21u - 1.31uv^2 - 0.95u^3 + 1.21v^3 + 1.38u^2v + 0.35u$ $v_t = 0.132v_{xx} + 0.211v_{yy} + 0.758v - 0.954uv^2 - 1.325u^3 - 1.054v^3 - 1.025u^2v + 0.254v$
2% noise	$u_t = 0.232u_{xx} + 0.324u_{yy} + 1.35u - 1.75uv^2 - 0.754u^3 + 0.865v^3 + 1.65u^2v + 0.489v + 0.0057uv$ $v_t = 0.3411v_{xx} + 0.2415v_{yy} + 1.42v - 1.24uv^2 - 0.865u^3 - 0.841v^3 - 0.68u^2v + 0.354u$
5% noise	$u_t = 0.568u_{xx} + 0.354u_{yy} + 1.39u - 0.415uv^2 - 1.687u^3 + 1.524v^3 + 0.714u^2v + 0.0681u^2 + 0.2541vu_x$ $v_t = 0.415v_{xx} + 0.654v_{yy} + 1.68v - 0.678uv^2 - 1.561u^3 - 1.392v^3 - 0.841u^2v + 0.0687uv_x + 0.498u$

The system of PDEs discovered through the hybrid modelling approach for different noise levels is shown in table 3.4

Table 3.4: Reaction-diffusion PDE system discovered using the hybrid approach

Correct PDE : $u_t = 0.1u_{xx} + 0.1u_{yy} + u - uv^2 - u^3 + v^3 + u^2v$ $v_t = 0.1v_{xx} + 0.1v_{yy} + v - uv^2 - u^3 - v^3 - u^2v$	
Noise level	Discovered PDE
Clean data	$u_t = 0.1u_{xx} + 0.1u_{yy} + 1.00u - 1.00uv^2 - 1.01u^3 + 0.99v^3 + 1.01u^2v$ $v_t = 0.1v_{xx} + 0.1v_{yy} + 1.00v - 0.99uv^2 - 1.02u^3 - 1.00v^3 - 1.01u^2v$

1% noise	$u_t = 0.12u_{xx} + 0.14u_{yy} + 1.08u - 1.11uv^2 - 0.96u^3 + 1.151v^3 + 1.230u^2v$ $v_t = 0.11v_{xx} + 0.101v_{yy} + 0.899v - 0.988uv^2 - 1.115u^3 - 1.004v^3 - 1.005u^2v$
2% noise	$u_t = 0.132u_{xx} + 0.189u_{yy} + 1.12u - 1.35uv^2 - 0.899u^3 + 0.862v^3 + 1.35u^2v$ $v_t = 0.3411v_{xx} + 0.2415v_{yy} + 1.42v - 1.24uv^2 - 0.865u^3 - 0.841v^3 - 0.88u^2v$
5% noise	$u_t = 0.143u_{xx} + 0.147u_{yy} + 1.24u - 0.729uv^2 - 0.758u^3 + 1.224v^3 + 0.814u^2v + 0.065v$ $v_t = 0.115v_{xx} + 0.198v_{yy} + 1.27v - 1.198uv^2 - 1.211u^3 - 0.854v^3 - 1.154u^2v + 0.049u$

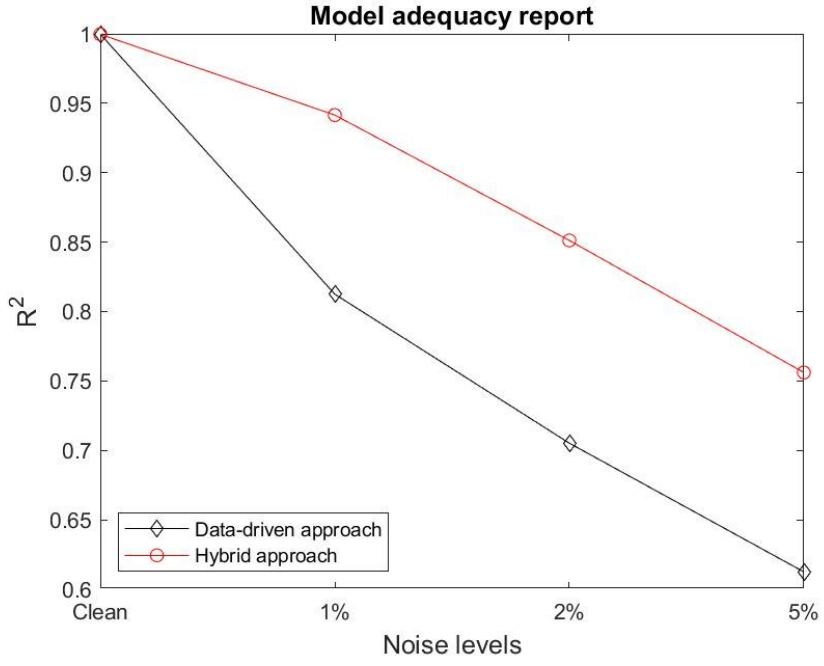

Figure 3.1: Model adequacy report(R^2) for Reaction-diffusion equation

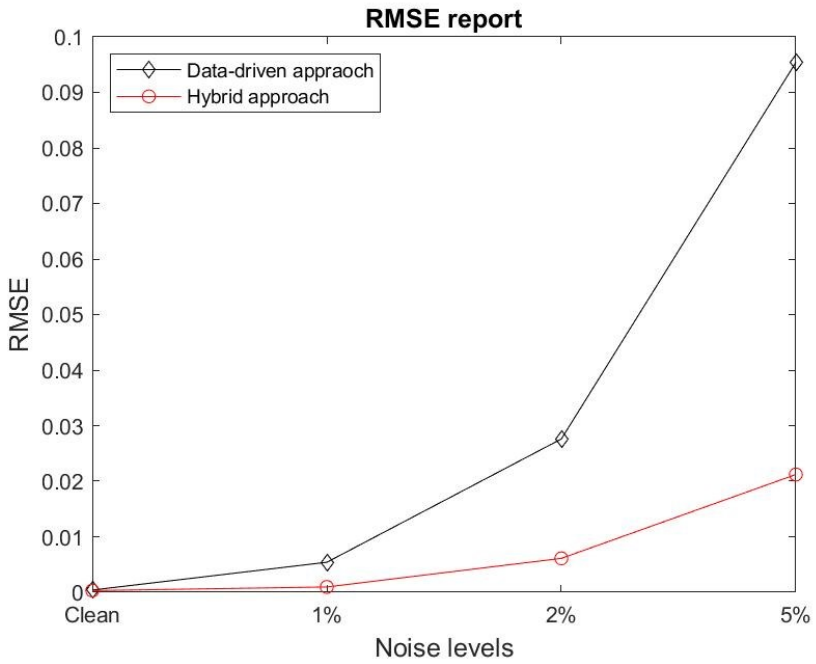

Figure 3.2: RMSE report for Reaction-diffusion equation

From figures 3.1 and 3.2, we can infer that the hybrid modelling approach provides better models compared to the data-driven models. Hybrid models are much suited for practical applications as they perform significantly better than black-box models in the presence of noise. In this case, we utilized the physical information to constrain the candidate library by limiting the maximum order of the PDE leading to a significant reduction in the computational errors, therefore obtaining a better model.

3.3.2 Keller-Segel equation for chemotaxis

The Keller-Segel equations describe the evolution of the density of cells and chemical secretion in the chemotactic attraction of biological systems.[43] It consists of 2 PDEs, explaining the drift diffusion in the first PDE and reaction diffusion of the chemical in the second PDE. [12]

$$u_t = \nabla.(D_u \nabla u - Xu\nabla w)$$
$$w_t = D_w \Delta \mathrm{w} - \mathrm{kw} + \mathrm{hu} \qquad (3.8)$$

Here u represents the density of cells and w represents the secreted chemical. D_u is the diffusion coefficient of the cells, X is the chemotactic sensitivity, $k\ and\ h$ represent the rate of production and degradation of the chemical by the cells respectively. For data generation we consider $D_u = 0.5, D_w = 0.5, X = 10.0, k = 0.05, h = 0.1$. Unlike the reaction-diffusion equation explained in section 3.3.1, the Keller-Segel equation has only one spatial dimension but it consists of coupled derivative terms because the chemotactic sensitivity X is a measure for the strength of their sensitivity to the gradient of the secreted chemical, w.

After substituting the above values and simplifying the terms, the Keller Segel equations are,

$$u_t = 0.50u_{xx} - 10uw_{xx} - 10u_x w_x$$
$$w_t = 0.50w_{xx} - 0.05w + 0.1u \quad (3.9)$$

We compare the models obtained from the data-driven method with the hybrid model approach. We implement the following approaches and compare the obtained models,

a) Data-driven approach – No prior assumptions are made. The candidate library is constructed with the highest order of spatial derivative and polynomials as 4 followed by sparse regression.
b) Hybrid model_1 – Similar to 3.3.1, we assume that we know the highest derivative and polynomial order and constrain the candidate library to that corresponding value. Here, we set the highest order of the derivative to 2 and the highest polynomial order to 1. This reduces the number of possible candidate terms aids in the reduction of computational error.
c) Hybrid model_2 - As explained in 3.2.3(b), the presence of coupled derivative terms makes the PDE discovery challenging for noisy data. In this case, the coupled term also becomes vanishingly small over most of the domain. As we are aware of the process from which the data is derived, utilizing the property of chemotactic sensitivity and its relation with cell density and secreted chemicals, we assumed that the coupled term, $u_x w_x$ is known and that there are no other coupled terms. We estimate the coefficient of $u_x w_x$ without considering the sparsity threshold value and perform sparse regression for other candidate library terms. We enforce the

regression to choose $u_x w_x$ irrespectively of this coefficient value. This also ensures that if the coupled term is vanishing over the domain but is a contributing term to the PDE, it is selected even though the values may be far below the sparsity threshold.

Gaussian noise with different magnitudes is added to the generated data to compare the model accuracy of the three methods at various noise levels.

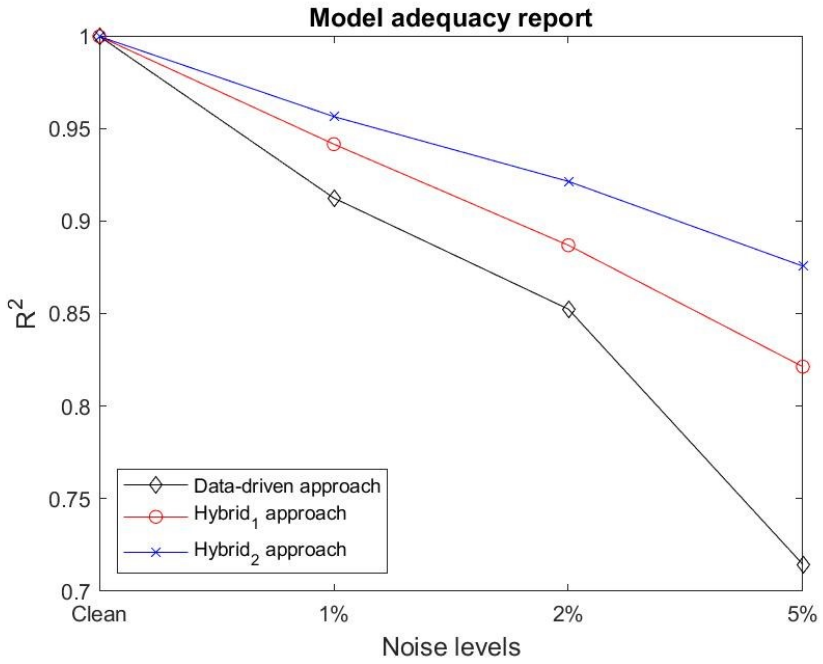

Figure 3.3: Model adequacy report(R^2) for Keller Segel equation

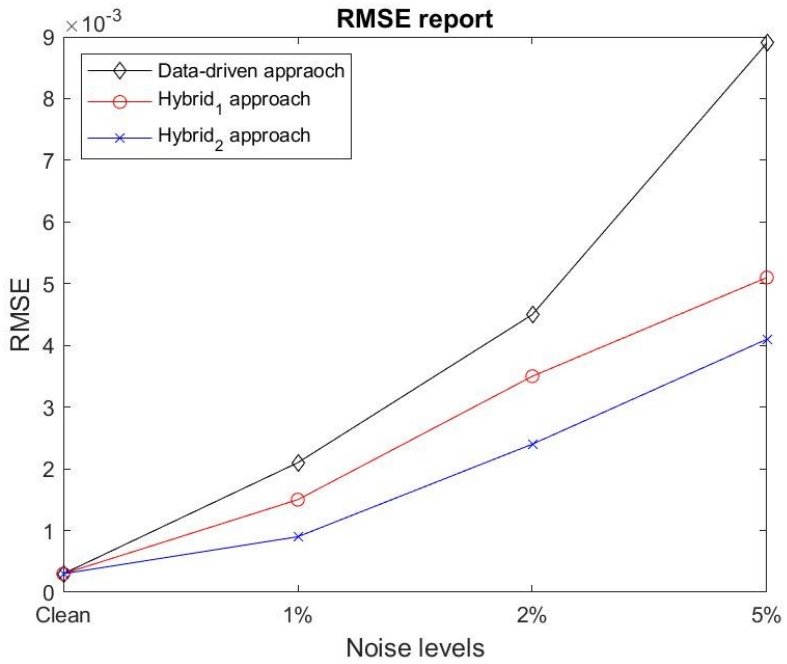

Figure 3.4: RMSE report for Keller Segel equation

We can conclude that incorporating the knowledge about the order of the PDE generated better models than the data-driven approach for higher noise levels. However, incorporating the knowledge about the coupled term of the PDE increases the model accuracy compared with the first hybrid approach. Hence, we conclude that incorporating detailed process-specific knowledge into the PDE discovery algorithm leads to higher accuracy models.

