IDENTIFICATION OF PHYSICAL PROCESSES VIA DATA DRIVEN METHODS

By

HARSHA VARDHAN REDDY VADDIREDDY

Bachelor of Engineering in Aerospace Engineering
Hindustan University
Chennai, India
2014

Master of Technology in Aerodynamics & Flight Mechanics
Indian Institute of Space Science & Technology
Trivandrum, India
2016

Submitted to the Faculty of the
Graduate College of
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
May, 2020

IDENTIFICATION OF PHYSICAL PROCESSES VIA DATA DRIVEN METHODS

Thesis Approved:

Dr. Omer San

Thesis Advisor

Dr. Arvind Santhanakrishnan

Dr. Kursat Kara

ACKNOWLEDGMENTS

I would like to thank my family and friends for constantly encouraging me all my life and my advisor, Dr. Omer San, for sharing his unparalleled knowledge and exemplary guidance throughout last two years. I acknowledge Dr. Adil Rasheed from Norwegian University of Science and Technology for his critical insights and guidance. I also indebted to my lab members Suraj Pawar and Shady Ahmed for there valuable help in both research and course works. Financial assistance from the US DOE Office of Science is deeply appreciated.

Acknowledgments reflect the views of the author and are not endorsed by committee members or Oklahoma State University.

To my father and mother for their unconditional love.

Dedication reflect the views of the author and are not endorsed by committee members or Oklahoma State University.

Name: HARSHA VARDHAN REDDY VADDIREDDY

Date of Degree: May, 2020

Title of Study: IDENTIFICATION OF PHYSICAL PROCESSES VIA DATA DRIVEN METHODS

Major Field: MECHANICAL & AEROSPACE ENGINEERING

Abstract: Extracting governing equations from data can be viewed as reverse engineering of Nature - using data to identify the physical laws/models. This approach is crucial for fields where data is abundant (such as geophysical flows, finance, and neuroscience) but the physical laws based on the first principles are not available. In recent years, the use of machine learning (ML) methods complemented the need for formulating mathematical models through the application of data analysis algorithms that allow accurate estimation of observed dynamics by learning automatically from the given observations. The neural networks and symbolic regression (SR) based approaches are the most popular ML frameworks used to learn the underlying physical process by only the observing data. While neural network approaches have shown great promise, its black-box nature makes it difficult to interpret the learned models. On the other hand, symbolic regression algorithms are capable of learning/finding an analytically tractable function in symbolic form. Hence to address the functional expressibility, a key limitation of the black-box machine learning methods, this study has explored the use of symbolic regression approaches for identifying relations and operators that accurately represent the underlying physical processes. This study demonstrates the use of an evolutionary algorithm called gene expression programming (GEP) and a sparse optimization algorithm called sequential threshold ridge regression (STRidge) in discovering physical models. The effectiveness of these algorithms is demonstrated on four different applications: (i) partial differential equation (PDE) discovery, (ii) truncation error analysis, (iii) hidden physics discovery and (iv) discovering subgrid-scale closure models. This study shows the GEP and STRidge algorithms are able to distill various linear/nonlinear PDEs, truncation error terms and unknown source terms of 1D and 2D PDEs. Furthermore, the classical Smagorinsky model is identified for subgrid-scale (SGS) closure from an array of tailored features in solving the 2D Kraichnan turbulence problem. Our results demonstrate the huge potential of these techniques in distilling complex nonlinear physics models from only observing the data. Furthermore, this study reveals the importance of feature selection/feature engineering and embedding the prior knowledge about the unknown dynamical system in terms of invariances for identifying models.

TABLE OF CONTENTS

Chapter Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Figure Page

ABBREVIATIONS

The following abbreviations are used in this manuscript:

ANN	Artificial neural networks
CS	Compressive sensing
DNS	Direct numerical simulation
EBR	Elite bases regression
FFX	Fast function extraction
GEP	Gene expression programming
GP	Genetic Programming
LASSO	Least absolute shrinkage and selection operator
LES	Large eddy simulation
MAP	Maximum a posteriori
MDE	Modified differential equation
MSE	Mean squared error
OLS	Ordinary least square
PDE-FIND	PDE-functional identification of nonlinear dynamics
RANS	Reynolds-averaged Navier-Stokes equations
SGS	Sub grid scale
SINDy	Sparse identification of nonlinear dynamics
SR	Symbolic regression
STLS	Sequential threshold least squares
STRidge	Sequential threshold ridge

CHAPTER I

Introduction

This thesis puts forth a modular approach for distilling hidden flow physics from discrete and sparse observations. To address functional expressiblity, a key limitation of the black-box machine learning methods, this study exploits the use of symbolic regression as a principle for identifying relations and operators that are related to the underlying processes. This approach combines evolutionary computation with feature engineering to provide a tool for discovering hidden parameterizations embedded in the trajectory of fluid flows in the Eulerian frame of reference. The presented approach in this study mainly involves gene expression programming (GEP) and sequential threshold ridge regression (STRidge) algorithms. Results have been demonstrated in three different applications: (i) equation discovery, (ii) truncation error analysis, and (iii) hidden physics discovery, for which we include both predicting unknown source terms from a set of sparse observations and discovering subgrid scale closure models. It is concluded that both GEP and STRidge algorithms are able to distill the Smagorinsky model from an array of tailored features in solving the Kraichnan turbulence problem. Presented results demonstrate the huge potential of these techniques in complex physics problems, and reveal the importance of feature selection and feature engineering in model discovery approaches[1].

[1]The work presented in this thesis has been published in Vaddireddy et al. (2020).

1.1 Motivation

Since the dawn of mathematical modelling of complex physical processes, scientists have been attempting to formulate predictive models to infer current and future states. These first principle models are generally conceptualized from conservation laws, sound physical arguments, and empirical heuristics drawn from either conducting experiments or hypothesis made by an insightful researcher. However, there are many complex systems (some being climate science, weather forecasting, and disease control modelling) with their governing equations known partially and their hidden physics await to be modelled. In the last decade, there have been rapid advances in machine learning (Jordan and Mitchell, 2015; Marx, 2013) and easy access to rich data, thanks to the plummeting costs of sensors and high performance computers.

This paradigm shift in data driven techniques can be readily exploited to distill new or improved physical models for nonlinear dynamical systems. Extracting predictive models based on observing complex patterns from vast multimodal data can be loosely termed as reverse engineering nature. This approach is not particularly new, for example, Kepler used planets' positional data to approximate their elliptic orbits. The reverse engineering approach is most appropriate in the modern age as we can leverage computers to directly infer physical laws from data collected from omnipresent sensors that otherwise might not be comprehensible to humans. Symbolic regression methods are a class of data driven algorithms that aim to find a mathematical model that can describe and predict hidden physics from observed input-response data. Some of the popular machine learning techniques that are adapted for the task of symbolic regression are neural networks (Rosenblatt, 1958; LeCun et al., 2015), compressive sensing/sparse optimization (Candes et al., 2008; Candes and Wakin, 2008), and evolutionary algorithms (Koza, 1992; Ferreira, 2001).

1.2 Evolutionary Algorithms

Symbolic regression (SR) approaches based on evolutionary computation (Koza, 1992; Ferreira, 2006) are a class of frameworks that are capable of finding analytically tractable functions. Traditional deterministic regression algorithms assume a mathematical form and only find parameters that best fit the data. On the other hand, evolutionary SR approaches aim to simultaneously find parameters and also learn the best-fit functional form of the model from input-response data. Evolutionary algorithms search for functional abstractions with a preselected set of mathematical operators and operands while minimizing the error metrics. Furthermore, the optimal model is selected from Pareto front analysis with respect to minimizing accuracy versus model complexity. Genetic programming (GP) (Koza, 1992) is a popular choice leveraged by most of the SR frameworks. GP is an extended and improved version of a genetic algorithm (GA) (Mitchell, 1998; Holland, 1992) which is inspired by Darwin's theory of natural evolution. Seminal work was done in identifying hidden physical laws (Schmidt and Lipson, 2009; Bongard and Lipson, 2007) from the input-output response using the GP approach. GP has been applied in the context of the SR approach in digital signal processing (Yang et al., 2005), nonlinear system identification (Ferariu and Patelli, 2009) and aerodynamic parametric estimation (Luo et al., 2015). Furthermore, GP as an SR tool was applied to identify complex closed-loop feedback control laws for turbulent separated flows (Brunton and Noack, 2015; Gautier et al., 2015; Duriez et al., 2015; Debien et al., 2016). Hidden physical laws of the evolution of a harmonic oscillator based on sensor measurements and the real world prediction of solar power production at a site were identified using GP as an SR approach (Quade et al., 2016).

Improved versions of GP focus on better representation of the chromosome, which helps in the free evolution of the chromosome with constraints on the complexity of its growth, and faster searches for the best chromosome. Some of these improved versions

of GP are gene expression programming (GEP) (Ferreira, 2001), parse matrix evolution (PME) (Luo and Zhang, 2012), and linear genetic programming (LGP) (Brameier and Banzhaf, 2007). GEP takes advantage of the linear coded chromosome approach from GA and the parse tree evolution of GP to alleviate the disadvantages of both GA and GP. GEP was applied to diverse applications as an SR tool to recover nonlinear dynamical systems (Faradonbeh and Monjezi, 2017; Faradonbeh et al., 2017; Hoseinian et al., 2017; Çanakcı et al., 2009). Recently, GEP was modified for tensor regression, termed as multi-GEP, and has been applied to recover functional models approximating the nonlinear behavior of the stress tensor in the Reynolds-averaged Navier-Stokes (RANS) equations (Weatheritt and Sandberg, 2016). Furthermore, this novel algorithm was extended to identify closure models in a combustion setting for large eddy simulations (LES) (Schoepplein et al., 2018). Similarly, a new damping function has been discovered using the GEP algorithm for the hybrid RANS/LES methodology (Weatheritt and Sandberg, 2017). Generally, evolutionary based SR approaches can identify models with complex nonlinear compositions given enough computational time.

1.3 Sparse Optimization/Compressive Sensing

Compressive sensing (CS) (Candes et al., 2008; Candes and Wakin, 2008) is predominately applied to signal processing in seeking the sparsest solution (i.e., a solution with the fewest number of features). Basis pursuit algorithms (Rauhut, 2010), also identified as sparsity promoting optimization techniques (Tibshirani, 1996; James et al., 2013), play a fundamental role in CS. Ordinary least squares (OLS) optimization generally results in identifying models with large complexity which are prone to overfitting. In sparse optimization, the OLS objective function is regularized by an additional constraint on the coefficient vector. This regularization helps in taming and shrinking large coefficients and thereby promoting sparsity in feature selection

and avoiding overfitted solutions. The least absolute shrinkage and selection operator (LASSO) (Tibshirani, 1996; Tibshirani et al., 2015) is one of the most popular regularized least squares (LS) regression methods. In LASSO, an L_1 penalty is added to the LS objective function to recover sparse solutions (Candes et al., 2006). In Bayesian terms, LASSO is a maximum a posteriori estimate (MAP) of LS with Laplacian priors. LASSO performs feature selection and simultaneously shrinks large coefficients which may manifest to overfit the training data. Ridge regression (Murphy, 2012) is another regularized variant where an L_2 penalty is added to the LS objective function. Ridge regression is also defined as a MAP estimate of LS with a Gaussian prior. The L_2 penalty helps in grouping multiple correlated basis functions and increases robustness and convergence stability for ill-conditioned systems. The elastic net approach (Zou and Hastie, 2005; Friedman et al., 2010) is a hybrid of the LASSO and ridge approaches combining the strengths of both algorithms.

Derived from these advances, a seminal work was done in employing sparse regression to identify the physical laws of nonlinear dynamical systems (Brunton et al., 2016). This work leverages the structure of sparse physical laws, i.e., only a few terms represent the dynamics. The authors have constructed a large feature library of potential basis functions that has the expressive power to define the dynamics and then seek to find a sparse feature set from this overdetermined system. To achieve this, a sequential threshold least squares (STLS) algorithm (Brunton et al., 2016) has been introduced in such a way that a hard threshold on OLS coefficients is performed recursively to obtain sparse solutions. This algorithm was leveraged to form a framework called sparse identification of nonlinear dynamics (SINDy) (Brunton et al., 2016) to extract the physical laws of nonlinear dynamical systems represented by ordinary differential equations (ODEs). This work re-envisioned model discovery from the perspective of sparse optimization and compressive sensing. The SINDy framework recovered various benchmark dynamical systems such as the chaotic Lorenz

system and vortex shedding behind a cylinder. However, STLS regression finds it challenging to discover physical laws that are represented by spatio-temporal data or high-dimensional measurements and have highly correlated features in the basis library. This limitation was addressed using a regularized variant of STLS called the sequential threshold ridge regression (STRidge) algorithm (Rudy et al., 2017). This algorithm was intended to discover unknown governing equations that are represented by partial differential equations (PDEs), hence forming a framework termed as PDE-functional identification of nonlinear dynamics (PDE-FIND) (Rudy et al., 2017). PDE-FIND was applied to recover canonical PDEs representing various nonlinear dynamics. This framework also performs reasonably well under the addition of noise to data and measurements. These sparse optimization frameworks generally have a free parameter associated with the regularization term that is tuned by the user to recover models ranging from highly complex to parsimonious.

In a similar direction of discovering governing equations using sparse regression techniques, L_1 regularized LS minimization was used to recover various nonlinear PDEs (Schaeffer et al., 2013; Schaeffer, 2017) using both high fidelity and distorted (noisy) data. Additionally, limited and distorted data samples were used to recover chaotic and high-dimensional nonlinear dynamical systems (Tran and Ward, 2017; Schaeffer et al., 2018). To automatically filter models with respect to model complexity (number of terms in the model) versus test accuracy, Bayes information criteria were used to rank the most informative models (Mangan et al., 2017). Furthermore, SINDy coupled with model information criteria is used to infer canonical biological models (Mangan et al., 2016) and introduce a reduced order modelling (ROM) framework (Loiseau et al., 2018). STRidge (Rudy et al., 2017) was applied as a deterministic SR method to derive algebraic Reynolds-stress models for the RANS equations (Schmelzer et al., 2018). Recently, various sparse regression algorithms like LASSO (Tibshirani, 1996), STRidge (Rudy et al., 2017), sparse relaxed regularized

regression (Zheng et al., 2018), and the forward-backward greedy algorithm (Zhang, 2009) were investigated to recover truncation error terms of various modified differential equations (MDEs) coming from canonical PDEs (Thaler et al., 2019). The frameworks discussed above assume that the structure of the model to be recovered is sparse in nature; that is, only a small number of terms govern the dynamics of the system. This assumption holds for many physical systems in science and engineering.

Fast function extraction (FFX) (McConaghy, 2011) is another deterministic SR approach based on pathwise regularized learning that is also called the elastic net algorithm Zou and Hastie (2005). The resulting models of FFX are selected through non-dominated filtering concerning accuracy and model complexity, similar to evolutionary computations. FFX is influenced by both GP and CS to better distill physical models from data. FFX has been applied to recover hidden physical laws (Quade et al., 2016), canonical governing equations (Vaddireddy and San, 2019) and Reynolds stress models for the RANS equations (Schmelzer et al., 2019). Some other potential algorithms for deterministic SR are elite bases regression (EBR) (Chen et al., 2017) and prioritized grammar enumeration (PGE) (Worm and Chiu, 2013). EBR uses only elite features in the search space selected by measuring correlation coefficients of features for the target model. PGE is another deterministic approach that aims for the substantial reduction of the search space where the genetic operators and random numbers from GP are replaced with grammar production rules and systematic choices.

1.4 Neural Networks

An artificial neural network (ANN), also referred to as deep learning if multiple hidden layers are used, is a machine learning technique that transforms input features through nonlinear interactions and maps to output target features (Rosenblatt, 1958; LeCun et al., 2015). ANNs attracted attention in recent times due to their exemplary performance in modelling complex nonlinear interactions across a wide range of applications

including image processing (Ciregan et al., 2012), video classification (Karpathy and Fei-Fei, 2015) and autonomous driving (Sallab et al., 2017). ANNs produce black-box models that are not quite open to physical inference or interpretability. Recently, physics-informed neural networks (PINNs) (Raissi et al., 2019) were proposed in the flavor of SR that is capable of identifying scalar parameters for known physical models. PINNs use a loss function in symbolic form to help ANNs adhere to the physical structure of the system. Along similar directions, a Gaussian process regression (GPR) has been also investigated for the discovery of coefficients by recasting unknown coefficients as GPR kernel hyper-parameters for various time dependent PDEs (Raissi et al., 2018; Raissi and Karniadakis, 2018). As a nonlinear system identification tool, the GPR approach provides a powerful framework to model dynamical systems (Kocijan et al., 2005; Gregorčič and Lightbody, 2008). State calibration with the four dimensional variational data assimilation (4D VAR) (Cordier et al., 2013) and deep learning techniques such as long short-term memory (LSTM) (Wang et al., 2018) have been used for model identification in ROM settings. Convolutional neural networks (CNNs) are constructed to produce hidden physical laws from using the insight of establishing direct connections between filters and finite difference approximations of differential operators (Cai et al., 2012; Dong et al., 2017). This approach has been demonstrated to discover underlying PDEs from learning the filters by minimizing the loss functions (Long et al., 2018, 2019).

1.5 Scope of the Current Work

In this work, we have exploited the use of SR in three different applications, equation discovery, truncation error analysis, and hidden physics discovery. We demonstrate the use of the evolutionary computation algorithm, GEP, and the sparse regression algorithm, STRidge, in the context of the SR approach to discover various physical laws represented by linear and nonlinear PDEs from observing input-response data.

We begin by demonstrating the identification of canonical linear and nonlinear PDEs that are up to fifth order in space. For identifying one particular PDE, we demonstrate the natural feature extraction ability of GEP and the limits in the expressive and predictive power of using a feature library when dealing with STRidge in discovering physical laws. We then demonstrate the discovery of highly nonlinear truncation error terms of the Burgers MDE using both GEP and STRidge. We highlight that the analysis of truncation errors is very important in the implicit large eddy simulation as a way to determine inherent turbulence models. This analysis is usually very tedious and elaborate, and our study provides a clear example of how SR tools are suitable in such research. Following truncation error terms identification, we apply GEP using sparse data to recover hidden source terms represented by complex function compositions for a one-dimensional (1D) advection-diffusion process and a two-dimensional (2D) vortex-merger problem. Furthermore, both GEP and STRidge are used to demonstrate the identification of the eddy viscosity kernel along with its ad-hoc modelling coefficient closing LES equations simulating the 2D decaying turbulence problem. An important result is the ability of the proposed methodology to distill the Smagorinsky model from an array of tailored features in solving the Kraichnan turbulence problem.

1.6 Organization

The rest of the thesis is organized as follows. Chapter II gives a brief description of the GEP and STRidge algorithms. In Chapter III, GEP, and STRidge are tested on identifying different canonical PDEs. Chapter IV deals with the identification of nonlinear truncation terms of the Burgers MDE using both STRidge and GEP. In Chapter V we exploit GEP for identification of hidden source terms in a 1D advection-diffusion process and a 2D vortex-merger problem. In Chapter V we demonstrate recovery of the eddy viscosity kernel and its modelling coefficient by both GEP and STRidge for closing the LES equations simulating the 2D decaying turbulence

problem. Finally, Chapter VI draws our conclusions and highlights some ideas for future extensions of this work.

CHAPTER II

Methodology

We recover various physical models from data using two symbolic regression tools namely, GEP, an evolutionary computing algorithm, and STRidge, which is a deterministic algorithm that draws its influences from compressive sensing and sparse optimization. We take the example of the equation discovery problem that is discussed in Chapter III to elaborate on the methodology of applying GEP and STRidge for recovering various physical models. We restrict the PDEs to be recovered to quadratic nonlinear and up to the fifth order in space. The general nonlinear PDE to be recovered is in the form of,

$$u_t = \mathscr{F}(\sigma, u, u^2, u_x, u_x^2, uu_x, u_{2x}, \ldots, u_{5x}^2), \tag{2.1}$$

where subscripts denote order of partial differentiation and σ is an arbitrary parameter. For example, consider the problem of identifying the viscous Burgers equation as shown below,

$$u_t + uu_x = \nu u_{2x}, \tag{2.2}$$

where $u(x,t) \in \mathbb{R}^{m \times n}$ is the velocity field and ν is the kinematic viscosity. In our study, m is the number of time snapshots and n is the number of spatial locations. The solution field $u(x,t)$ is generally obtained by solving Eq. 2.2 analytically or numerically. The solution field might also be obtained from sensor measurements that can be arranged as shown below,

$$\mathbf{u} = \overbrace{\left[\begin{array}{cccc} u_1(t_1) & u_2(t_1) & \dots & u_n(t_1) \\ u_1(t_2) & u_2(t_2) & \dots & u_n(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ u_1(t_m) & u_2(t_m) & \dots & u_n(t_m) \end{array}\right]}^{\text{spatial locations}} \Bigg\} \text{time snapshots} \tag{2.3}$$

For recovering PDEs, we need to construct a library of basis functions called as feature library that contains higher order derivatives of the solution field $u(x,t)$. Higher order spatial and temporal partial derivative terms can be approximated using any numerical scheme once the recording of the discrete data set given by Eq. 2.3 is available. In our current setup, we use the leapfrog scheme for approximating the temporal derivatives and central difference schemes for spatial derivatives as follows,

$$\left.\begin{aligned} u_t &= \frac{u_j^{p+1} - u_j^{p-1}}{2dt} \\ u_{2t} &= \frac{u_j^{p+1} - 2u_j^p + u_j^{p-1}}{dt^2} \\ u_x &= \frac{u_{j+1}^p - u_{j-1}^p}{2dx} \\ u_{2x} &= \frac{u_{j+1}^p - 2u_j^p + u_{j-1}^p}{dx^2} \\ u_{3x} &= \frac{u_{j+2}^p - 2u_{j+1}^p + 2u_{j-1}^p - u_{j-2}^p}{2dx^3} \\ u_{4x} &= \frac{u_{j+2}^p - 4u_{j+1}^p + 6u_j^p + -4u_{j-1}^p - u_{j-2}^p}{dx^4} \\ u_{5x} &= \frac{u_{j+3}^p - 4u_{j+2}^p + 5u_{j+1}^p - 5u_{j-1}^p + 4u_{j-2}^p - u_{j-3}^p}{2dx^5} \end{aligned}\right\}, \tag{2.4}$$

where temporal and spatial steps are given by dt and dx, respectively. Within the expressions presented in Eq. 2.4, the spatial location is denoted using subscript index j, and the temporal instant using superscript index p.