3.3.3 Chromatographic studies

Chromatography is an efficient tool for separations in pharmaceutical, food, and agrochemical industries. The solute movement in fixed bed and moving bed adsorption columns is studied using the equilibrium theory of chromatography. The solute movement is described only by considering convection and equilibrium between solid and fluid phases. The mass transfer and dispersion effects are neglected. The equilibrium theory provides accurate descriptions of the propagation of shocks, waves, and semishocks. Equilibrium

theory provides explicit solutions when applied to simple isotherms, such as the Langmuir isotherm for single and multicomponent systems [41].

We consider a system of PDEs corresponding to the equilibrium theory for the binary chromatography of a species applied to the general Langmuir isotherm.

$$\frac{\partial C_1}{\partial t} + v\frac{\partial C_1}{\partial x} - D\frac{\partial^2 C_1}{\partial x^2} + \frac{1-\varepsilon}{\varepsilon}\frac{\partial q_1}{\partial t} = 0 \tag{3.10a}$$

$$\frac{\partial C_2}{\partial t} + v\frac{\partial C_2}{\partial x} - D\frac{\partial^2 C_2}{\partial x^2} + \frac{1-\varepsilon}{\varepsilon}\frac{\partial q_2}{\partial t} = 0 \tag{3.10b}$$

$$\frac{\partial q_2}{\partial t} = k_2(q^*{}_2 - q_2) \tag{3.11a}$$

$$\frac{\partial q_1}{\partial t} = k_1(q^*{}_1 - q_1) \tag{3.11b}$$

$$q^*{}_1 = \frac{H_{11}C_1}{1 + K_{11}C_1 + K_{12}C_2} + \frac{H_{21}C_1}{1 + K_{21}C_1 + K_{22}C_2} \tag{3.12a}$$

$$q^*{}_2 = \frac{H_{12}C_2}{1 + K_{11}C_1 + K_{12}C_2} + \frac{H_{22}C_1}{1 + K_{21}C_1 + K_{22}C_2} \tag{3.12b}$$

C_1, C_2 are fluid and solid phase concentrations, ν is the interstitial velocity, ϵ is the void fraction of the bed, H_i is Henry's constant, and K_i is the adsorption equilibrium constant of component i.

Initial conditions: $C(x, t = 0) = 0$

Boundary conditions: $C_1(x = 0, t) = 4; \quad C_2(x = 0, t) = 4$

$$\frac{\partial C}{\partial x}\Big|_{x=L} = 0$$

Constant values used for data generation are,

$z\epsilon[0, L] \quad L = 20cm$, length of the column,

$t\epsilon[0, tspan]$ $tspan = 1500$ seconds ,

$v = 0.075\ cm/s$ - interstitial velocity ,

D = 0.00087,

$\epsilon = 0.697$,

$k_1 = 0.1s^{-1}, k_2 = 0.1s^{-1}$,

$K_{11} = 0.12\frac{mL}{mg}$, $K_{21} = 0.015\frac{mL}{mg}$,

$K_{12} = 0.33\frac{mL}{mg}$, $K_{22} = 0.015\frac{mL}{mg}$,

$H_{11} = 1.32\ \frac{mL}{mg}$, $H_{12} = 3.63\ \frac{mL}{mg}$,

$H_{21} = 1.47\ \frac{mL}{mg}$, $H_{22} = 1.47\ \frac{mL}{mg}$.

In this case, obtaining a black box model without incorporating any prior information is not feasible due to the presence of $\frac{\partial q_i}{\partial t}$, $i = 1,2$. We have to incorporate the time derivative of q_i as the prior information, or using the equilibrium theory of chromatography, we have to simplify equation (3.11) using (3.12) and substitute the resulting terms for the time derivative of q in the system of PDEs (3.10). We considered both cases and compared the obtained models.

Hybrid_1 approach: As we have the data of $C_1, C_2, t, x, q_1, and\ q_2$, we assume the highest order of the PDE as 2 and incorporate the time derivative of $q_1 and\ q_2$ in the candidate library along with spatial derivatives and polynomials of C_1, C_2.

Hybrid_2 approach: Instead of calculating the temporal derivatives of $q_1 and\ q_2$, we substitute the simplified equation derived from the model equation and generalized Langmuir isotherm.

A model adequacy comparison for models obtained using the Hybrid_1 and Hybrid_2 approaches among different levels of noise in the data is shown in figures 3.5 and 3.6.

Substituting (3.11) and other constant values to (3.10),

$$\frac{\partial C_1}{\partial t} + 0.075\frac{\partial C_1}{\partial x} - 0.0087\frac{\partial^2 C_1}{\partial x^2} + 0.434\left[0.1\left(q^*{}_1 - q_1\right)\right] = 0 \tag{3.13a}$$

$$\frac{\partial C_2}{\partial t} + 0.075\frac{\partial C_2}{\partial x} - 0.0087\frac{\partial^2 C_2}{\partial x^2} + 0.434\left[0.1\left(q^*{}_2 - q_2\right)\right] = 0 \tag{3.13b}$$

After simplification of $q_1^* \; and \; q_2^*$, long division is performed to obtain a linear combination of terms. The first four terms of the resultant long division were considered as an approximation. $q_1^* \; and \; q_2^*$ are functions of $C_1 \; and \; C_2$.

$$q_1^* = 1.45C_1 + 0.632C_2 + \frac{0.124C_2}{C_1} + \frac{4.067C_2^2}{C1} \tag{3.14a}$$

$$q_2^* = 3.10C_1 + 1.258C_2 + \frac{0.574C_2}{C1} + \frac{2.145C_2^2}{C1} \tag{3.14b}$$

Substituting the value of $q_1^* \; and \; q_2^*$ in 3.13a and 3.13b,

$$\frac{\partial C_1}{\partial t} = -0.075\frac{\partial C_1}{\partial x} + 0.00087\frac{\partial^2 C_1}{\partial x^2} - 0.2058(1.45C_1 + 0.632C_2 + \frac{0.124C_2}{C_1} + \frac{4.067C_2^2}{C_1} - q_1) \tag{3.15a}$$

$$\frac{\partial C_2}{\partial t} = -0.075\frac{\partial C_2}{\partial x} + 0.00087\frac{\partial^2 C_2}{\partial x^2} - 0.2058(3.10C_1 + 1.258C_2 + \frac{0.574C_2}{C_1} + \frac{2.145C_2^2}{C_1} - q_2) \tag{3.15b}$$

Based on equations 3.15a and 3.15b, we enforce the sparse regression to estimate coefficients of $C_1, C_2, \frac{C_2}{C_1}, \frac{C_2^2}{C_1}$ without considering the sparsity threshold. Gaussian noise with different magnitudes is added to the generated data to compare the model accuracy of both methods at various noise levels. The system of PDEs discovered for clean data is shown in 3.16a and 3.16b followed by model adequacy report and RMSE reports.

The system of PDEs discovered for clean data is,

$$\frac{\partial C_1}{\partial t} = -0.0699\frac{\partial C_1}{\partial z} + 0.0008\frac{\partial^2 C_1}{\partial z^2} - 0.2881C_1 + 0.1294C_2 + \frac{0.0224C_2}{C_1} + \frac{0.8379C_2^2}{C1} - 0.2049q_1 \tag{3.16a}$$

$$\frac{\partial C_2}{\partial t} = -0.0687\frac{\partial C_2}{\partial z} + 0.0008\frac{\partial^2 C_2}{\partial z^2} - 0.1745C_1 + 0.2519C_2 + \frac{0.1098C_2}{C2} + \frac{0.4407C_2^2}{C1} - 0.2051q_2 \tag{3.16b}$$

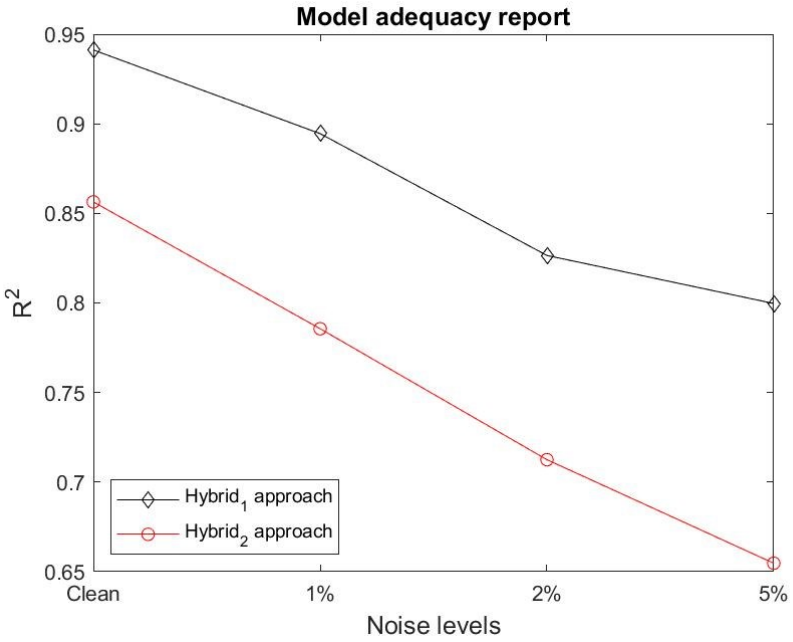

Figure 3.5: Model adequacy report(R^2) for the chromatographic PDE system

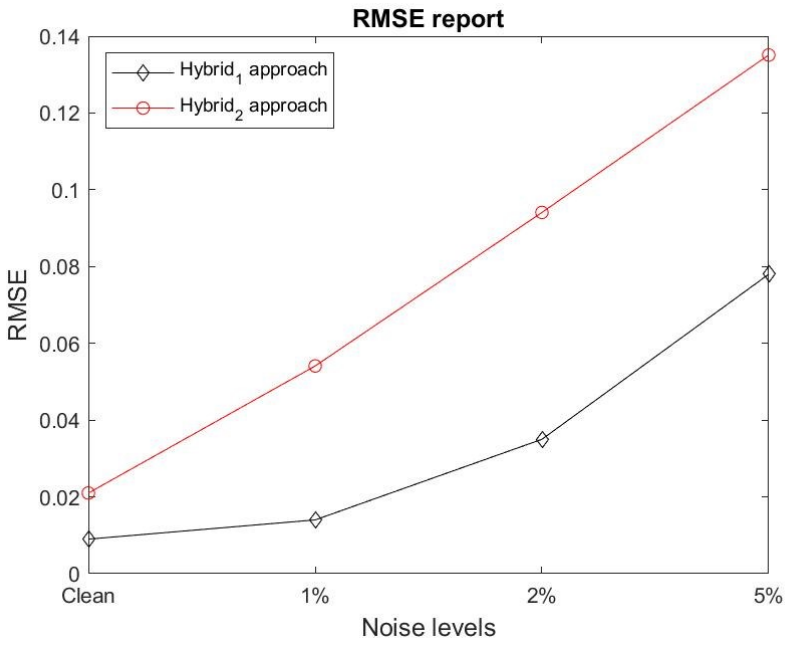

Figure 3.6: RMSE report for the chromatographic PDE system

Although a completely data-driven approach was not feasible for the chromatographic system of PDEs, we can observe that the first approach generates higher accuracy models than the second approach. This is because we are directly using the available q_1, q_2 data, calculating their time derivatives and incorporating those values into the candidate library. However, in the second approach, we perform functional approximations of the model equations which may be the reason for relatively lower accuracy values. If we are able to produce better approximations of equations 3.13a and 3.13b and incorporate those terms into the candidate library, we will be able to generate better models using the model simplification approach as well.

The common conclusion from all the three case studies discussed in section 3.3 is that incorporating any known process information prior to the PDE discovery will enhance the model accuracy, reduce the computational cost, and yield interpretable results. The hybrid modelling approach is able to generate models even for a complex system such as the chromatographic system of PDEs for which a black box model approach is highly infeasible. The hybrid modelling approach can be extended to the discovery of single PDE and discovery of PDE systems with more than 2 equations as well.

3.4 Conclusions

In this work, the sparse regression framework coupled with the hybrid modelling approach is introduced to discover a system of PDEs. A recursive framework is introduced to obtain a system of PDEs by extending the candidate library and recursively performing the sparse regression. To reduce the computational costs and to provide physical insights into the PDE discovery, a hybrid modelling approach was proposed where physical information about process was incorporated in the form of mathematical constraints prior to performing the PDE discovery. We make assumptions directly in mathematical format and incorporate them into the discovery framework to formulate a robust approach and study the advantages of hybrid modelling. We consider different cases of prior knowledge and incorporate them to minimize the computational errors due to the coupled derivative terms, and simplification of the known set of complicated terms of the PDE. We utilize the Keller-Segel equation of chemotaxis, the reaction-diffusion equation, and a system of PDEs obtained from chromatographic studies based on the equilibrium theory applied to the general Langmuir isotherm to demonstrate the various cases of hybrid modelling applied to the discovery of a system of PDEs.

Chapter 4

Modelling temperature dynamics of SAGD process in an oil reservoir

4.1 Introduction

Process models are developed based on first principles modelling, proxy modelling approaches, or data-driven methods. First-principle based models rely totally upon the physics of the process and can be infeasible to obtain for complex processes due to computational complexity or insufficient physical knowledge. Proxy or surrogate models are utilized for such processes which is an integration of different statistical techniques to achieve interpretable and reliable models. The data-driven models are based completely on the data retrieved from the past. Although data-driven models are proven to be reliable within a particular range of operating conditions, reliability outside the operating range or long-term prediction accuracy can be lower compared to physics-based models. Each process is governed by a few states and parameters. The evolution of the states over time and space provides information about the progress of the process itself. Studying the dynamics of these states by the development of governing equations for them helps in the prediction of future states, leading to better control and optimization. In this work, we focus on developing partial differential equation(PDE) models to explain the temperature dynamics in a petroleum reservoir. Petroleum reservoirs have a huge geographical spread and involve multiphase fluid flow through porous media. They are a large scale distributed parameter process with spatiotemporal variations involving multiple inputs and outputs [20].

The two main categories of oil and bitumen extraction are surface mining and in-situ recovery. Surface mining is utilized for shallow reservoirs, in which bitumen can be extracted using mechanical equipment. However, only 20% of the Canadian oil sands are shallow enough for surface mining. The remaining 80% have to be extracted using in-situ methods. Steam assisted gravity drainage(SAGD), Cyclic steam stimulation(CSS), and In-situ combustion are the thermal recovery technologies used for in-situ recovery. SAGD was invented by Dr Roger Butler in the 1970s. SAGD is an enhanced oil recovery technology for producing heavy crude oil and bitumen [52]. A pair of horizontal wells are drilled into the oil reservoir. High-pressure steam is continuously injected into the injector well which is placed above the producer well which reduces the viscosity of the surrounding bitumen in the oil sands. The less viscous oil flows down to the producer well due to gravity and is pumped out for further processing. The oil reservoir consists of a caprock layer, an impermeable layer, and the oil sands layer. Caprock is the topmost layer containing the oil and gas moving out from the oil sands layer. The oil sands layer holds the bitumen and the impermeable layer is at the bottom. The oil reservoirs generally consist of multiple injector and producer wells which are responsible for high pressure and temperature in the reservoir. Figure 4.1 shows the schematic of a SAGD process with a two-well injector and producer running in parallel [53]. We are interested in modelling the temperature dynamics for the reservoir by studying the steam chamber evolution across the reservoir at different instants of time. We obtain the data using CMG-STARS, a commercial reservoir flow simulator based on first principles [20][54].

In chapter 2 and chapter 3, we explored the areas of PDE discovery using data-driven and hybrid approaches. We tried to provide the best possible estimate of the PDEs in the presence of noise using denoising and various gradient estimation framework. The issues of interpretability, high computational cost, and discovery of a system of PDEs for complex systems were addressed by incorporating prior physical knowledge into the PDE discovery framework. However, in both cases, the PDEs were assumed to have constant parameters, i.e., the terms on the right-hand side of Equations 2.1 and 3.2 do not have an explicit dependence on time or space. Extracting the temporal or spatial variations of the coefficients in the PDE gives us the freedom to study many more systems and to discover

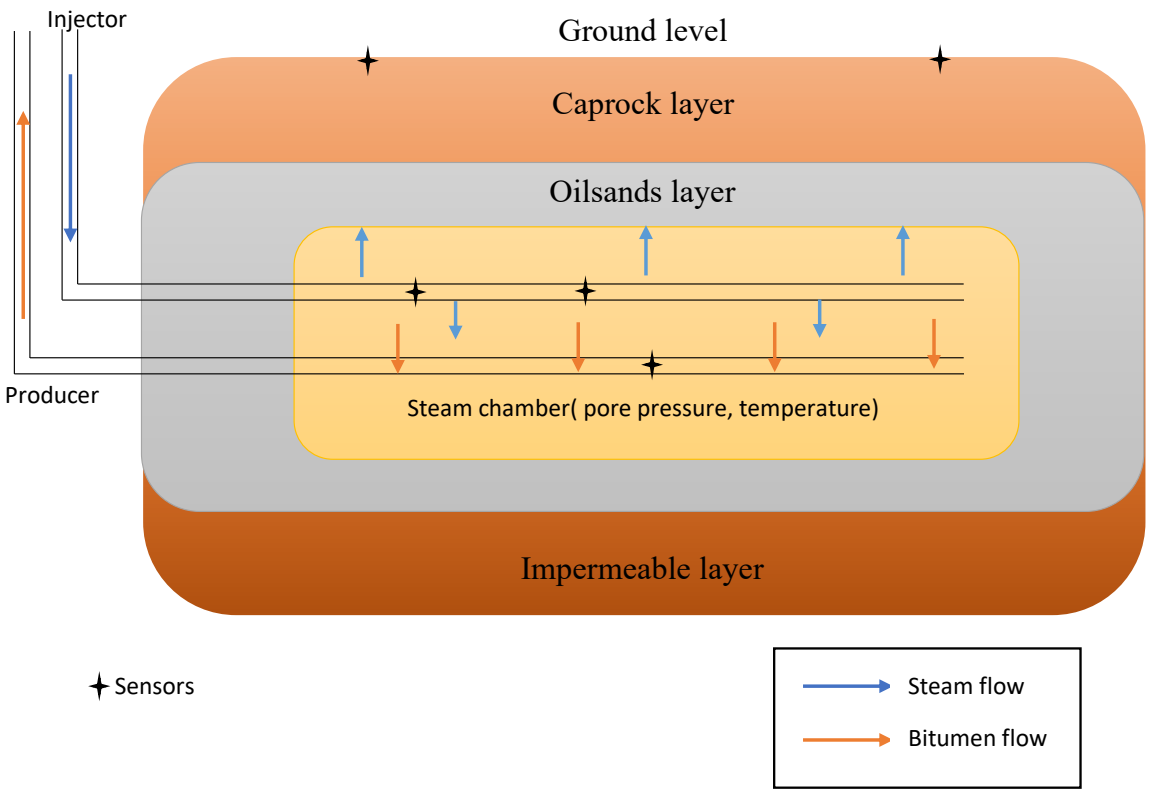

Figure 4.1: Schematic representation of SAGD process(front view)

governing equations using data-driven or hybrid modelling approaches. Rudy et al. proposed the sequential group threshold regression (SGTR) to discover the parametric PDEs using data-driven approaches [18]. A separate regression is constructed for each time step to obtain the corresponding time-varying coefficients. Similarly, it can be extended to spatial variations also. However, a general form of the varying coefficients cannot be estimated in this method. SGTR also has the limitation of an incomplete candidate library. To overcome some of these limitations, Xu et al. proposed the step-wide deep-learning genetic algorithm PDE (DLGA_PDE) approach which is an integration of deep-learning and genetic-algorithm approaches[55]. A dual neural network is utilized; the first neural network obtains the structure of the PDE and discovers the type of variation in the process. The second neural network calculates the values of varying coefficients for the terms identified by the first neural network. However, both the methods are data-driven and the physical interpretability of the model development framework and the obtained model in the above-said methods is not clear.

The parametric variation in each process can be characterised by the process parameters. Geostatistical simulation packages use a set of algorithms to generate spatial properties[56]. They can be classified as geostatistical estimation algorithms and geostatistical simulation algorithms. The estimation algorithms are used to obtain unbiased estimates of the reservoir properties such as porosity and permeability using interpolation techniques. However, the simulation algorithms provide multiple equally probable realizations of the property to be estimated. The simulation algorithms are preferred over estimation algorithms as they incorporate spatial variability of the property while generating multiple realizations. The spatial variability of reservoirs is incorporated in the algorithm using a variogram.

A variogram is a measure of the average correlation of spatially related properties and can be described as shown in equation 4.1.

$$\gamma(h) = \frac{1}{2N(h)}[\{Z(u+h) - Z(u)\}^2] \tag{4.1}$$

$Z(u)$ is the value of the spatial property, h is the lag distance, and $N(h)$ is the number of pairs of points under study.

The sequential Gaussian simulation(SGS) is widely used to generate an ensemble of equiprobable spatial distributions of the reservoir property that are consistent with a specified variogram.

In the case of reservoirs, we know that the parameters are available only at discrete spatial locations, but have a significant impact on the reservoir productivity. Permeability of the rocks is an important property contributing to reservoir productivity. Therefore, studying the spatial variability of the permeability across the reservoir and incorporating it into the temperature modelling framework is important.

In this work, we utilize the generated ensemble of 200 realizations of permeability through SGS utilizing the properties of the variogram in [54]. By doing this, we are developing a permeability distribution across the reservoir. Flow simulation is carried out using CMG-STARS to obtain an ensemble of 200 realizations of temperature and pore pressure. As mentioned earlier, our objective is to obtain PDE models to explain the temperature

dynamics across the reservoir. Generating 200 realizations of temperature facilitates us to extract the spatial parametric variations involved across the reservoir. The permeability which is one of the petrophysical parameters of the reservoir is used as prior physical knowledge and utilized in the hybrid modelling to improve the model accuracy. We develop data-driven and hybrid models that govern the temperature dynamics in an oil reservoir. The obtained PDE models have spatial parametric variation i.e., unlike the case studies shown in sections 2.3 and 3.3, the coefficients of the terms on the right-hand side of the equation will have a spatial dependency. The general parametric PDE with spatial variations is shown in equation 4.2

$$u_t = N(\Theta(u); [\xi_1(x), \xi_2(x), \dots \dots \dots \xi_n(x)]) \quad (4.2)$$

$$\Theta(u) = [u, u^2, u, u_x, u_{xx}, \dots uu_{xx}, u^2 u_{xx} \dots u_y, u_{yy}, \dots uu_{yy}, u^2 u_{yy} \dots u_x u_y \dots] \quad (4.3)$$

where u_t refers to derivatives of the data with respect to time, t. $\Theta(u)$ denotes the candidate library consisting of all the potential terms of the PDE. x *and* y correspond to the spatial dimensions and Θ consists of derivatives of the data with respect to x *and* y, polynomials and their combinations. $\xi_1 \dots \dots \xi_n$ denote the vector of varying coefficients, n being the size of the candidate library. N represents the linear combination of the terms in Θ.

4.2 Modelling temperature dynamics using data-driven and hybrid approaches

We develop PDE models to explain the temperature dynamics in an oil reservoir using a data-driven and hybrid approach. The data for both approaches is obtained using CMG-STARS, a commercial simulator used for reservoir flow simulation. The method of data generation is explained in section 4.2.1 and is taken from the work done by Ganesh et al.,2019 for uncertainty quantification of the factor of safety in a SAGD process [54].

4.2.1 Data generation and visualization

The data utilized to model the temperature dynamics was generated by Ganesh et al.[54]. A sequentially coupled simulation platform, consisting of CMG-STARS for reservoir flow simulation and FLAC3D was used to generate the required data. The coupling framework enables us to visualize the interactions between flow and deformation response in subsurface modelling in the SAGD process. The pore pressure and temperature obtained from CMG-STARS are used to compute the stress and strain through FLAC3D which is utilized to update the porosity and permeability of the reservoir. This updated information is used to re-compute the pore pressure and temperature and the loop continues[57]. As explained in section 4.1, SGS is used to generate realizations of permeability using the properties of the variogram aiding the study of spatial parametric variations. A schematic of the coupled reservoir-geomechanics simulation is shown in figure 4.2[54]

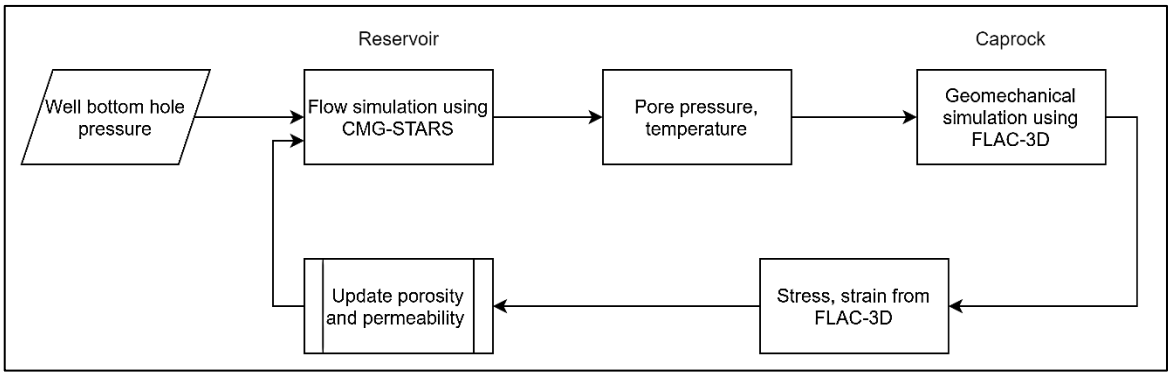

Figure 4.2: Sequentially coupled simulation between the CMG-STARS and FLAC3D

A wellpad of the McKay river oil sand area is selected and utilized for the coupled simulations study. A total of 6 well pairs operate in the pad with well spacing of 100m. The grid sizes and numbers are non-uniform to retain the heterogeneity due to the sedimentation process. The schematic of the reservoir model with different layers and grid sizes is shown in figure 4.4. The Till formation corresponds to the overburden and was set to 120m. Part of the overburden (50m) was included in the CMG-STARS model. (STARS 2015.10). The

caprock corresponds to the Clear Water formations and the reservoir corresponds to the Wabiskaw and McMurray formations. The grid sizes were 5m per cell in the overburden (5m*10 grids); 5m per grid in the caprock (5m*6 grids); 2m per grid in the reservoir (2m*35 grids); and 5m per grid in the underburden (5m*10 grids). The other main parameters for the flow simulator are summarized in Table 4.1. The schematic of the reservoir model with permeability distributions and the geomechanical model is shown in figures 4.3 [54]. Although we are not utilizing the geomechanical model in this work, we have provided information about it as the permeability values are updated based on the geomechanical simulation.