We note that other approaches such as automatic differentiation or spectral

differentiation for periodic domains can easily be adopted within our study. Both GEP and STRidge take the input library consisting of features (basis functions) that are built using Eq. 2.2 and Eq. 2.3. This core library, used for the equation discovery problem in Chapter III, is shown below,

$$\left.\begin{aligned} \mathbf{V(t)} &= \begin{bmatrix} \mathbf{U}_t \end{bmatrix} \\ \widetilde{\mathbf{\Theta}}(\mathbf{U}) &= \begin{bmatrix} \mathbf{U} & \mathbf{U}_x & \mathbf{U}_{2x} & \mathbf{U}_{3x} & \mathbf{U}_{4x} & \mathbf{U}_{5x} \end{bmatrix} \end{aligned}\right\}. \tag{2.5}$$

The solution $u(x,t)$ and its spatial and temporal derivatives are arranged with size $m \cdot n \times 1$ in each column of Eq. 2.5,. For example, the features (basis functions) $\mathbf{U}$ and $\mathbf{U}_{2x}$ are arranged as follows,

$$\mathbf{U} = \begin{bmatrix} u(x_0,t_0) \\ u(x_0,t_1) \\ | \\ u(x_j,t_p) \\ | \\ u(x_n,t_m) \end{bmatrix}, \quad \mathbf{U}_{2x} = \begin{bmatrix} u_{2x}(x_0,t_0) \\ u_{2x}(x_0,t_1) \\ | \\ u_{2x}(x_j,t_p) \\ | \\ u_{2x}(x_n,t_m) \end{bmatrix}, \tag{2.6}$$

where subscript j denotes the spatial location and subscript p denotes the time snapshot. The features (basis functions) in the core library $\widetilde{\mathbf{\Theta}}(\mathbf{U})$ is expanded to include interacting features limited to quadratic nonlinearity and also a constant term. The final expanded library is given as,

$$\mathbf{\Theta}(\mathbf{U}) = \begin{bmatrix} \mathbf{1} & \mathbf{U} & \mathbf{U}^2 & \mathbf{U}_x & \mathbf{U}\mathbf{U}_x & \mathbf{U}_x^2 & \dots & \mathbf{U}_{5x}^2 \end{bmatrix}, \tag{2.7}$$

where the size of the library is $\mathbf{\Theta}(\mathbf{U}) \in \mathbb{R}^{m \cdot n \times N_\beta}$ and N_β is number of features (basis functions) i.e., $N_\beta = 28$ for our setup. For example, if we have 501 spatial points and 101 time snapshots with 28 bases, then $\mathbf{\Theta}(\mathbf{U})$ (Eq. 2.7) contains 501×101 rows and

28 columns.

Note that the core feature library $\widetilde{\mathbf{\Theta}}(\mathbf{U})$ in Eq. 2.5 is given as an input to GEP to recover PDEs and the algorithm extracts higher degree nonlinear interactions of core features in $\widetilde{\mathbf{\Theta}}(\mathbf{U})$ automatically. However, for sparse optimization techniques such as STRidge, explicit input of all possible combinations of core features in Eq. 2.5 are required. Therefore, $\mathbf{\Theta}(\mathbf{U})$ in Eq. 2.7 forms the input to STRidge algorithm for equation identification. This forms the fundamental difference in terms of feature building for both algorithms. The following Section 2.1 gives a brief introduction to GEP and its specific hyper-parameters that control the efficacy of the algorithm in identifying physical models from observing data. Furthermore, the Section 2.2 describes how to form linear system representations in terms of $\mathbf{V}(\mathbf{t})$ and $\mathbf{\Theta}(\mathbf{U})$ and briefly describe STRidge optimization approach to identifying sparse features and thereby building parsimonious models using spatio-temporal data.

2.1 Gene Expression Programming

Gene expression programming (GEP) (Ferreira, 2001, 2002) is a genotype-phenotype evolutionary optimization algorithm which takes advantage of simple chromosome representation of genetic algorithm (GA) (Mitchell, 1998) and the free expansion of complex chromosomes of genetic programming (GP) (Koza, 1992). As in most evolutionary algorithms, this technique also starts with generating initial random populations, iteratively selecting candidate solutions according to a fitness function, and improving candidate solutions by modifying through genetic variations using one or more genetic operators. The main difference between GP and GEP is how both techniques define the nature of their individuals. In GP, the individuals are nonlinear entities of different sizes and shapes represented as parse trees and in GEP the individuals are encoded as linear strings of fixed length called genome and chromosome, similar to GA representation of individual and later expressed as nonlinear entities of

different size and shape called phenotype or expression trees (ET). GEP is used for a very broad range of applications, but here it is introduced as a symbolic regression tool to extract constraint free solutions from input-response data.

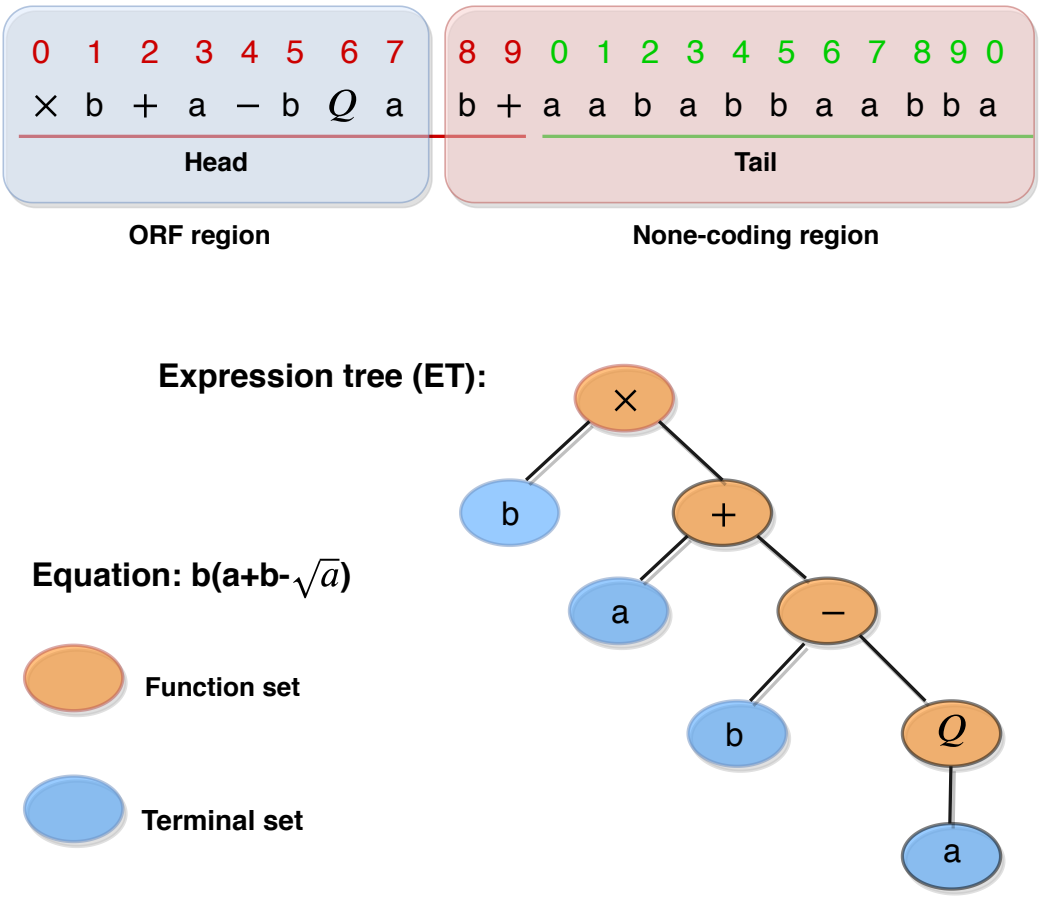

Figure 2.1: ET of a gene/chromosome with its structure in GEP. Q represents the square root operator.

The arrangement of a typical gene/chromosome in GEP is shown in Fig. 2.1. The GEP gene is composed of head and tail regions as illustrated in Fig. 2.1. The head of a gene consists of both symbolic terms from functions (elements from a function set F) and terminals (elements from a terminal set T) whereas the tail consists of only terminals. The function set F may contain arithmetic mathematical operators (e.g., $+, \times, -, /$), nonlinear functions (e.g., sin, cos, tan, arctan, sqrt, exp), or Boolean operators (e.g., Not , Nor , Or , And) and the terminal set contains the symbolic variables. The gene always starts with a randomly generated mathematical operator

from the function set F. The head length is one of the important hyper-parameters of GEP, and it is determined using trial and error as there is no definite method to assign it. Once the head length is determined, the size of the tail is computed as a function of the head length and the maximum arity of a mathematical operator in the function set F (Ferreira, 2006). It can be calculated by the following equation,

$$\text{tail length} = \text{head length} \times (a_{max} - 1) + 1, \tag{2.8}$$

where a_{max} is the maximum argument of a function in F. The single gene can be extended to multigenic chromosomes where individual genes are linked using a linking function (eg., $+, \times, /, -$). The general rule of thumb is to have a larger head and higher number of genes when dealing with complex problems (Ferreira, 2006).

The structural organization of the GEP gene is arranged in terms of open reading frames (ORFs) inspired from biology where the coding sequence of a gene equivalent to an ORF begins with a start codon, continue with an amino acid codon and ends with a termination codon. In contrast to a gene in biology, the start site is always the first position of a gene in GEP, but the termination point does not always coincide with the last position of a gene. These regions of the gene are termed non coding regions downstream of the termination point. Only the ORF region is expressed in the ET and can be clearly seen in Fig. 2.1.

Even though the none-coding regions in GEP genes do not participate in final solution, the power of GEP evolvability lies in this region. The syntactically correct genes in GEP evolve after modification through diverse genetic operators due to this region chromosome. This is the paramount difference between GEP and GP implementations where in latter, many syntactically invalid individuals are produced and need to be discarded while evolving the solutions and additional special constraint are imposed on the depth/complexity of candidate solution to be evolved to avoid bloating problem (Duriez et al., 2015).

Fig. 2.2 displays the typical flowchart of the GEP algorithm. The process is described briefly below,

1. The optimization procedure starts with a random generation of chromosomes built upon combinations of functions and terminals. The size of the random population is a hyper-parameter and the larger the population size, better the probability of finding the best candidate solution.

2. After the population is generated, the chromosomes are expressed as ETs, which is converted to a numerical expression. This expression is then evaluated using a fitness function. In our setup, we employ the mean squared error between the best predicted model f^* and the true model f as the fitness function given by,

$$MSE = \frac{1}{N}\sum_{l=1}^{N}\left(f^*_{(lk)} - f_{(l)}\right)^2, \tag{2.9}$$

   where f^*_{lk} is the value predicted by the chromosome k for the fitness case l (out of N samples cases) and f_l is the true or measurement value for the l^{th} fitness case.

3. The termination criteria is checked after all fitness evaluations, to continue evolving or to save the best fitness chromosome as our final predicted model. In our current setup, we terminate after a specified number of generations.

4. The evolvability/reproduction of chromosome through genetic operators which is the core part of the GEP evolutionary algorithm executes if termination criteria is not met. Before the genetic operations on chromosome begins, the best chromosome according to fitness function is cloned to the next generations using a selection method. Popular selection methods include tournament selection with elitism and roulette-wheel selection with elitism. In our current setup, we use tournament selection with elitism.