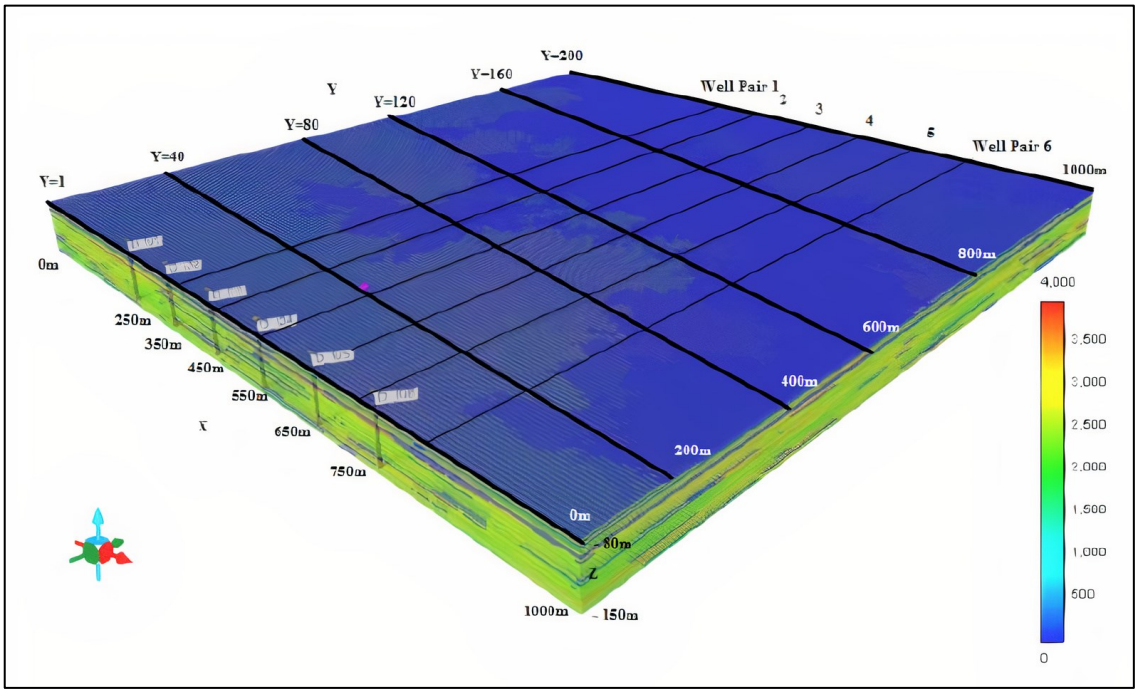

Figure 4.3: The working reservoir model in 3D with permeability I distribution along with the well pairs

Table 4. 1 CMG-STARS simulation parameters [54]

	Porosity	Permeability J (md)	Permeability J (md)	Permeability K (md)	S_o Oil saturation	S_w Water saturation
Overburden	N/A	100	100	50	0.0	1.0
Caprock	0.1	1	1	0.001	0.0	1.0
Reservoir sand	0.3	5000	5000	3330	0.85	0.15
Reservoir shale	0.1	50	50	33	0.15	0.85
Underburden	0.1	1	1	0.001	0.0	1.0

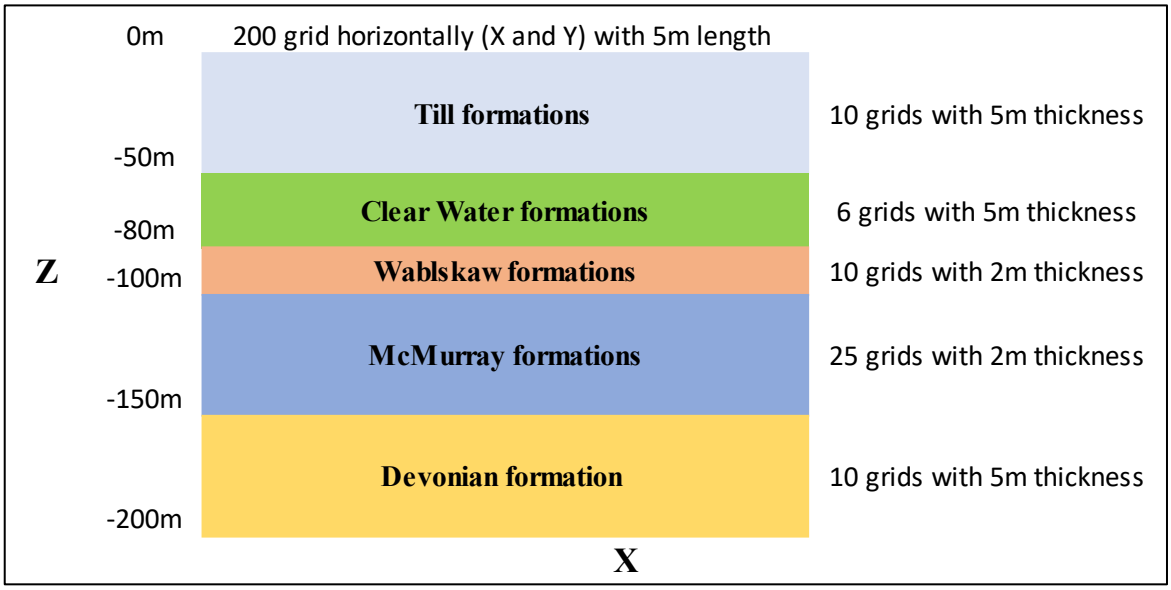

Figure 4.4: Schematic of the reservoir model with grid size

The Wabiskaw and McMurray formations are included in both the reservoir and geomechanical model, whereas the Till, Clearwater, and Devonian formations are not included in the reservoir model for computational efficiency, as the flow in these formations can be neglected due to low permeability. The injectors open from day 1 for preheating of the cold bitumen near the wellbore and the producers open in 150 days for production; hence, for the purpose of modelling, data from day 160 and onwards is considered. A pseudorandom binary sequence (PRBS) input for the well BHP for injectors and producers is designed to provide excitation to the system and generate suitable data for the purposes of modelling and identification.

As we are interested to study spatial parametric variability in the PDE models, we generate an ensemble of realizations. Since we consider working with a two-dimensional reservoir problem, we conduct a single heterogeneous three-dimensional simulation and use each two-dimensional slice perpendicular to the well direction (XZ) as a separate realization. This approach considers the cross-flow terms capturing the true dynamic flow behaviour during the SAGD process, and is, therefore, our preferred approach for creating 2D realizations. To summarize, 200 XZ frames along the Y direction of the 3D model are used as 200 2D realizations to construct the ensemble. The procedure of obtaining 200 realizations from a 3D simulation is shown in figure 4.5 [54].

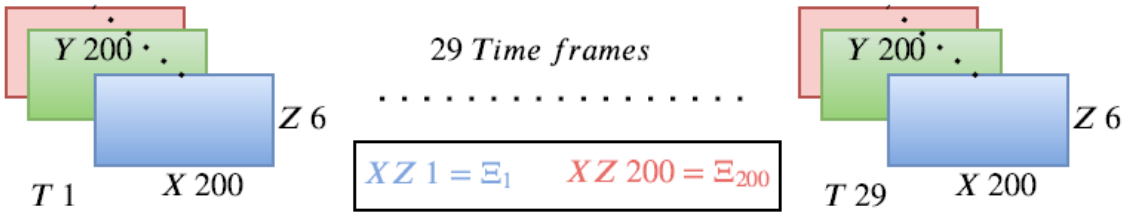

Figure 4.5: Procedure of obtaining 200 realizations

Data at every instant T has 200 grid points in X, 200 grid points in Y and 6 grid points in Z taken over 29 time-frames. In sequential coupling, the permeability distribution is updated at each time step. The posterior permeability distributions obtained after all the updates will be used in hybrid modelling and are depicted in figure 4.6 [54].

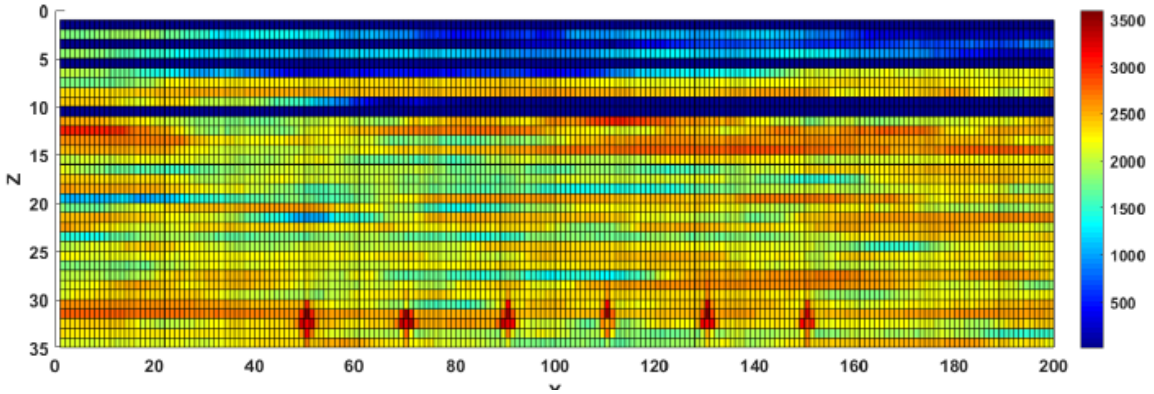

Figure 4.6(a): Average reservoir XZ permeability over all realizations at time t=1.

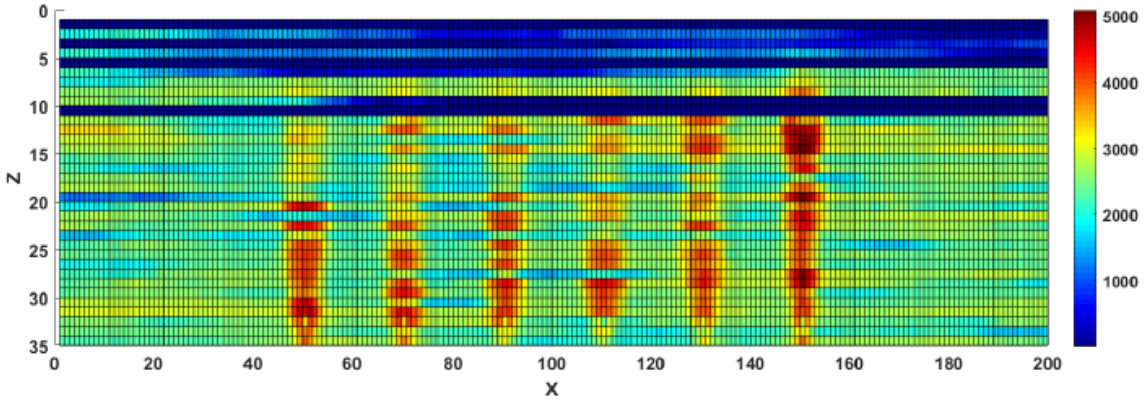

Figure 4.6(b): Average reservoir XZ permeability over all realizations at time t=10.

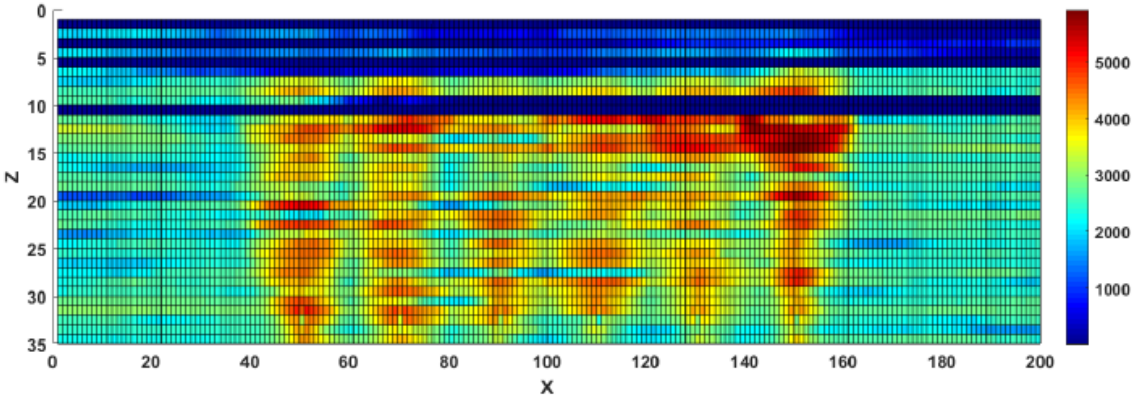

Figure 4.6(c): Average reservoir XZ permeability over all realizations at time t=25.

4.2.2 Parametric PDE discovery

The sparse regression framework explained in chapters 2 and 3 is extended to discover the parametric PDE from multiple realizations. The conclusions made from chapter 2 about the best gradient estimation method and the optimal sparsity regularization were validated in chapter 3 for a system of PDEs. We utilize the same validated approaches to model the temperature dynamics of the oil reservoir. The model accuracies obtained from data-driven and hybrid modelling approaches are compared and detailed results are shown in the results section.

From section 4.2.1, we have 200 realizations of temperature and permeability data obtained from the sequential coupling of CMG-STARS and FLAC3D. Each realization has two spatial dimensions X, Z and temporal dimension T. The process of discovering parametric PDEs involves the disintegration of the ensemble into individual realizations. PDE of constant-coefficient is discovered for each realization using a similar approach and all the 200 PDEs obtained are integrated to form a parametric PDE explaining the temperature dynamics of an oil reservoir. As the realizations are representations of temperature at different spatial locations, the obtained parametric PDE will have the terms which contribute significantly to the temperature dynamics and the coefficients of those terms will be a function of spatial points in the reservoir and will represent the spatial variation of the temperature dynamics.

The temperature dynamics is expressed by the 1-D heat transfer equation in a first principle model[58][59]. The one-dimensional heat transfer ahead of a moving steam chamber interface is expressed as,

$$\frac{\partial^2 T}{\partial \xi} + \left(\frac{U}{a}\right)\left(\frac{\partial T}{\partial \xi}\right) = \left(\frac{1}{\alpha}\right)\left(\frac{\partial T}{\partial t}\right) \tag{4.4}$$

where T is the temperature, ξ in the perpendicular distance from the steam interface,and α is the reservoir thermal diffusivity.