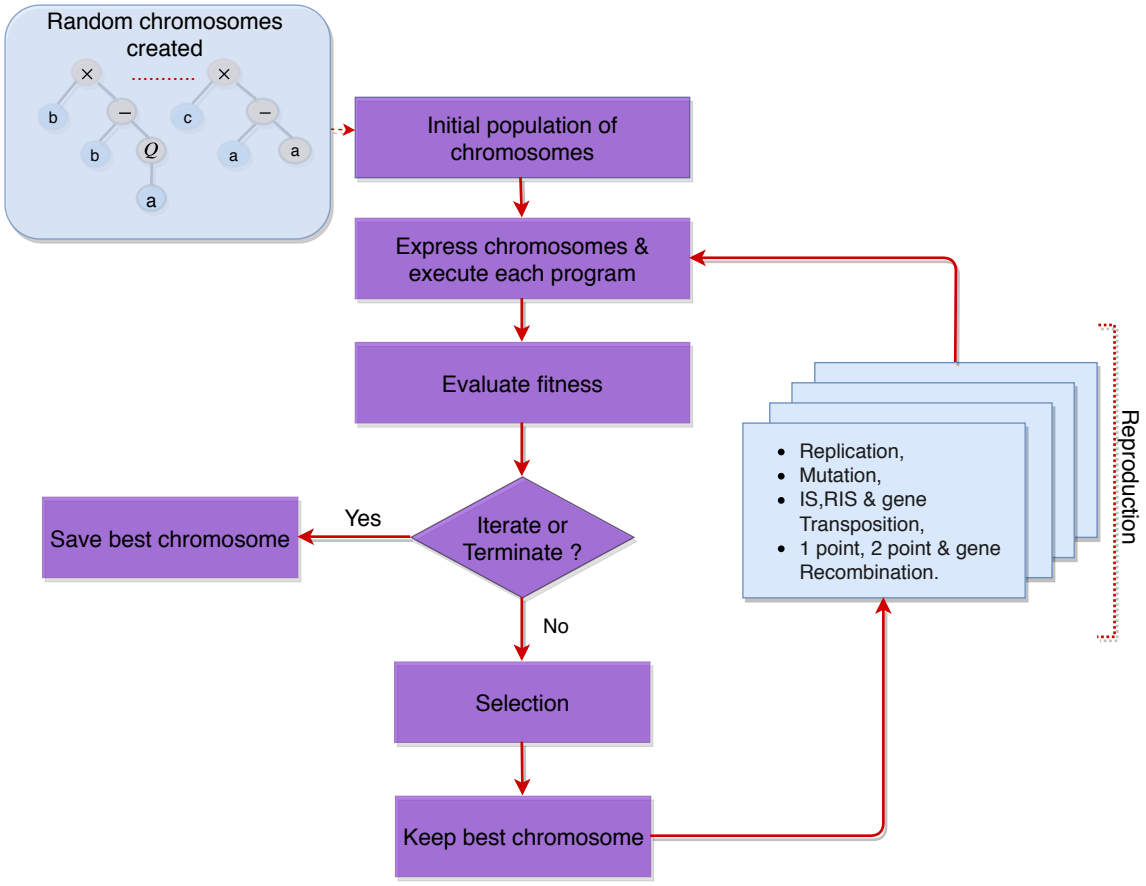

Figure 2.2: Flowchart of the gene expression programming.

5. The four genetic operators that introduce variation in populations are mutation, inversion, transposition, and recombination. The GEP transposition operator is applied to the elements of the chromosome in three ways: insertion sequence (IS), root insertion sequence (RIS) and gene insertion sequence and similarly three kinds of recombination are applied namely one point, two point, and gene recombination.

6. The process is continued up to termination criteria is met, which is the number of generations in our current setup.

Numerical constants occur in most mathematical models and, therefore, it is important to any symbolic regression tools to effectively integrate floating point constants in their optimization search. GP (Koza, 1992) handles numerical constants

Table 2.1: GEP hyper-parameters for various genetic operators selected for all the test cases in this study.

Hyper-parameters	**Value**
Selection	Tournament selection
Mutation rate	0.05
Inversion	0.1
IS transposition rate	0.1
RIS transposition rate	0.1
Gene transposition rate	0.1
One point recombination	0.3
Two point recombination	0.2
Gene recombination	0.1
Dc specific mutation rate	0.05
Dc specific inversion rate	0.1
Dc specific transposition rate	0.1
Random constant mutation rate	0.02

by introducing random numerical constants in a specified range to its parse trees. The random constants are moved around the parse trees using the crossover operator. GEP handles the creation of random numerical constants (RNCs) by using an extra terminal '?' and a separate domain Dc composed of symbols chosen to represent random numerical constants (Ferreira, 2006). This Dc specific domain starts from the end of the tail of the gene.

For each gene, RNCs are generated during the creation of a random initial population and kept in an array. To maintain the genetic variations in the pool of RNCs, additional genetic operators are introduced to take effect on Dc specific regions. Hence in addition to the usual genetic operators such as mutation, inversion, transposition and recombination, the GEP-RNC algorithm has Dc specific inversion, transposition, and random constant mutation operators. Hence, with these modifications to the algorithm, an appropriate diversity of random constants can be generated and evolved through operations of genetic operators. The values for each genetic operator selected for this study are listed in Table 2.1. These values are selected from various examples given by Ferreira (Ferreira, 2006) combined with the trial and error approach. Additionally, to simplify our study, we use the same parameters for all the test cases even though they may not be the best values for the test case under investigation.

Once decent values of genetic operators that can explore the search space are selected, the size of the head length, population, and the number of genes form the most important hyper-parameters for GEP. Generally, larger head length and a greater number of genes are selected for identifying complex expressions. Larger population size helps in a diverse set of initial candidates which may help GEP in finding the best chromosome in less number of generations. However, computational overhead increases with an increase in the size of the population. Furthermore, the best chromosome can be identified in fewer generations with the right selection of the linking function between genes. GEP algorithm inherently performs poor

in predicting the numerical constants that are ubiquitous in physical laws. Hence, the GEP-RNC algorithm is used where a range of random constants are predefined to help GEP to find numerical constants. This also becomes important in GEP identifying the underlying expression in fewer generations. Finally, we note that due to the heuristic nature of evolutionary algorithms, any other combinations of hyper-parameters might work perfectly in identifying the symbolic expressions. In this study, we use geppy (Shuhua, 2019), an open source library for symbolic regression using GEP, which is built as an extension to distributed evolutionary algorithms in Python (DEAP) package (Fortin et al., 2012). All codes used in this study are made available on Github (https://github.com/sayin/SR).

2.2 Sequential Threshold Ridge Regression

Compressive sensing/sparse optimization (Baraniuk, 2007; Candes and Wakin, 2008) has been exploited for sparse feature selection from a large library of potential candidate features and recovering dynamical systems represented by ODEs and PDEs (Brunton et al., 2016; Rudy et al., 2017; Mangan et al., 2017) in a highly efficient computational manner. In our setup, we use this STRidge (Rudy et al., 2017) algorithm to recover various hidden physical models from observed data. In continuation with the Chapter II where we define feature library $\mathbf{\Theta}(\mathbf{U})$ and target/output data $\mathbf{V}(\mathbf{t})$, this section briefly explains the formation of an overdetermined linear system for STRidge optimization to identify various physical models from data.

The Burgers PDE given in Eq. 2.2 or any other PDE under consideration can be written in the form of linear system representation in terms of $\mathbf{\Theta}(\mathbf{U})$ and $\mathbf{V}(\mathbf{t})$,

$$\mathbf{V}(t) = \mathbf{\Theta}(\mathbf{U}) \cdot \boldsymbol{\beta}, \tag{2.10}$$

where $\boldsymbol{\beta} = [\beta_1, \beta_2, \ldots, \beta_{N_\beta}]$ is coefficient vector of size $\mathbb{R}^{N_\beta}$ where N_β is number of features (basis functions) in library $\mathbf{\Theta}(\mathbf{U})$. Note that $\mathbf{\Theta}(\mathbf{U})$ is an over-complete library

(the number of measurements is greater than the number of features) and having rich feature (column) space to represent the dynamics under consideration. Thus, we form an overdetermined linear system in Eq. 2.10. The goal of STRidge is to find a sparse coefficient vector $\boldsymbol{\beta}$ that only consists of active features, which best represent the dynamics. The rest of the features are hard thresholded to zero. For example, in the Burgers equation given by Eq. 2.2, STRidge ideally has to find the coefficient vector $\boldsymbol{\beta}$ that corresponds to the features uu_x and u_{2x} and simultaneously it should set all other feature coefficients to zero.

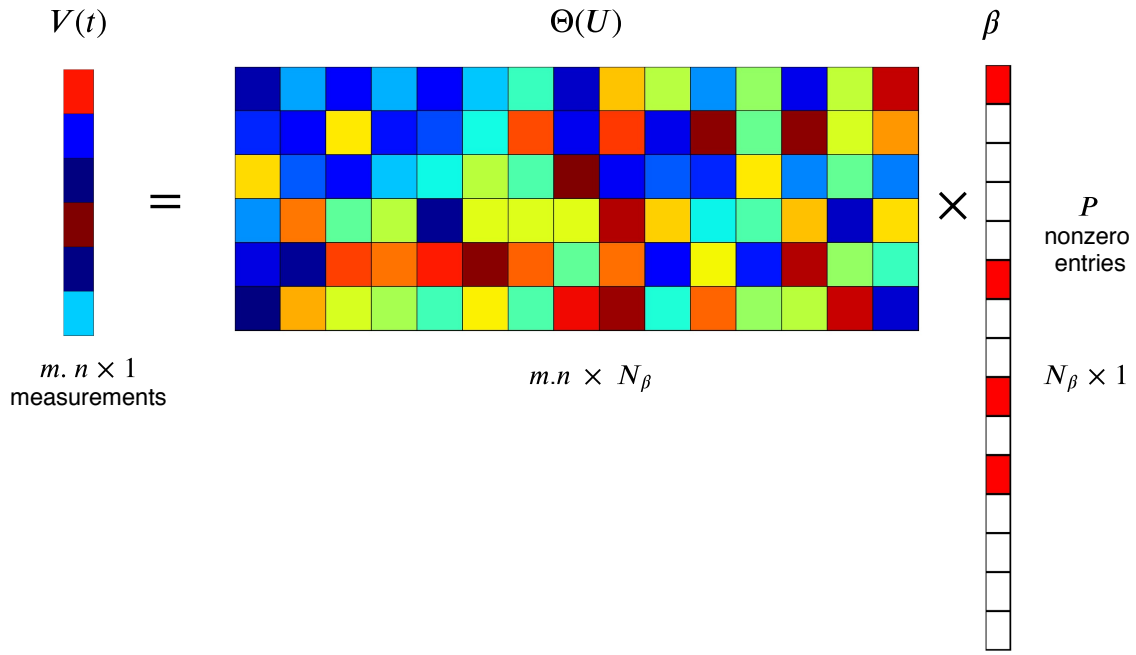

Figure 2.3: Structure of compressive matrices with sparse non zero entries in coefficient vector $\boldsymbol{\beta}$. Red boxes in $\boldsymbol{\beta}$ vector correspond to active feature coefficients and all other coefficients being set to zero.

The linear system defined in Eq. 2.10 can be solved for β using the ordinary least squares (OLS) problem. But OLS minimization tries to form a functional relationship with all the features in $\boldsymbol{\Theta}(\mathbf{U})$ resulting in all non zero values in the coefficient vector $\boldsymbol{\beta}$. Thus solving Eq. 2.10 using OLS infers radically complex functional form to represent the underlying PDE and generally results in overfitted models. Regularized least square minimization can be applied to constraint the coefficients and avoid

overfitting. Hence regularized LS optimization is preferred to identify the sparse features (basis functions) along with their coefficient estimation. Typical estimation of sparse coefficient vector with P non zero entries in $\boldsymbol{\beta}$ is shown in Fig. 2.3. General sparse regression objective function to approximate the solution of the coefficient vector $\boldsymbol{\beta}$ is given by,

$$\beta^* = \arg\min_\beta ||\boldsymbol{\Theta} \cdot \beta - \mathbf{V}(\mathbf{t})||_2^2 + \lambda||\beta||_0, \tag{2.11}$$

where λ is regularizing weight and $||\beta||_0$ corresponds to L_0 penalty which makes the problem np-hard. Hence to arrive at convex optimization problem of Eq. 2.12, L_1 and L_2 penalty is generally used to approximate the solution of the coefficient vector $\boldsymbol{\beta}$.