All the first principle-based modelling approaches to obtain models for the SAGD process consider multiple assumptions based on conservation laws and operating conditions. Although the assumptions made are different in each work in the literature, the physics governing the unit processes remains the same. The 1-D heat equation is always considered while explaining the heat transfer in a steam chamber. As we are interested in modelling the temperature dynamics, we tried incorporating some information from the 1-D heat equation into the PDE discovery framework.

Also, from the data generation, we have 200 realizations of the permeability data for the same spatial locations and time frames as the temperature data. Permeability is one of the main petrophysical parameters which is directly related to reservoir productivity and also

to the states of the reservoir. As the temperature is one of the states of the reservoir, we incorporated the permeability data in discovering the PDE for the temperature dynamics.

We discovered the temperature dynamics using two different approaches and compared the models obtained.

a. Data-driven modelling: Although hybrid models proved to generate better PDE models as shown in chapter 3, the development of data-driven models is important as the availability of data for the petrophysical parameter might not be available for some cases. We developed 2 different data-driven models with modifications in the candidate library based on information obtained from the 1-D heat equation. In the first trial, we set the highest order of the polynomial and the spatial derivative to be 2. In the second trial, we considered the normal assumption of the data-driven approach in which the highest order of the polynomial and the derivative is 4. Sparse regression was performed in both methods(table 2.1 and 2.2) to obtain parsimonious PDE models. This process was repeated for 200 realizations and the obtained equations were re-integrated to form a parametric PDE.
b. Hybrid Modelling approach: Incorporating any prior information into the PDE discovery framework will improve the model accuracy and also provides physical interpretability to the obtained model. Hence, we utilized the available permeability data to develop hybrid models for temperature dynamics. The permeability is assumed to be directly proportional to the gradient of temperature in the first principle modelling frameworks. Although some works assume proportionality of permeability to the second-order derivative of the temperature, most works consider the first-order derivative. As the permeability values are available for each spatial location and each time step for the reservoir (Wabiskaw and McMurry formations), the values of the first-order spatial derivative of the data for Wabiskaw and McMurry formations were scaled using the corresponding permeability values. This is equivalent to providing weights to the first-order spatial derivatives using the permeability values before the sparse regression. This procedure was repeated for 200 realizations, and permeability values of that particular realization were used to scale the values of the temperature gradient. After obtaining PDE models for 200

realizations, they were integrated to form a parametric PDE by calculating the average value of the coefficient of each parameter across all 200 realizations to form the PDE governing the temperature dynamics. Similar trials by modification of the candidate library as described for data-driven approaches was performed.

A flowchart summary of the framework utilized for modelling temperature dynamics in an oil reservoir is shown in figure 4.7.

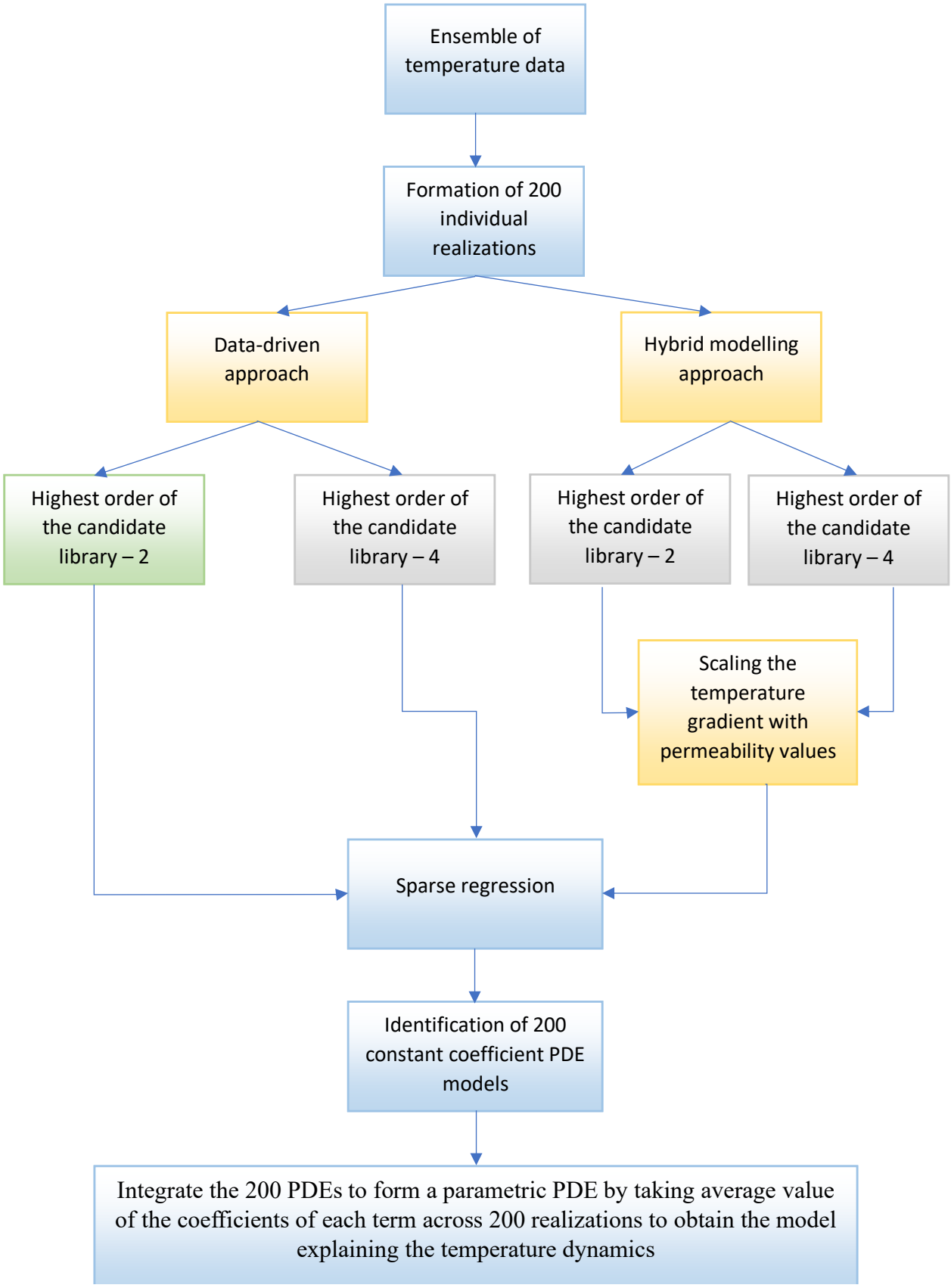

Figure 4.7: Flowchart for discovery of parametric PDE to discover the PDE for temperature dynamics

4.3 Results

The parametric PDE models obtained after re-integrating the individual PDEs resulted from multiple realizations through data-driven and the hybrid approach are presented in this section. x *and* y in the figures correspond to the spatial dimensions. The model accuracies are analysed using the R-Squared(R-sq) and root mean squared error (RMSE). The description of these accuracy metrics is shown in equations (2.14-2.17).

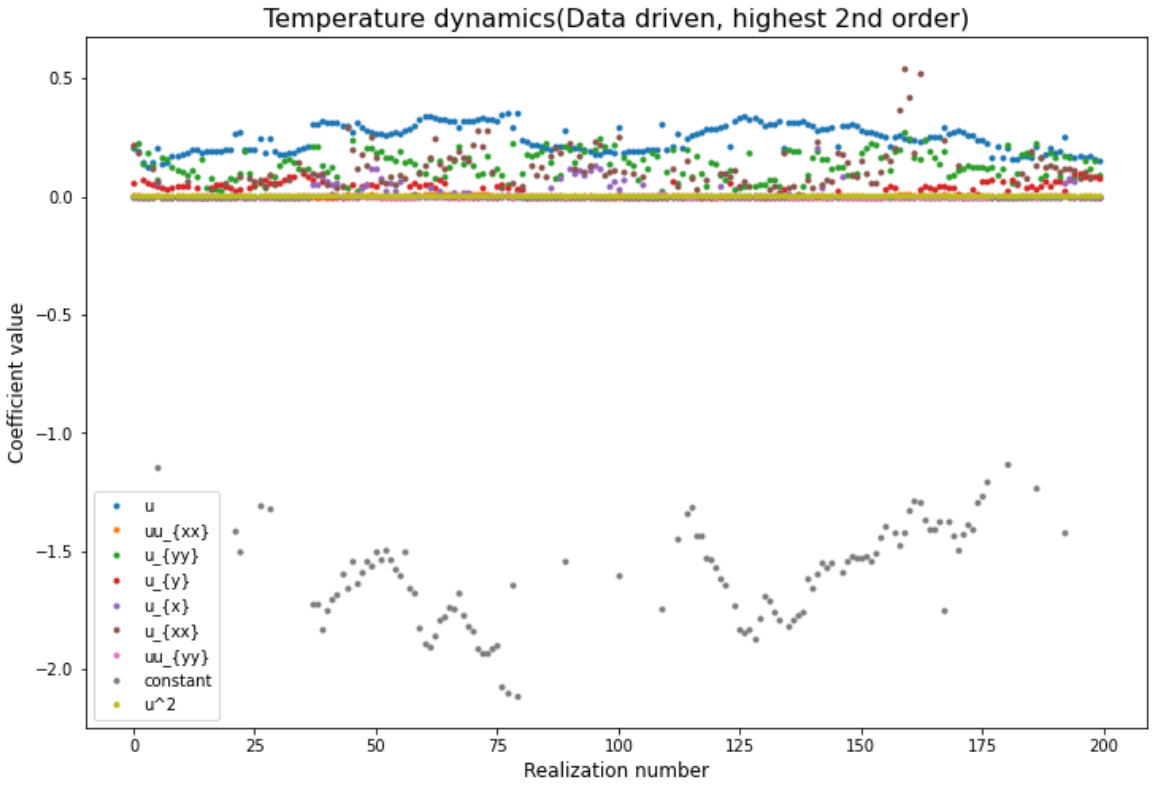

Figure 4.8: Sparse representation of the temperature dynamics - data-driven approach (maximum order- 2)

Figure 4.8 depicts parsimonious solution with the contributing terms and their spatial variation in 200 realizations for the data-driven approach when the maximum order of the library was set to 2. Figure 4.9 represents the sparse representation of the temperature dynamics obtained using the hybrid modelling approach with the highest order of the candidate library being 2.

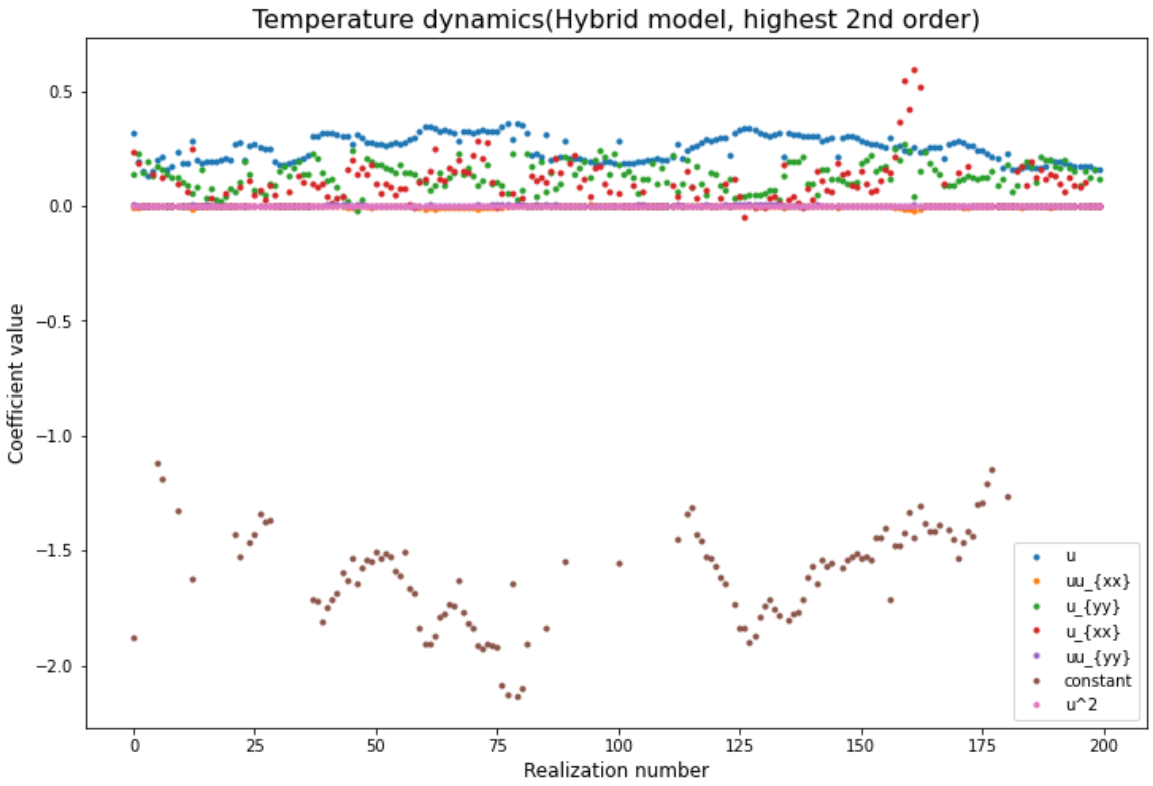

Figure 4. 9 Sparse representation of the temperature dynamics - Hybrid modelling approach (maximum order- 2)

From Figures 4.8 and 4.9, we can observe the presence of common terms and their variations in the data-driven and hybrid modelling approach. However, the hybrid model has fewer terms compared to the data-driven approach. The model accuracy of the hybrid model is higher than that of the data-driven model. Detailed results and values are provided in figure 4.16.