The addition of L_1 penalty to LS objective function which corresponds to maximum a posteriori estimate (MAP) of Laplacian prior and termed as least absolute shrinkage and selection operator (LASSO) in compressive sensing. It is defined by,

$$\beta^* = \arg\min_\beta ||\boldsymbol{\Theta} \cdot \beta - \mathbf{V}(\mathbf{t})||_2^2 + \lambda||\beta||_1. \tag{2.12}$$

However, the performance of LASSO deteriorates when the feature space is correlated (Rudy et al., 2017). The sequential threshold least squares (STLS) algorithm was proposed to identify dynamical systems represented by ODEs (Brunton et al., 2016). In STLS, a hard threshold is performed on least square estimates of regression coefficients and hard threshold is recursively performed on remaining non zero coefficients. However, the efficacy of STLS reduces when dealing with the identification of systems containing multiple correlated columns in $\boldsymbol{\Theta}$. Hence L_2 regularized least squares termed as ridge regression (Murphy, 2012), which corresponds to the maximum a posteriori estimate using a Gaussian prior, is proposed to handle the identification

of PDEs. Ridge regression is defined by,

$$\begin{aligned}\beta^* &= \arg\min_\beta ||\boldsymbol{\Theta} \cdot \beta - \mathbf{V}(\mathbf{t})||_2^2 + \lambda||\beta||_2, \\ &= (\boldsymbol{\Theta}^T\boldsymbol{\Theta} + \lambda^T I)^{-1}\boldsymbol{\Theta}^T\mathbf{V}(\mathbf{t}). \end{aligned} \tag{2.13}$$

Ridge regression is substituted for ordinary least squares in STLS and the resulting algorithm as sequential threshold ridge regression (STRidge) (Rudy et al., 2017). The STRidge framework (Rudy et al., 2017) is illustrated in Algorithm 1 for the sake of completeness. Note that, if $\lambda = 0$, STRidge becomes STLS procedure. For more elaborate details on updating tolerance (tol) to perform hard thresholding in Algorithm 1, readers are encouraged to refer supplementary document of Rudy et al. (2017).

Algorithm 1: STRidge($\boldsymbol{\Theta}$, $\mathbf{V}(\mathbf{t})$, λ, tol, iters) (Rudy et al., 2017)

Input: $\boldsymbol{\Theta}, \mathbf{V}(\mathbf{t}), \lambda, tol$, iters
Output: β^*
$\beta^* = \arg\min_\beta ||\boldsymbol{\Theta} \cdot \beta - \mathbf{V}(\mathbf{t})||_2^2 + \lambda||\beta||_2^2$
large $= \{p : |\beta_p^*| \geq tol\}$
$\beta^*[$ large$] = 0$
$\beta^*[\text{large}] = \text{STRidge}(\boldsymbol{\Theta}[:, \text{large}], \mathbf{V}(\mathbf{t}), \lambda, tol, \text{iters} - 1)$
return β^*

We use the framework provided by Rudy et al. (2017) in our current study. The hyper-parameters in STRidge include the regularization weight λ and tolerance level tol which are to be tuned to identify appropriate physical models. In the present study, the sensitivity of feature coefficients for various values of λ and the final value of λ where the best model is identified is showed. The following sections deal with various numerical experiments to test the GEP and STRidge frameworks.

CHAPTER III

PDE Discovery

Table 3.1: Summary of canonical PDEs selected for recovery.

PDE	**Exact solution**	**Constant parameters**	**Discretization** n **(spatial)** m **(temporal)**
Wave eq. $u_t = -au_x$	$u(t,x) = \sin(2\pi(x-at))$	$a = 1.0$	$x \in [0,1]$ $(n = 101)$, $t \in [0,1]$ $(m = 101)$
Heat eq. $u_t = \alpha u_{2x}$	$u(t,x) = \sin(x)\exp(-\alpha t)$	$\alpha = 1.0$	$x \in [-\pi, \pi]$ $(n = 201)$, $t \in [0,1]]$ $(m = 101)$
Burgers eq. (i) $u_t = -uu_x + \nu u_{2x}$	$u(t,x) = \frac{x}{(t+1)\left(1+(\sqrt{t+1})\exp(\frac{1}{16\nu}\frac{4x^2-t-1}{t+1})\right)}$	$\nu = 0.01$	$x \in [0,1]$ $(n = 101)$, $t \in [0,1]$ $(m = 101)$
Burgers eq. (ii) $u_t = -uu_x + \nu u_{2x}$	$u(t,x) = \frac{2\nu\pi\exp(-\pi^2\nu t)\sin(\pi x)}{a+\exp(-\pi^2\nu t)\cos(\pi x)}$	$\nu = 0.01$, $a = 5/4$	$x \in [0,1]$ $(n = 101)$, $t \in [0,100]$ $(m = 101)$
Korteweg-de Vries eq. $u_t = -\alpha uu_x - \beta u_{3x}$	$u(t,x) = 12\left(\frac{4\cosh(2x-8t)+\cosh(4x-64t)+3}{(3\cosh(x-28t)+\cosh(3x-36t))^2}\right)$	$\alpha = 6.0$, $\beta = 1.0$	$x \in [-10,10]$ $(n = 501)$, $t \in [0,1]$ $(m = 201)$
Kawahara eq. $u_t = -uu_x - \alpha u_{3x} - \beta u_{5x}$	$u(t,x) = \frac{105}{169}\mathrm{sech}\left(\frac{1}{2\sqrt{13}}(x-at)\right)^4$	$\alpha = 1.0$, $\beta = 1.0$, $a = 36/169$	$x \in [-20,20]$ $(n = 401)$, $t \in [0,1]$ $(m = 101)$
Newell-Whitehead-Segel eq. $u_t = \kappa u_{2x} + \alpha u - \beta u^q$	$u(t,x) = \frac{1}{\left(1+\exp(\frac{x}{\sqrt{6}} - \frac{5t}{6})\right)^2}$	$\kappa = 1.0$, $\alpha = 1.0$, $\beta = 1.0$, $q = 2$	$x \in [-40,40]$ $(n = 401)$, $t \in [0,2]$ $(m = 201)$
Sine-Gordon eq. $u_{2t} = \kappa u_{2x} - \alpha\sin(u)$	$u(t,x) = 4\tan^{-1}(\mathrm{sech}(x)t)$	$\kappa = 1.0$, $\alpha = 1.0$	$x \in [-2,2]$ $(n = 401)$, $t \in [0,1]$ $(m = 101)$

Partial differential equations (PDEs) play a prominent role in all branches of science and engineering. They are generally derived from conservation laws, sound physical arguments, and empirical heuristic from an insightful researcher. The recent explosion of machine learning algorithms provides new ways to identify hidden physical laws represented by PDEs using only data. In this section, we demonstrate the identification of various linear and nonlinear canonical PDEs using the GEP and STRidge algorithms from using data alone. Analytical solutions of PDEs are used to form the data.

Table 3.2: GEP hyper-parameters selected for identification of various PDEs.

Hyper-parameters	**Wave eq.**	**Heat eq.**	**Burgers eq. (i)**	**Burgers eq. (ii)**
Head length	2	2	4	2
Number of genes	1	2	1	2
Population size	25	25	20	50
Generations	100	100	500	500
Length of RNC array	10	10	30	5
Random constant minimum	-10	-1	-1	-1
Random constant maximum	10	1	1	1

Table 3.3: GEP hyper-parameters selected for identification of various PDEs.

Hyper-parameters	**KdV eq.**	**Kawahara eq.**	**NWS eq.**	**Sine-Gordon eq.**
Head length	6	2	5	3
Number of genes	5	1	3	2
Population size	20	20	30	100
Generations	500	100	100	500
Length of RNC array	30	5	25	20
Random constant minimum	1	-1	-10	-10
Random constant maximum	10	1	10	10

Table 3.1 summarizes various PDEs along with their analytical solutions $u(t, x)$ and domain discretization. Building a feature library and corresponding response data to identify PDEs is discussed in detail in Chapter II.

Table 3.4: GEP functional and terminal set used for equation discovery. '?' is a random constant.

Parameter	**Value**
Function set	$+, -, \times, /$, sin, cos
Terminal set	$\widetilde{\mathbf{\Theta}}(\mathbf{U})$, ?
Linking function	$+$

We reiterate the methodology for PDE identification in Chapter II. The analytical solution $u(t, x)$ is solved at discrete spatial and temporal locations resulting from the discretization of space and time domains as given in Table 3.1. The discrete analytical solution is used as input data for calculating higher order spatial and temporal data using the finite difference approximations listed in Eq. 2.4. Furthermore, the feature library is built using discrete solution $u(t, x)$ and higher order derivative which is discussed in Chapter II. As GEP is a natural feature extractor, core feature library $\widetilde{\mathbf{\Theta}}(\mathbf{U})$ given in Eq. 2.5 is enough to form input data, i.e., GEP terminal set. Table 3.4 shows the function set and terminal set used for equation identification and Table 2.1 lists the hyper-parameter values for various genetic operators. However, extended core feature library $\mathbf{\Theta}(\mathbf{U})$ which contains a higher degree interactions of features is used as input for STRidge as the expressive power of STRidge depends on exhaustive combinations of features in the input library. The temporal derivative of $u(t, x)$ is target or response data $\mathbf{V}(\mathbf{t})$ given in Eq. 2.5 for both GEP and STRidge.

3.1 Wave Equation

Our first test case is the wave equation which is a first order linear PDE. The PDE and its analytical solution are listed in Table 3.1. We choose the constant wave speed

$a = 1.0$ for propagation of the solution $u(t, x)$. Fig. 3.1 shows the analytical solution $u(t, x)$ of the wave equation. The GEP hyper-parameters used for identification of the wave equation are listed in Table 3.2. We use a smaller head length and a single gene for simple cases like a linear wave PDE. We note that any other combinations of hyper-parameters may identify the underlying PDE. Fig. 3.2 illustrates the identified PDE in the ET form. When the ET form is simplified, we can show that the resulting equation is the correct wave PDE, identified with its wave propagation speed parameter a.

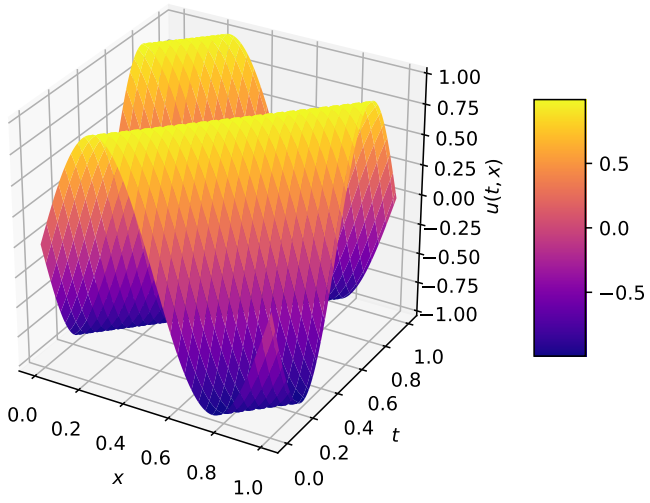

Figure 3.1: Analytical solution of the wave equation.

The regularization weight (λ) in STRidge is swept across various values as shown in Fig. 3.3. The yellow line in Fig. 3.3 represents the value of λ at which the best identified PDE is selected. Note that in this simple case STRidge was able to find the wave equation for almost all the values of λ's that are selected. Table 3.5 shows the wave PDE recovered by both GEP and STRidge.

Table 3.5: Wave equation identified by GEP and STRidge.

	Recovered	**Test error**
True	$u_t = -1.00\ u_x$	
GEP	$u_t = -1.00\ u_x$	1.72×10^{-28}
STRidge	$u_t = -1.00\ u_x$	9.01×10^{-29}

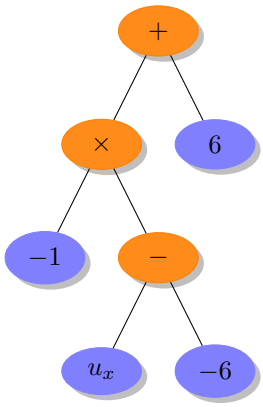

Figure 3.2: Wave equation in terms of ET identified by GEP.

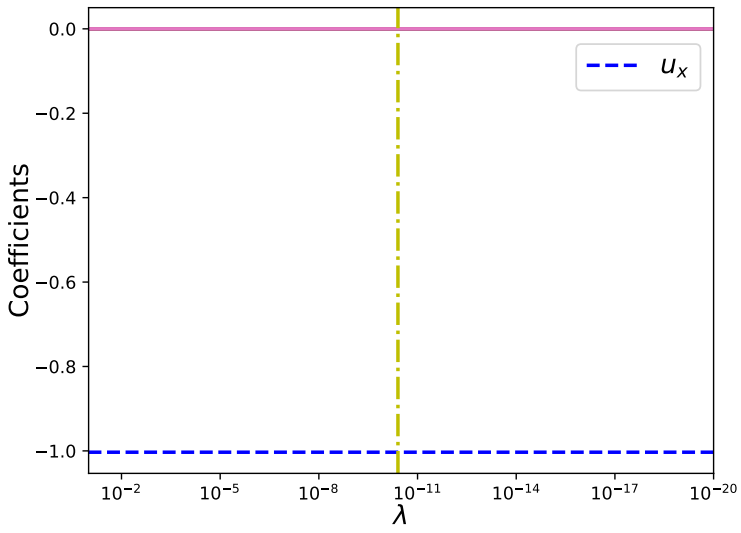

Figure 3.3: STRidge coefficients as a function of regularization parameter λ for the wave equation.

3.2 Heat Equation

We use the heat equation which is a second order linear PDE to test both SR approaches. The PDE and its analytical solution is listed in Table 3.1. The physical parameter $\alpha = 1.0$ may represent thermal conductivity. Fig. 3.4 displays the analytical solution $u(t, x)$ of the heat equation. Table 3.2 lists the GEP hyper-parameters used for identification of the heat equation. Fig. 3.5 shows the identified PDE in the form of an ET. When the ET form is simplified, we can show that the resulting model is the heat equation identified with its coefficient α.

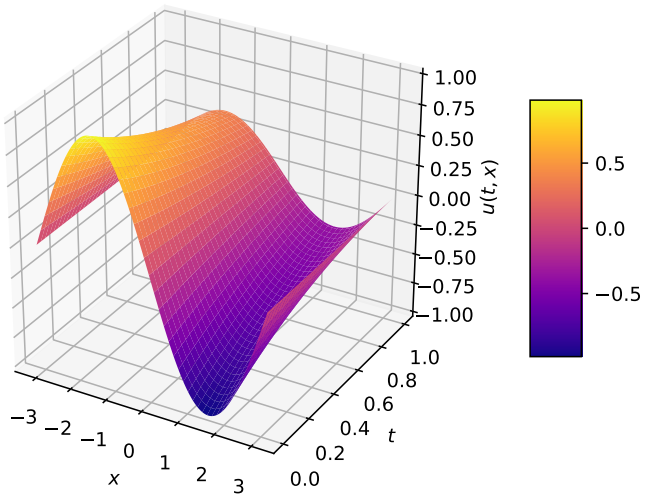

Figure 3.4: Analytical solution of the heat equation.

The regularization weight (λ) in STRidge is swept across various values as shown Fig. 3.6. The yellow line in Fig. 3.6 represents the value of λ selected at which STRidge finds the heat equation accurately. Note that STRidge was able to find the heat equation for low values of the regularization weight λ as shown in Fig. 3.6. Table 3.6 shows the heat equation recovered by both GEP and STRidge. STRidge was able to find a more accurate coefficient (α) value than GEP. Furthermore, a small constant value is also identified along with the heat equation by GEP.

Table 3.6: Heat equation identified by GEP and STRidge.

	Recovered	**Test error**
True	$u_t = 1.00\ u_{2x}$	
GEP	$u_t = 0.99\ u_{2x} - 5.33 \times 10^{-15}$	5.55×10^{-24}
STRidge	$u_t = 1.00\ u_{2x}$	4.09×10^{-30}

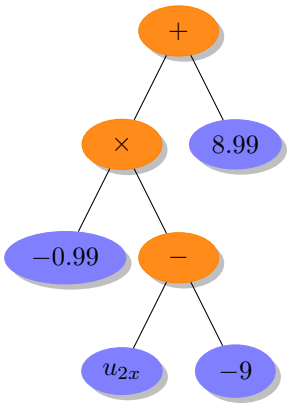

Figure 3.5: Heat equation in terms of ET identified by GEP.

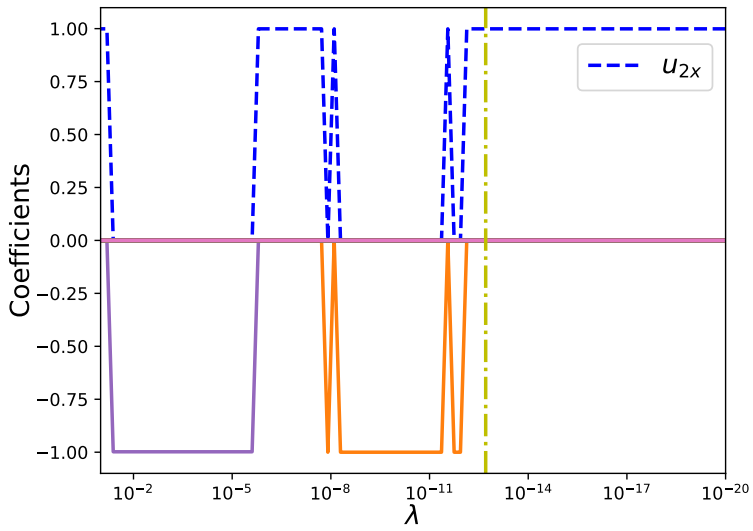

Figure 3.6: STRidge coefficients as a function of regularization parameter λ for the heat equation.

3.3 Burgers Equation (i)

Burgers equation is a fundamental nonlinear PDE occurring in various areas such as fluid mechanics, nonlinear acoustics, gas dynamics and traffic flow Bateman (1915); Whitham (2011). The interest in the Burgers equation arises due to the non linear term uu_x and presents a challenge to both GEP and STRidge in the identification of its PDE using data. The form of the Burgers PDE and its analytical solutionMaleewong and Sirisup (2011) is listed in Table 3.1. The physical parameter $\nu = 0.01$ can be considered as the kinematic viscosity in fluid flows. Fig. 3.7 shows the analytical solution $u(t, x)$ of the Burgers equation. Table 3.2 shows the GEP hyper-parameters used for identification of the Burgers equation. Fig. 3.8 shows the identified PDE in the form of the ET. When ET form is simplified, we can show that the resulting model is the Burgers equation identified along with the coefficient of the nonlinear term and the kinematic viscosity. GEP uses more generations for identifying the Burgers PDE due to its nonlinear behavior along with the identification of feature interaction term uu_x.

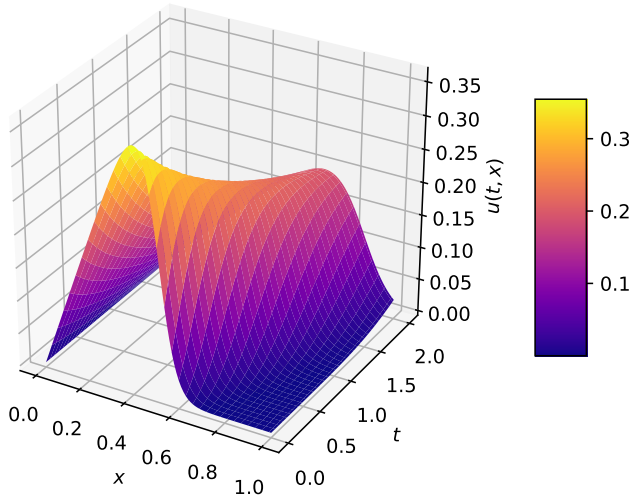

Figure 3.7: Analytical solution of the Burgers equation (i).

The regularization weight (λ) in STRidge is swept across various values as shown in Fig. 3.9. The yellow line in Fig. 3.9 represents the value of λ at which the best identified PDE is selected. Note that the STRidge algorithm was able to find the Burgers equation at multiple values of regularization weights λ. Table 3.7 shows the Burgers PDE recovered by both GEP and STRidge. There is an additional constant coefficient term recovered by GEP. Furthermore, the recovery of the nonlinear term using a limited set of input features shows the usefulness of GEP.

Table 3.7: Burgers equation (i) identified by GEP and STRidge.

	Recovered	**Test error**
True	$u_t = -uu_x + 0.01\ u_{2x}$	
GEP	$u_t = -uu_x + 0.01\ u_{2x} - 1.23 \times 10^{-5}$	6.10×10^{-08}
STRidge	$u_t = -uu_x + 0.01\ u_{2x}$	5.19×10^{-08}

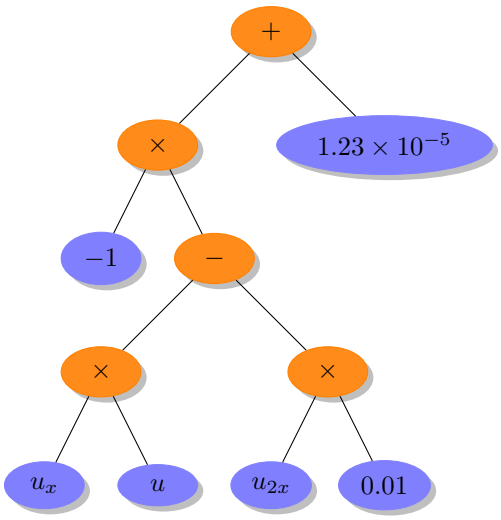

Figure 3.8: Burgers equation (i) in terms of ET identified by GEP.

3.4 Burgers Equation (ii)

Burgers PDE with a different analytical solution is used to test the effectiveness of GEP and STRidge as the input data is changed but represented by the same

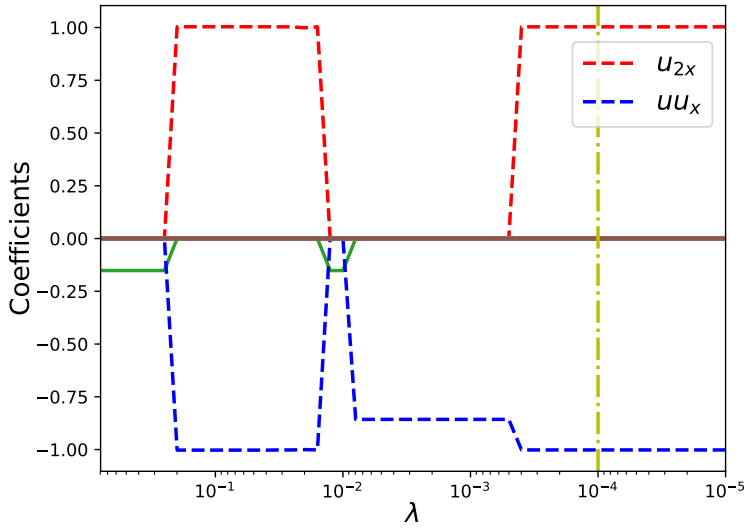

Figure 3.9: STRidge coefficients as a function of regularization parameter λ for the Burgers equation (i).

physical law. The analytical solution of the Burgers equation (ii) is listed in Table 3.1. The physical parameter $\nu = 0.01$ is used to generate the data. Fig. 3.10 shows the alternate analytical solution $u(t, x)$ of the Burgers equation. Table 3.2 shows the GEP hyper-parameters used for identification of the Burgers equation (ii). Fig. 3.11 shows the identified PDE in the form of ET. When ET form is simplified, we can show that the resulting model is the Burgers equation identified along with the coefficient of nonlinear term and kinematic viscosity. With an alternate solution, GEP uses a larger head length, more genes, and a larger population for identifying the same Burgers PDE.

The regularization weight (λ) in STRidge is swept across various values as shown Fig. 3.12. The yellow line in Fig. 3.12 represents the value of λ at which the best identified PDE is selected. Note that STRidge was able to find the Burgers equation at various values of regularization weight λ. Table 3.8 shows the Burgers PDE recovered by both GEP and STRidge.