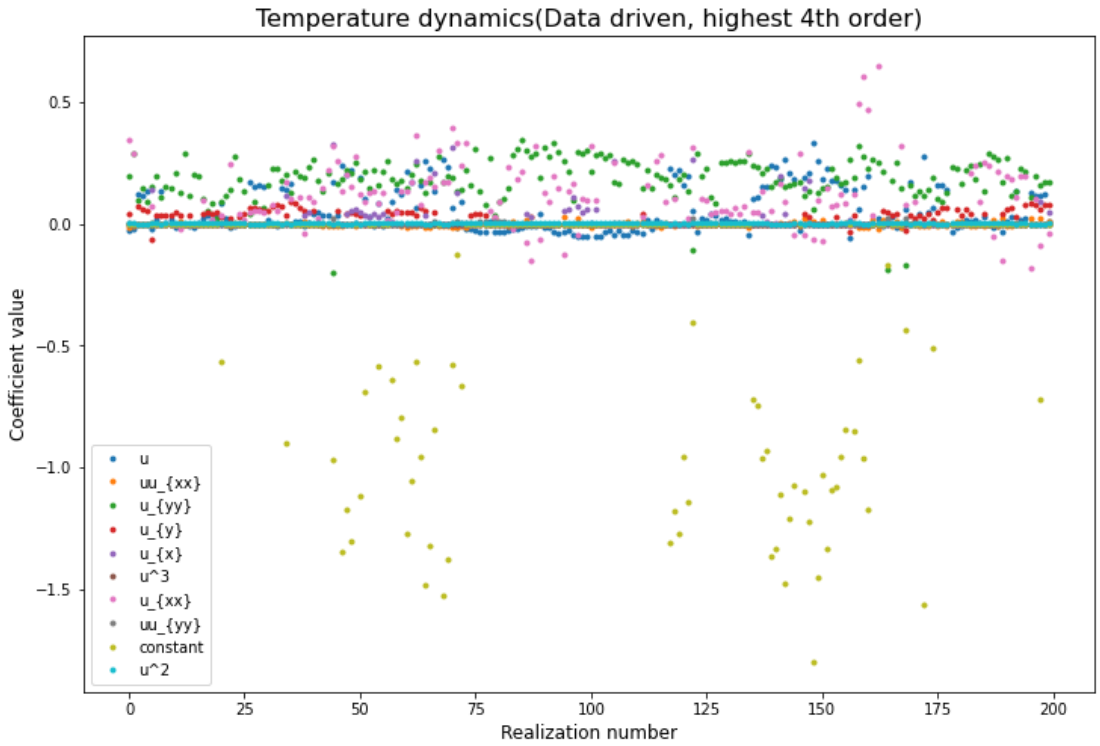

Figure 4.10: Sparse representation of the temperature dynamics – data-driven approach (maximum order- 4)

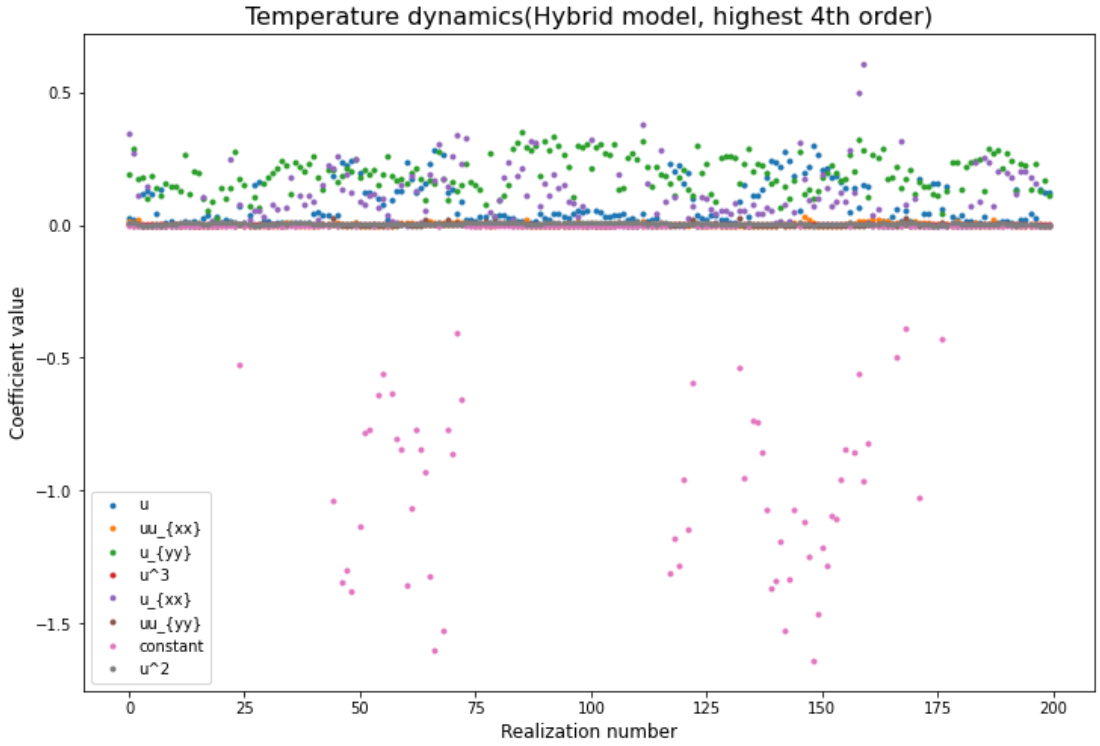

Figure 4. 11 Sparse representation of the temperature dynamics – Hybrid modelling approach (maximum order- 4)

Figures 4.10 and 4.11 represent the temperature dynamics discovered by the data-driven and hybrid modelling approach respectively for the highest order in the candidate library being 4. Comparing the sparse solutions obtained from four different cases, although the model accuracies are varying, the significantly contributing terms remain the same across the realizations and in all the cases.

The R-sq and RMSE values for all four cases are presented in figures 4.12 and 4.13 respectively. The labels for the methods in figures 4.12 and 4.13 are:

Method A – Data-driven approach with highest order in the library as 2.

Method B – Hybrid approach with highest order in the library as 2.

Method C – Data-driven approach with highest order in the library as 4.

Method D – Hybrid approach with highest order in the library as 4.

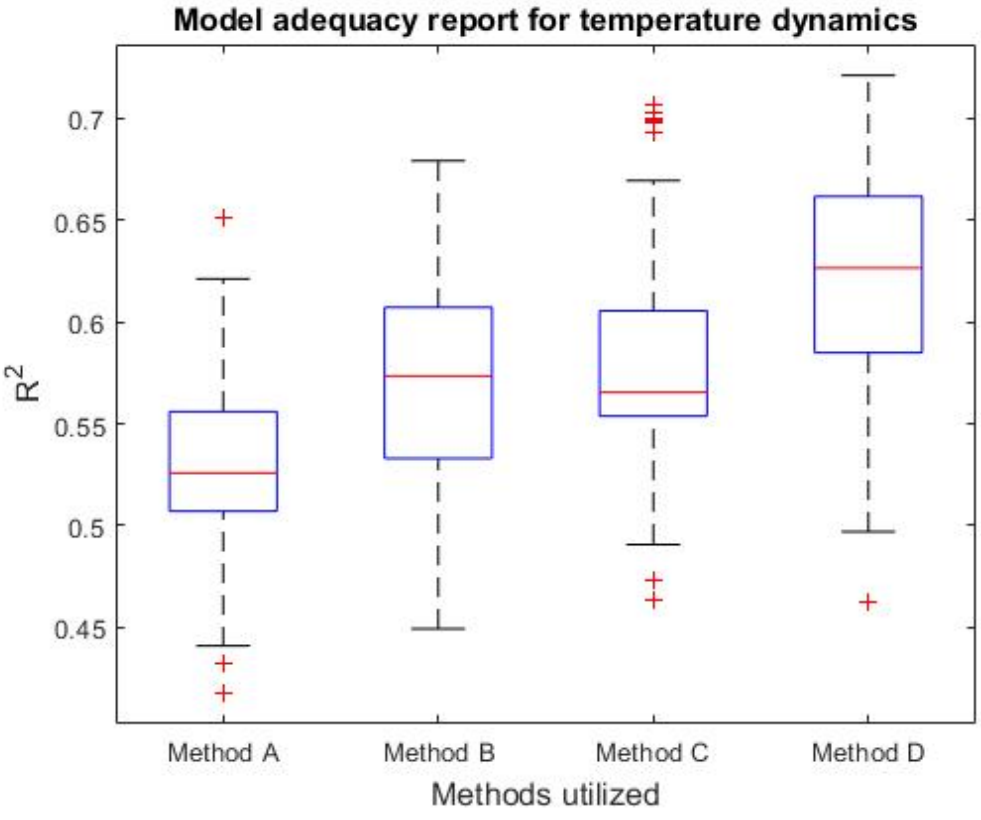

Figure 4. 12 Model adequacy report (R^2) for four different methods

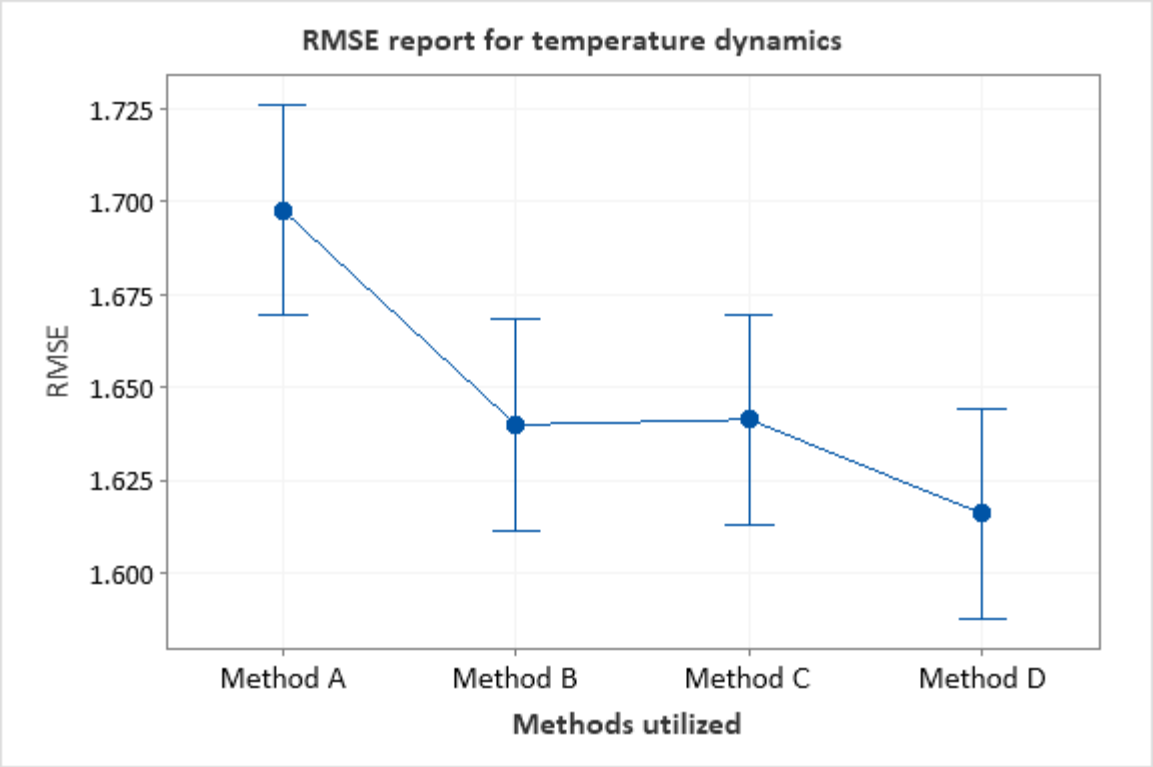

Figure 4. 13 RMSE values for models obtained through different methods

From figures 4.12 and 4.13, we can observe that the hybrid modelling approach is able to generate models with better accuracy. However, limiting the order of the candidate library to 2 cannot be suggested as the accuracy of the models obtained are significantly lower compared to the other case where the order is 4. The magnitude of R-sq values seems to be low compared to the case studied displayed in chapters 2 and 3. This may be due to a lot of uncertainties and the possibility of errors in the data generation framework. For the case studies in chapters 2 and 3, we had solved the PDE using standard solvers and white noise was added to the solution to study the effect of noise. In the case of oil reservoirs, the data is generated using an industrial simulator which is a combination of multiple algorithms and first principle-based models. Obtaining the permeability realization from the variogram involve multiple assumptions and random sampling strategies. Furthermore, CMG-STARS and FLAC3D also involve many steps prone to errors due to the assumptions made in the reservoir model.

We successfully determined the terms involved in explaining the temperature dynamics of the oil reservoir and also presented its spatial variation by considering an ensemble of realizations. Parametric PDE was discovered using the prior physical knowledge about the spatial variation and also by incorporating the permeability values in the algorithm. Further, we tried to build probability distributions of the coefficients of each contributing term obtained from the data-driven approach across realizations. This can be compared with the probability distribution of permeability, which is calculated by taking the average permeability of each realization.

An example of this study is shown below by presenting the probability distributions of permeability, $u \; and \; u_{yy}$. The data was standardized before developing the probability distribution. The probability distributions can be utilized for uncertainty quantification and related studies.