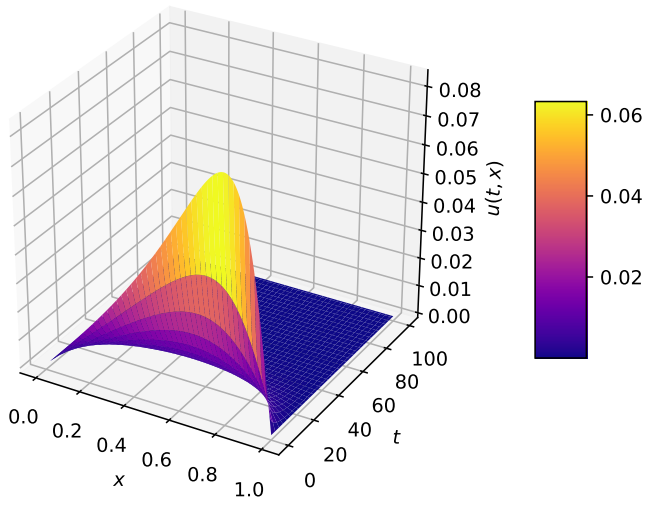

Figure 3.10: Analytical solution of the Burgers equation (ii).

Table 3.8: Burgers equation (ii) identified by GEP and STRidge.

	Recovered	**Test error**
True	$u_t = -1.00\ uu_x + 0.01\ u_{2x}$	
GEP	$u_t = -1.01\ uu_x + 0.01\ u_{2x} - 3.33 \times 10^{-6}$	1.94×10^{-09}
STRidge	$u_t = -0.99\ uu_x + 0.01\ u_{2x}$	1.85×10^{-08}

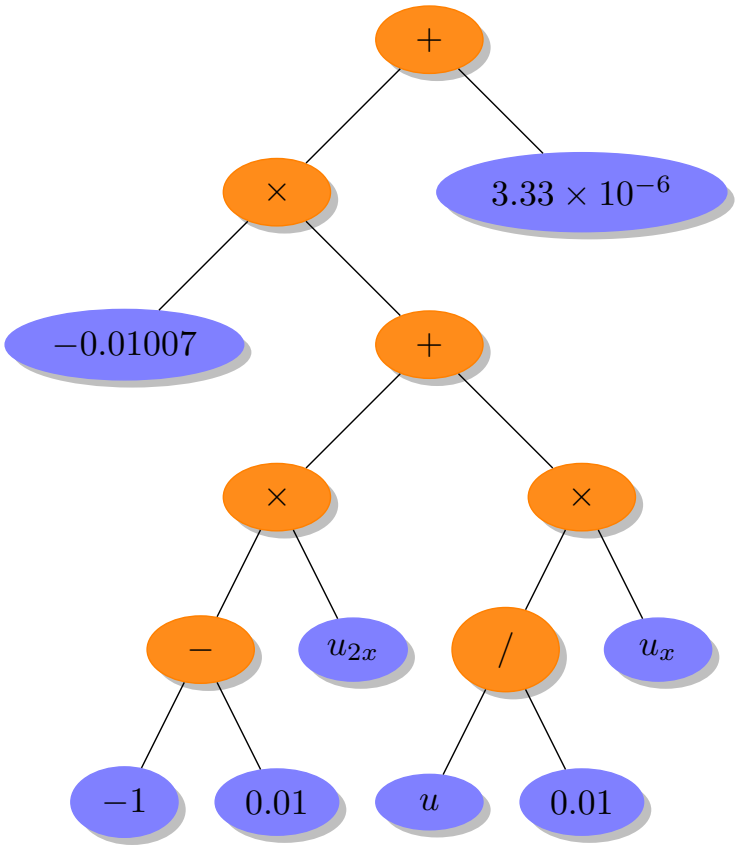

Figure 3.11: Burgers equation (ii) in terms of ET identified by GEP.

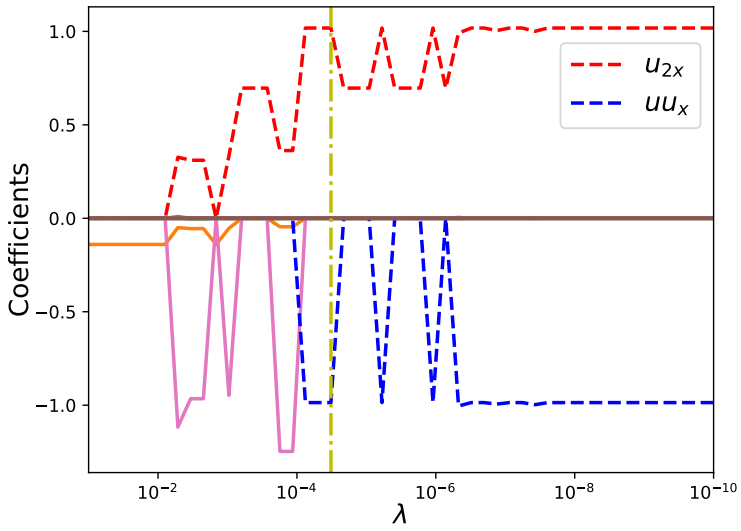

Figure 3.12: STRidge coefficients as a function of regularization parameter λ for the Burgers equation (ii).

3.5 Korteweg-de Vries (KdV) Equation

Korteweg and de Vries derived the KdV equation to model Russell's phenomenon of solitons Korteweg and de Vries (1895); Wazzan (2009). The KdV equation also appears when modelling the behavior of magneto-hydrodynamic waves in warm plasma's, acoustic waves in an inharmonic crystal and ion-acoustic waves Ozis and Ozer (2006). Many different forms of the KdV equation available in the literature but we use the form given in Table 3.1. Fig. 3.13 shows the analytical solution $u(t, x)$ of the KdV equationLamb Jr (1980). It can be seen that this analytical solution refers to two solutions colliding together which forms good test case for SR techniques like GEP and STRidge. Table 3.3 shows the GEP hyper-parameters used for identification of the KdV equation. Due to the higher nonlinear dynamics represented by higher order PDE, GEP requires large head length and genes compared to other test cases in equation discovery. Fig. 3.14 shows the identified PDE in the form of the ET. When ET form is simplified, we can observe that the resulting model is the KdV equation identified along with its coefficients.

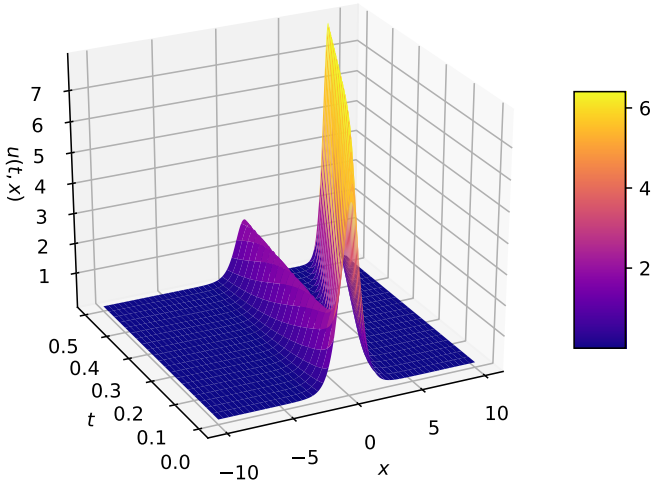

Figure 3.13: Analytical solution of the KdV equation.

The regularization weight (λ) in STRidge is swept across various values as shown Fig. 3.15. The yellow line in Fig. 3.15 represents the value of λ at which the best identified PDE is selected. Note that STRidge was able to find the KdV equation at various values of the regularization weights (λ). Table 3.9 shows the KdV equation recovered by both GEP and STRidge. The physical model identified by STRidge is more accurate to the true PDE than the model identified by GEP.

Table 3.9: KdV equation identified by GEP and STRidge.

	Recovered	**Test error**
True	$u_t = -6.00\ uu_x + 1.00\ u_{3x}$	
GEP	$u_t = -5.96\ uu_x + 0.99\ u_{3x} - 5.84 \times 10^{-4}$	0.29
STRidge	$u_t = -6.04\ uu_x + 1.02\ u_{3x}$	0.02

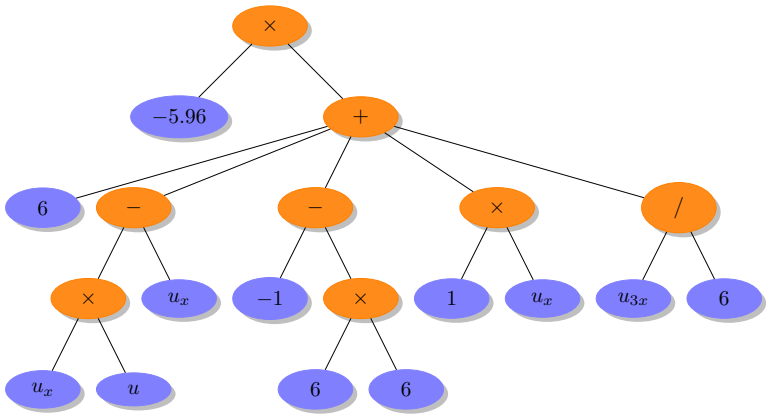

Figure 3.14: KdV equation in terms of ET identified by GEP.

3.6 Kawahara Equation

We consider the Kawahara equation, which is a fifth-order nonlinear PDE Kawahara (1972) shown in Table 3.1. This equation is sometimes also referred to as a fifth-order KdV equation or singularly perturbed KdV equation. The fifth-order KdV equation is one of the most well known nonlinear evolution equation which is used in the theory of magneto-acoustic waves in a plasma Kawahara (1972), capillary-gravity waves

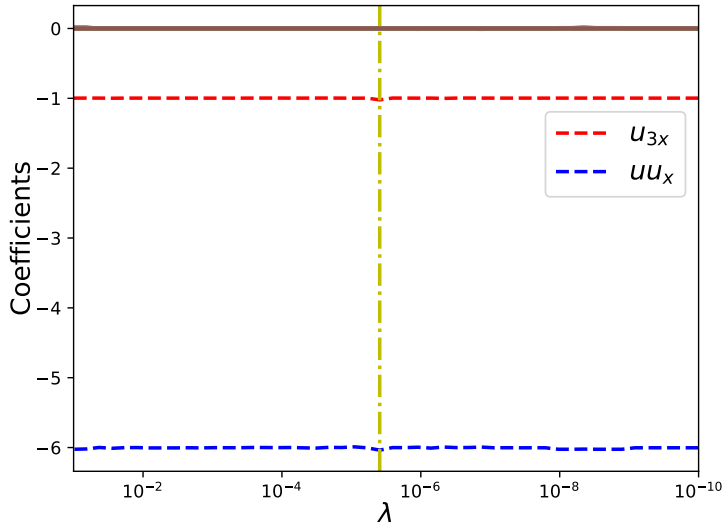

Figure 3.15: STRidge coefficients as a function of regularization parameter λ for the KdV equation.

Kawahara et al. (1975) and the theory of shallow water waves Hunter and Scheurle (1988). This test case is intended to test GEP and STRidge for identifying higher order derivatives from observing data. We use an analytical solutionSirendaoreji (2004) which is a traveling wave solution given in Table 3.1. This analytical solution also satisfies the linear wave equation and hence both GEP and STRidge may recover a wave PDE (not shown here) as this is the sparsest model represented by observed data (Fig. 3.16). For simplifying the analysis, we remove the potential basis u_x from the feature library Schaeffer (2017) ($\mathbf{\Theta}(\mathbf{U})$) for STRidge and additionally include uu_x basis in core feature library ($\widetilde{\mathbf{\Theta}}(\mathbf{U})$) for GEP.