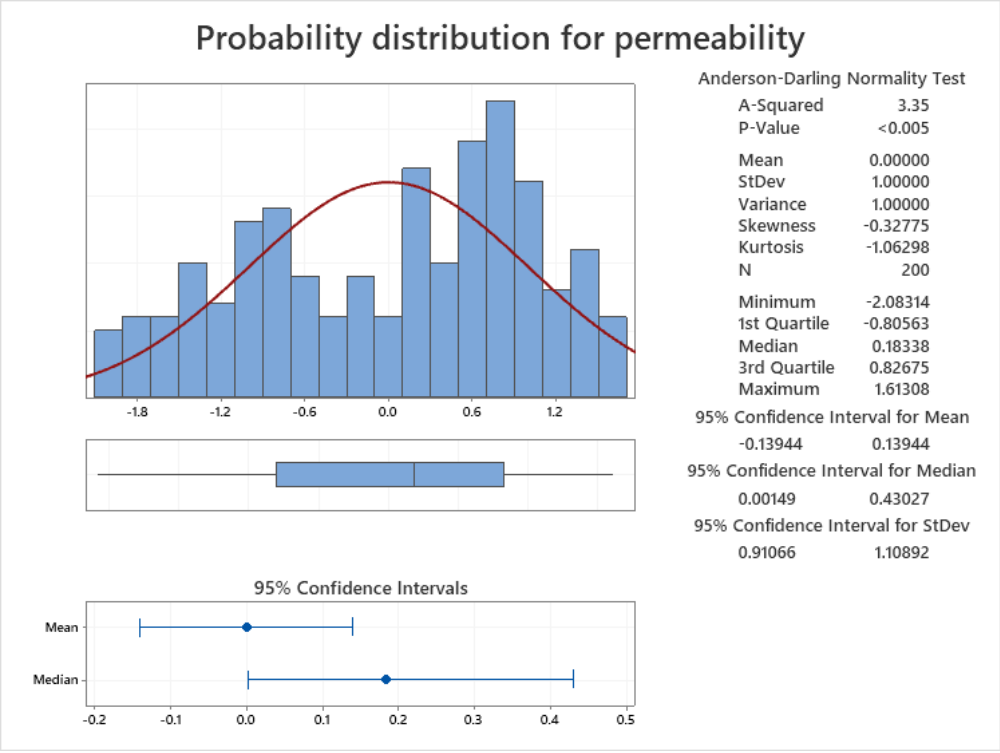

Figure 4. 14 Probability distribution for permeability

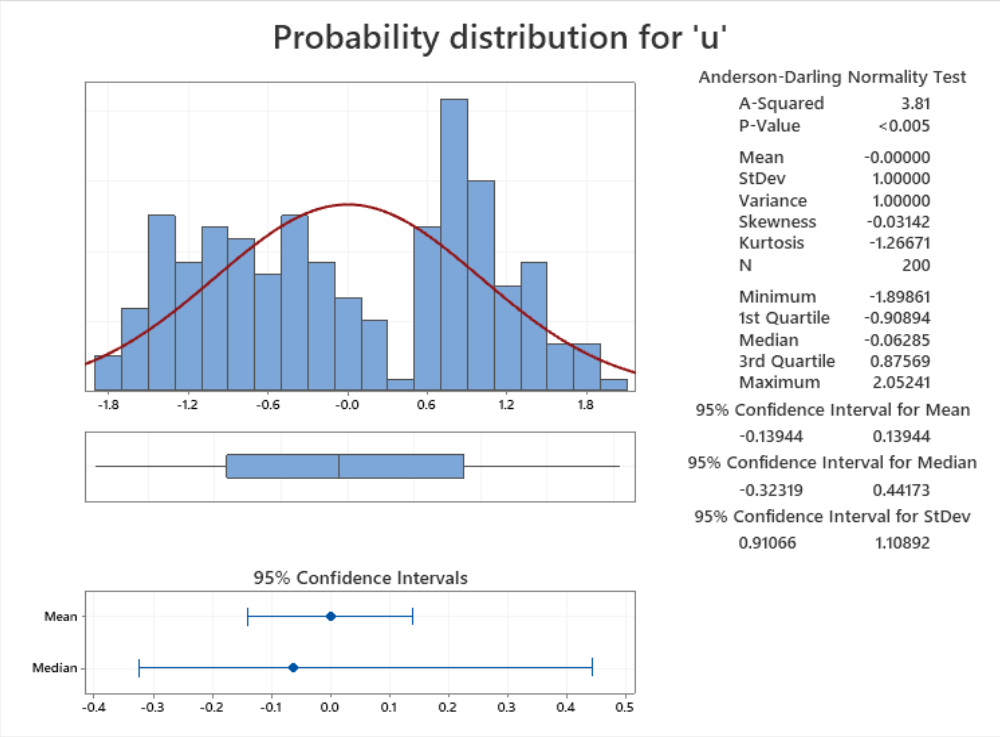

Figure 4. 15 Probability distribution for coefficients of 'u'

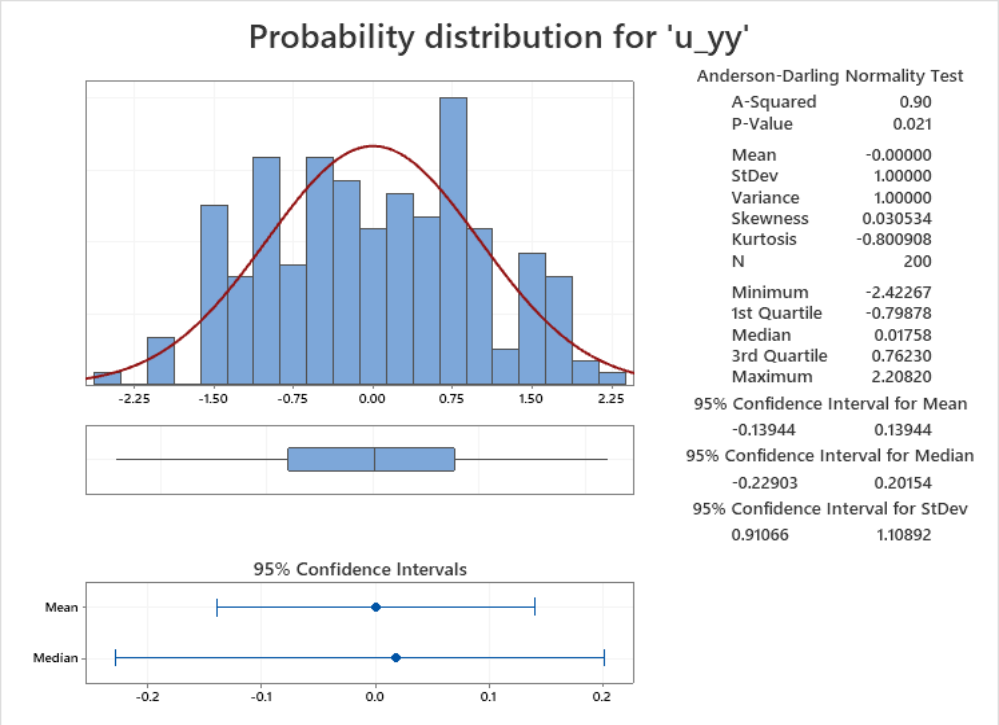

Figure 4. 16 Probability distribution for coefficients of 'u_yy'

4.4 Conclusions

Modelling complex processes and their parameter variations over time and space plays a crucial role in prediction, control, and optimization. In this chapter, we focus on developing parametric PDE models to explain the temperature dynamics of a SAGD process in an oil reservoir. The governing PDE for the temperature dynamics is unknown and the data required for PDE discovery is obtained using commercial industrial simulator CMG-STARS coupled with the FLAC3D simulator. The variograms obtained from surveys and studies are converted to permeability distributions using the SGS algorithm to obtain the distribution of permeability over the spatial domain. An ensemble data containing 200 realizations of temperature and permeability are obtained from the simulators. These realizations span over the entire spatial domain of the reservoir and explain the steam chamber evolution over time as well. We obtain different PDE models for each realization and then integrate them to obtain a PDE that explains the temperature dynamics and the coefficients of the PDE are functions of space,i.e., they have spatial variation. As the 200 realizations are spanned across the reservoir, the spatial variation of the coefficients of the PDE is captured by integrating the individual PDEs of all realizations. We also provide the probability distributions of each term involved in the PDE explaining its spatial variation. A data-driven and hybrid modelling approach is implemented to discover the parametric PDE and the model accuracies are compared. The hybrid model, which was developed by incorporating the permeability values in the discovery algorithm yields the best model to explain the temperature dynamics in an oil reservoir.

Chapter 5

Conclusions

5.1 Summary

The objective of this thesis is to demonstrate the discovery of partial differential equations from spatiotemporal data using data-driven and physics-based hybrid modelling approaches. The first-principles based modelling framework is not feasible for complex processes due to computational cost or insufficient knowledge about the process. A well known data-driven approach to model the PDEs from spatiotemporal data obtained from experiments or industries is the sparse regression framework. However, the algorithm has its limitations and uncertainties. Temporal and spatial gradient estimation in the presence of noise has a significant impact on the model discovery. We performed a detailed study to infer the best possible gradient estimation method for different levels of inherent noise in the data. The reduction in numerical errors in the gradient computation will lead to higher accuracy models. The parsimony in the obtained models is driven by sparse regression. A comparative study of different sparsity regularizations was performed on different systems. The regularization approach which generated higher accuracy models promoting parsimony consistently across systems of a different order was inferred as the optimal norm that can be reliable for PDE discovery.

With the inferences obtained about handling uncertainties in the PDE discovery algorithm from the earlier study, we proposed an approach to discover a system of PDEs using data-driven and hybrid modelling approaches. The computational complexity and cost of the discovery of a system of PDEs were addressed by incorporating prior physical knowledge about the system into the algorithm. We utilized the process knowledge to constrain the candidate library, to enforce the sparse regression to select the known terms of the PDE, and incorporated simplified or functional approximations of the known complex terms of the PDE. The proposed hybrid modelling approach reduced the computational cost

significantly and yielded higher accuracy models for noisy data due to the physical knowledge incorporated in the system. The proposed algorithm was demonstrated on three different complex PDE systems with coupled equations and can be extended to PDE systems with a higher number of equations as well.

The development of PDE models for process parameters in an oil reservoir is a challenging task due to the availability of data only at specific discrete locations. We utilized the variogram properties to generate an ensemble of realizations throughout the spatial domain of the reservoir. This was accomplished by sequential coupling of industrial simulators, CMG-STARS and FLAC3D. Parametric PDE models describing the temperature dynamics in an oil reservoir were obtained by data-driven and hybrid modelling approaches. PDEs discovered for the ensemble was integrated to form a parametric PDE, which ensured spatial dependency of the coefficients of the PDE terms. The probability distributions explaining the spatial dependency of PDE term across the reservoir is also provided for further investigations.

Summarizing chapters 2,3, and 4, we developed data-driven and hybrid modelling frameworks to discover the underlying partial differential equation that governs the process. We inferred the best approaches to handle the uncertainties in the sparse regression framework. The inferences and the algorithm were utilized to discover PDEs, a system of PDEs, and PDEs for complex industrial processes with spatial parametric dependency.

The discovery or the identification of the PDE models has a wide range of applicability. Some of the possible applications in chemical engineering and process control are listed below.

- PDE identification helps in understanding the governing mechanism of an unknown system by identification of the underlying PDE. We may be able to interpret the nature of the process for which only experimental spatiotemporal data is available by analyzing the order of the PDE terms obtained. This will give us a better physical insight to the system under study and reduces the uncertainties involved.
- As the identified PDE explains the propagation of the specific property through time and space, we will be able to study a multidimensional system, such as an oil reservoir, in a lower-dimensional space. The process dynamics of the multiple

dimensions are captured in a lower-dimensional representation using a PDE. The obtained PDE can model the dynamics of multiple spatial dimensions, temporal dimensions, and also spatial and temporal parametric variations. In the case of an oil reservoir, modelling the temperature or pressure dynamics accurately reduces the cost of frequent data measurement from discrete locations and can also help in the reliability estimation of equipments that are working in extreme conditions.

- Modelling differential equations play an important role in reaction kinetics. We can try to model a whole reaction network with a set of equations and identify the driving mechanism of the reaction and also the important nodes of the network. Also, when multiple complex mechanisms are involved, such as integration of chemical reactions and transport within the network, the development of the differential equation model assists reduce the complexity in the development of the solution.
- Discovery of PDE models in a process industry can assist in the fault diagnosis in the process or any particular equipment. For example, any heat loss occurring in a heat exchanger network can be identified or predicted if the model is developed by incorporating historical data involving heat loss under certain conditions which might be challenging and costly through black box approaches.
- The hybrid modelling framework of PDEs assists in improving the working range of the model predictive control(MPC) framework when compared with the data-driven models. Data-driven MPC or soft sensors are limited to the range of the data utilized for the model development, however, the hybrid models will be more efficient even outside the range of the data due to the incorporated physical knowledge of the system.

5.2 Future work

Some possibilities for future work are highlighted below:

- Designing of controllers for processes using the PDE models obtained.
- Developing a graph based network to study causal relationship and identify the important nodes of the system.
- Extending the PDE discovery to chemical reaction engineering domain to obtain data-driven insights to reaction dynamics.
- Performing uncertainty quantification for parametric PDE models obtained.
- Obtaining PDE models by considering uncertainties in the data, for example, missing data, addition of colored noise, etc.
- Performing real-time optimization and development of model predictive control strategies.
- Integrating generative adversarial networks (GANs) to the PDE discovery algorithm to obtain interpretable models from fewer data points.
- Working towards development of a robust hybrid modelling approach that can be applied to multiple processes.

Bibliography

[1] C. Tsay, F. Lejarza, M. A. Stadtherr, and M. Baldea, "Modeling, state estimation, and optimal control for the US COVID-19 outbreak," *Sci. Rep.*, vol. 10, no. 1, pp. 1–12, 2020, doi: 10.1038/s41598-020-67459-8.

[2] M. Zamir, T. Abdeljawad, F. Nadeem, A. Wahid, and A. Yousef, "An optimal control analysis of a COVID-19 model," *Alexandria Eng. J.*, vol. 60, no. 3, pp. 2875–2884, 2021, doi: 10.1016/j.aej.2021.01.022.

[3] J. Bongard and H. Lipson, "Automated reverse engineering of nonlinear dynamical systems," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 104, no. 24, pp. 9943–9948, 2007, doi: 10.1073/pnas.0609476104.

[4] K. Champion, B. Lusch, J. Nathan Kutz, and S. L. Brunton, "Data-driven discovery of coordinates and governing equations," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 116, no. 45, pp. 22445–22451, 2019, doi: 10.1073/pnas.1906995116.

[5] S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Data-driven discovery of partial differential equations," *Sci. Adv.*, vol. 3, no. 4, 2017, doi: 10.1126/sciadv.1602614.

[6] H. Schaeffer, "Learning partial differential equations via data discovery and sparse optimization," *Proc. R. Soc. A Math. Phys. Eng. Sci.*, vol. 473, no. 2197, 2017, doi: 10.1098/rspa.2016.0446.

[7] L. Boninsegna, F. Nüske, and C. Clementi, "Sparse learning of stochastic dynamical equations," *J. Chem. Phys.*, vol. 148, no. 24, 2018, doi: 10.1063/1.5018409.

[8] E. Kaiser, J. N. Kutz, and S. L. Brunton, "Sparse identification of nonlinear dynamics for model predictive control in the low-data limit," *Proc. R. Soc. A Math. Phys. Eng. Sci.*, vol. 474, no. 2219, 2018, doi: 10.1098/rspa.2018.0335.

[9] S. Zhang, G. Lin, and G. Lin, "Robust data-driven discovery of governing physical laws with error bars Subject Areas : Author for correspondence :," *Proc. R. Soc. A Math. Phys. Eng. Sci.*, vol. 474, no. 2245, 2018, [Online]. Available: https://royalsocietypublishing.org/doi/pdf/10.1098/rspa.2018.0305.

[10] Z. Long, Y. Lu, X. Ma, and B. Dong, "PDE-Net: Learning PDEs from data," *35th Int. Conf. Mach. Learn. ICML 2018*, vol. 7, pp. 5067–5078, 2018.

[11] Z. Long, Y. Lu, and B. Dong, "PDE-Net 2.0: Learning PDEs from data with a numeric-symbolic hybrid deep network," *J. Comput. Phys.*, vol. 399, 2019, doi: 10.1016/j.jcp.2019.108925.

[12] G. Both, S. Choudhury, P. Sens, and R. Kusters, "DeepMoD : Deep learning for model discovery in noisy data," *J. Comput. Phys.*, vol. 428, p. 109985, 2021, doi: 10.1016/j.jcp.2020.109985.

[13] H. C. & D. Z. Hao Xu, “DL-PDE: Deep-Learning Based Data-Driven Discovery of Partial Differential Equations from Discrete and Noisy Data,” *Commun. Comput. Phys.*, vol. 29, no. 3, pp. 698–728, 2021, doi: doi:10.4208/cicp.OA-2020-0142.

[14] H. Xu, H. Chang, and D. Zhang, “DLGA-PDE : Discovery of PDEs with incomplete candidate library via combination of deep learning and genetic algorithm,” *J. Comput. Phys.*, vol. 418, p. 109584, 2020, doi: 10.1016/j.jcp.2020.109584.

[15] 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,” *J. Comput. Phys.*, vol. 378, pp. 686–707, 2019, doi: 10.1016/j.jcp.2018.10.045.

[16] M. Raissi and G. E. Karniadakis, “Hidden physics models: Machine learning of nonlinear partial differential equations,” *J. Comput. Phys.*, vol. 357, pp. 125–141, 2018, doi: 10.1016/j.jcp.2017.11.039.

[17] M. Raissi, P. Perdikaris, and G. E. Karniadakis, “Machine learning of linear differential equations using Gaussian processes,” *J. Comput. Phys.*, vol. 348, no. July, pp. 683–693, 2017, doi: 10.1016/j.jcp.2017.07.050.

[18] S. Rudy, A. Alla, S. L. Brunton, and J. N. Kutz, “Data-driven identification of parametric partial differential equations,” *SIAM J. Appl. Dyn. Syst.*, vol. 18, no. 2, pp. 643–660, 2019, doi: 10.1137/18M1191944.

[19] H. Xu, D. Zhang, and J. Zeng, “Deep-learning of parametric partial differential equations from sparse and noisy data,” *Phys. Fluids*, vol. 037132, pp. 1–17, 2021, doi: 10.1063/5.0042868.

[20] A. Ganesh, “Proxy model-based closed loop reservoir management: A data driven approach,” University of Alberta, 2019.

[21] M. Schmidt and H. Lipson, “Distilling Natural Laws,” *Science (80-.).*, vol. 324, no. April, pp. 81–85, 2009.

[22] S. L. Brunton, J. L. Proctor, J. N. Kutz, and W. Bialek, “Discovering governing equations from data by sparse identification of nonlinear dynamical systems,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 113, no. 15, pp. 3932–3937, 2016, doi: 10.1073/pnas.1517384113.

[23] H. Schaeffer, G. Tran, and R. Ward, “Learning Dynamical Systems and Bifurcation via Group Sparsity,” pp. 1–16, 2017, [Online]. Available: http://arxiv.org/abs/1709.01558.

[24] N. M. Mangan, J. N. Kutz, S. L. Brunton, and J. L. Proctor, “Model selection for dynamical systems via sparse regression and information criteria,” *Proc. R. Soc. A Math. Phys. Eng. Sci.*, vol. 473, no. 2204, 2017, doi: 10.1098/rspa.2017.0009.

[25] J. Berg and K. Nyström, “Data-driven discovery of PDEs in complex datasets,” *J. Comput. Phys.*, vol. 384, pp. 239–252, 2019, doi: 10.1016/j.jcp.2019.01.036.

[26] D. A. Messenger and D. M. Bortz, “Weak SINDy For Partial Differential Equations,” pp. 1–26, 2020, doi: 10.1016/j.jcp.2021.110525.

[27] H. Xu, H. Chang, and D. Zhang, “Dl-pde: Deep-learning based data-driven discovery of partial differential equations from discrete and noisy data,” *Commun. Comput. Phys.*, vol. 29, no. 3, pp. 698–728, 2021, doi: 10.4208/CICP.OA-2020-0142.

[28] J. H. Lagergren, J. T. Nardini, G. Michael Lavigne, E. M. Rutter, and K. B. Flores, “Learning partial differential equations for biological transport models from noisy spatio-temporal data,” *Proc. R. Soc. A Math. Phys. Eng. Sci.*, vol. 476, no. 2234, 2020, doi: 10.1098/rspa.2019.0800.

[29] W. E. Schiesser, *Computational Mathematics in Engineering and Applied Science(1st ed.)*. CRC Press, 1994.

[30] R. J. LeVeque, “Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems,” *SIAM*, vol. 98, 2007.

[31] R. J. R. I. Knowles, “Methods for numerical differentiation of noisy data,” *Electron. J. Differ. Eq.*, pp. 235–246, 2014.

[32] O. Bruno and D. Hoch, “Numerical differentiation of approximated functions with limited order-of-accuracy deterioration,” *SIAM J. Numer. Anal*, vol. 50, no. 3, pp. 1581–1603, 2012.

[33] G. D. Hesthaven JS, Gottlieb S, “Spectral methods for time-dependent problems,” *Cambridge, UK Cambridge Univ. Press*, vol. 21, 2007.

[34] D. P. Kingma and J. Lei Ba, “ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION,” in *Iclr*, 2015, pp. 1–15, [Online]. Available: https://arxiv.org/pdf/1412.6980.pdf %22 entire document.

[35] H. Zou and T. Hastie, “Regularization and variable selection via the elastic net,” *J. R. Stat. Soc. Ser. B Stat. Methodol.*, vol. 67, no. 5, p. 768, 2005, doi: 10.1111/j.1467-9868.2005.00527.x.

[36] S. L. Brunton and J. N. Kutz, *Data Driven Science & Engineering - Machine Learning, Dynamical Systems, and Control*. 2017.

[37] B. P. Epps and E. M. Krivitzky, “Singular value decomposition of noisy data: noise filtering,” *Exp. Fluids*, vol. 60, no. 8, pp. 1–23, 2019, doi: 10.1007/s00348-019-2768-4.

[38] B. P. Epps and E. M. Krivitzky, “Singular value decomposition of noisy data: mode corruption,” *Exp. Fluids*, vol. 60, no. 8, pp. 1–30, 2019, doi: 10.1007/s00348-019-2761-y.

[39] J. N. Kutz, *Data-Driven Modeling & Scientific Computation: Methods for Complex Systems & Big Data*. Oxford University Press, London, 2013.

[40] R. Miura, “Korteweg-de Vries Equation and Generalizations. I. A Remarkable

Explicit Nonlinear Transformation," *J. Math. Phys.*, vol. 9, no. 8, pp. 1202–1204, 1968.

[41] A. Rajendran and M. Mazzotti, "Local equilibrium theory for the binary chromatography of species subject to a generalized langmuir isotherm. 2. Wave interactions and chromatographic cycle," *Ind. Eng. Chem. Res.*, vol. 50, no. 1, pp. 352–377, 2011, doi: 10.1021/ie1015798.

[42] S. C. Anco and A. F. Cheviakov, "On the different types of global and local conservation laws for partial differential equations in three spatial dimensions: Review and recent developments," *Int. J. Non. Linear. Mech.*, vol. 126, no. March, 2020, doi: 10.1016/j.ijnonlinmec.2020.103569.

[43] E. F. Keller and L. A. Segel, "Initiation of slime mold aggregation viewed as an instability," *J. Theor. Biol.*, vol. 26, no. 3, pp. 399–415, 1970, doi: 10.1016/0022-5193(70)90092-5.

[44] H. J. A. F. Tulleken, "Grey-box modelling and identification using physical knowledge and bayesian techniques," *Automatica*, vol. 29, no. 2, pp. 285–308, 1993, doi: 10.1016/0005-1098(93)90124-C.

[45] Q. Xiong and A. Jutan, "Grey-box modelling and control of chemical processes," *Chem. Eng. Sci.*, vol. 57, no. 6, pp. 1027–1039, 2002, doi: 10.1016/S0009-2509(01)00439-0.

[46] B. Sohlberg, "Grey box modelling for model predictive control of a heating process," *J. Process Control*, vol. 13, no. 3, pp. 225–238, 2003, doi: 10.1016/S0959-1524(02)00030-6.

[47] B. Sohlberg and E. W. Jacobsen, *Grey Box Modelling – Branches and Experiences*, vol. 41, no. 2. IFAC, 2008.

[48] J. Schubert, R. Simutis, M. Dors, I. Havlik, and A. Lübbert, "Bioprocess optimization and control: Application of hybrid modelling," *J. Biotechnol.*, vol. 35, no. 1, pp. 51–68, 1994, doi: 10.1016/0168-1656(94)90189-9.

[49] T. Taguchia and Y. Yamashitab, "A Hybrid Approach for Process Optimization of Distillation Reflux Condition using First Principle Models and Least Squares Regression," *Comput. Aided Chem. Eng.*, vol. 44, pp. 229–234, 2018, doi: 10.1016/B978-0-444-64241-7.50033-1.

[50] P. Bhalode, N. Metta, Y. Chen, and M. Ierapetritou, *Efficient Data-based Methodology for Model enhancement and Flowsheet analyses for Continuous Pharmaceutical Manufacturing*, vol. 48, no. 1. Elsevier Masson SAS, 2020.

[51] M. von Stosch, R. Oliveira, J. Peres, and S. Feyo de Azevedo, "Hybrid semi-parametric modeling in process systems engineering: Past, present and future," *Comput. Chem. Eng.*, vol. 60, pp. 86–101, 2014, doi: 10.1016/j.compchemeng.2013.08.008.

[52] R. M. Butler, "" STEAM-ASSISTED GRAVITY DRAINAGE : CONCEPT , Stleam-assisted Gravity Drainage : Concept , Development , Performance and

Future University of Calgary," *J. Can. Pet. Technol.*, vol. 33(02), pp. 44–50, 1994.

[53] A. Ganesh, J. Xiong, R. J. Chalaturnyk, and V. Prasad, "Proxy models for caprock pressure and temperature dynamics during steam-assisted gravity drainage process," *Comput. Chem. Eng.*, vol. 121, pp. 594–607, 2019, doi: 10.1016/j.compchemeng.2018.10.023.

[54] A. Ganesh, B. Zhang, R. J. Chalaturnyk, and V. Prasad, "Uncertainty quantification of the factor of safety in a steam-assisted gravity drainage process through polynomial chaos expansion," *Comput. Chem. Eng.*, vol. 133, pp. 1–15, 2020, doi: 10.1016/j.compchemeng.2019.106663.

[55] H. Xu, D. Zhang, and J. Zeng, "Deep-learning of parametric partial differential equations from sparse and noisy data," *Phys. Fluids*, vol. 33, no. 3, pp. 1–30, 2021, doi: 10.1063/5.0042868.

[56] A. Raghu, X. Yang, S. Khare, J. Prakash, B. Huang, and V. Prasad, "Reservoir history matching using constrained ensemble Kalman filtering," *Can. J. Chem. Eng.*, vol. 96, no. 1, pp. 145–159, 2018, doi: 10.1002/cjce.22965.

[57] J. Kim, H. A. Tchelepi, and R. Juanes, "Stability and convergence of sequential methods for coupled flow and geomechanics: Drained and undrained splits," *Comput. Methods Appl. Mech. Eng.*, vol. 200, no. 23–24, pp. 2094–2116, 2011, doi: 10.1016/j.cma.2011.02.011.

[58] J. . Carslaw, H.S., Jaeger, *Conduction of Heat in Solids*. Oxford University Press, London, 1959.

[59] Q. Zhang, H. Liu, X. Dong, Y. Liu, G. Li, and Y. Wang, "A new comprehensive model to estimate the steam chamber expansion and recovery performance of entire SAGD process," *J. Pet. Sci. Eng.*, vol. 185, no. April 2019, p. 106629, 2020, doi: 10.1016/j.petrol.2019.106629.