Table 3.3 shows the GEP hyper-parameters used for the identification of the Kawahara equation. Due to simplifying the feature library, GEP requires smaller head length and single gene. Fig. 3.17 shows the identified PDE in the form of ET. When ET form is simplified, we can show that the resulting model is the Kawahara equation identified correctly along with its coefficients. For STRidge, the regularization weight (λ) is swept across various values as shown in Fig. 3.18. The yellow line in Fig. 3.18

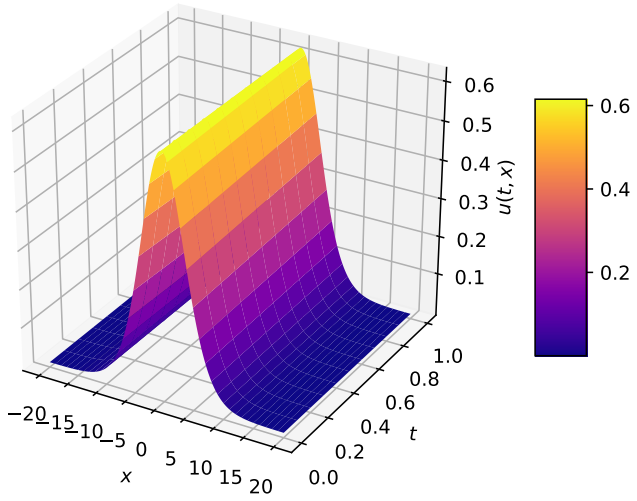

Figure 3.16: Analytical solution of the Kawahara equation.

represents the value of λ at which the best identified PDE is selected. Note that STRidge was able to find the Kawahara equation at various values of regularization weights (λ). Table 3.10 shows the Kawahara equation identified by both GEP and STRidge.

Table 3.10: Kawahara equation identified by GEP and STRidge.

	Recovered	**Test error**
True	$u_t = -1.0\ uu_x - 1.00\ u_{3x} - 1.0\ u_{5x}$	
GEP	$u_t = -1.0\ uu_x - 1.00\ u_{3x} - 1.0\ u_{5x} - 8.27 \times 10^{-8}$	5.29×10^{-11}
STRidge	$u_t = -1.0\ uu_x - 0.99\ u_{3x} - 1.0\ u_{5x}$	1.35×10^{-12}

3.7 Newell-Whitehead-Segel Equation

Newell-Whitehead-Segel (NWS) equation is a special case of the Nagumo equation Zhi-Xiong and Ben-Yu (1992). Nagumo equation is a nonlinear reaction-diffusion equation that models pulse transmission line simulating a nerve axon Nagumo et al. (1962), population genetics Aronson and Weinberger (1978), and circuit theory Scott

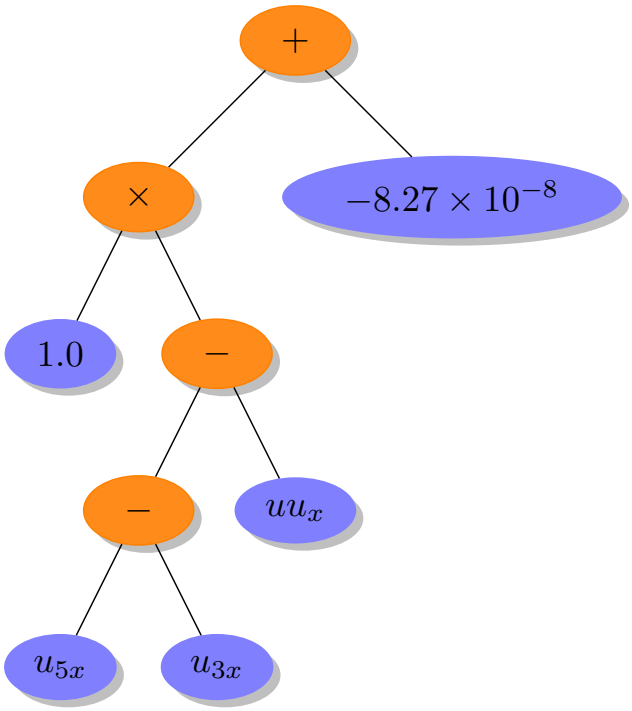

Figure 3.17: Kawahara equation in terms of ET identified by GEP.

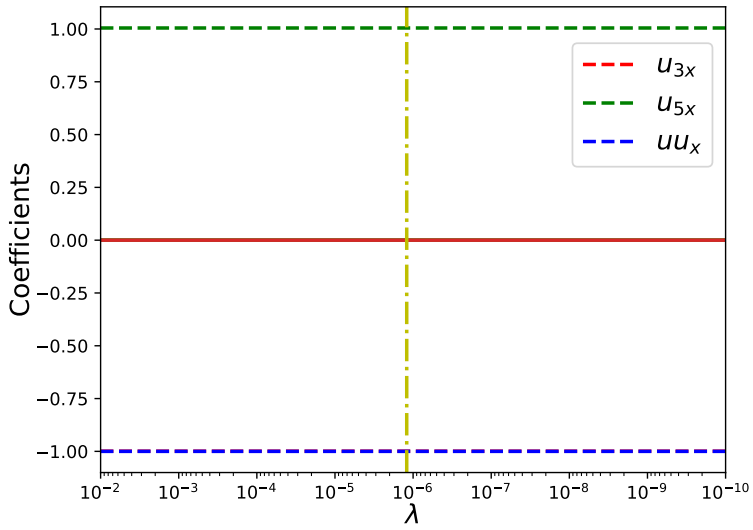

Figure 3.18: STRidge coefficients as a function of regularization parameter λ for the Kawahara equation.

(1963). The NWS equation and its analytical solution are shown in Table 3.1. We use a traveling wave solution Dehghan and Fakhar-Izadi (2011) that satisfies both wave and NWS equations (Fig. 3.19). We carry similar changes to the feature library that was applied to discovering the Kawahara equation.

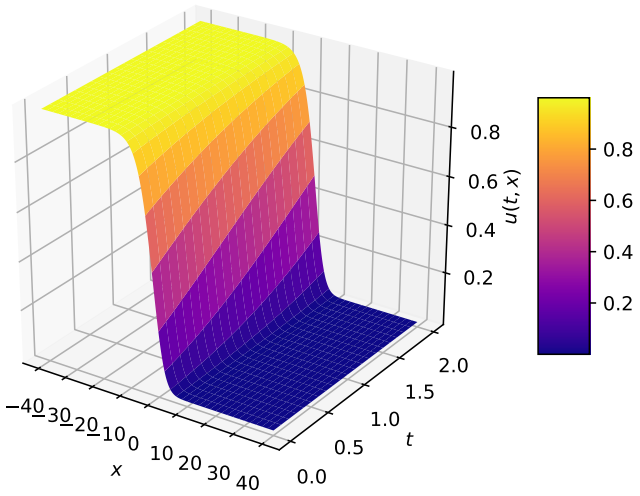

Figure 3.19: Analytical solution of the NWS equation.

Table 3.3 shows the GEP hyper-parameters used for identification of the NWS equation. However contrast to identifying the Kawahara equation with smaller head length and single gene from simplifying the feature library, for NWS case GEP requires larger head length and more genes for identifying PDE as shown in Table 3.3. This is due to the identification of nonlinear interaction feature u^2 that appears in the NWS equation. Fig. 3.20 shows the identified PDE in the form of ET. When ET form is simplified, we can show that the resulting model is the NWS equation identified along with its coefficients. For STRidge, the regularization weight (λ) is swept across various values as shown Fig. 3.21. The yellow line in Fig. 3.21 represents the value of λ at which the best identified PDE is selected. Note that STRidge was able to find the NWS equation at various values of regularization weights (λ). Table 3.11 shows

the NWS equation identified by both GEP and STRidge.

Table 3.11: NWS equation identified by GEP and STRidge.

	Recovered	**Test error**
True	$u_t = 1.00\ u_{2x} + 1.00\ u - 1.00\ u^2$	
GEP	$u_t = 0.99\ u_{2x} + 0.99\ u - 0.99\ u^2 - 8.27 \times 10^{-8}$	3.02×10^{-11}
STRidge	$u_t = 1.00\ u_{2x} + 0.99\ u - 0.99\ u^2$	1.36×10^{-11}

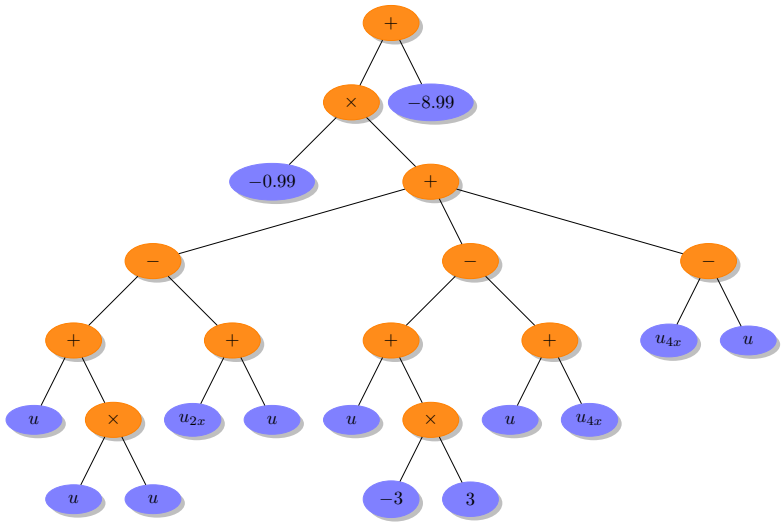

Figure 3.20: NWS equation in terms of ET identified by GEP.

3.8 Sine-Gordon Equation

Sine-Gordon equation is a nonlinear PDE that appears in propagating of fluxions in Josephson junctions Barone et al. (1971), dislocation in crystals Perring and Skyrme (1962) and nonlinear optics Whitham (2011). Sine-Gordon equation has a sine term that needs to be identified by GEP and STRidge by observing data (Fig. 3.22). This test case is straight forward for GEP as the function set includes trigonometric operators that help to identify the equation. However, the application of STRidge is suitable if features library is limited to basic interactions and does not contain a basis with trigonometric dependencies. STRidge may recover infinite series approximations

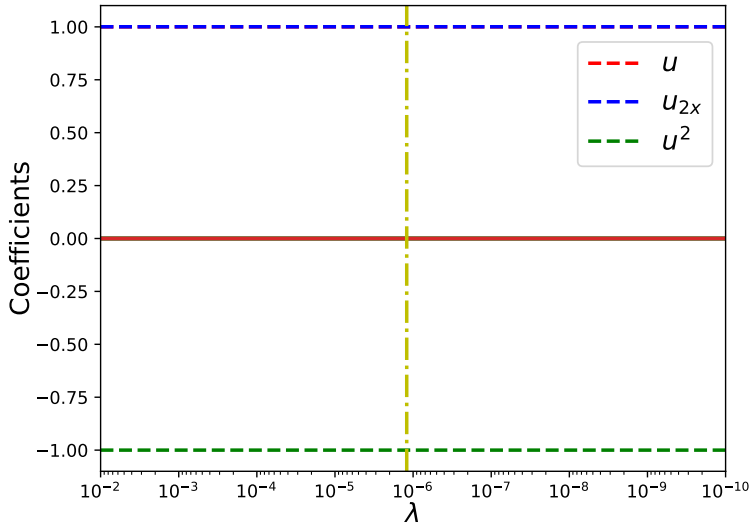

Figure 3.21: STRidge coefficients as a function of regularization parameter λ for the NWS equation.

if higher degree basic feature interactions are included in the feature library Brunton et al. (2016). Note that the output or target data for the Sine-Gordon equation consists of second order temporal derivative of velocity field $u(t, x)$. Hence, $\mathbf{V(t)}$ consists of u_{2t} measurements instead of u_t.

Table 3.3 shows the GEP hyper-parameters used for identifying the Sine-Gordon equation. For our analysis, GEP found the best model when the larger population size used. Fig. 3.23 shows the identified PDE in the form of ET. When ET form is simplified, we can show that the resulting model is the Sine-Gordon equation identified along with its coefficients. Table 3.12 shows the identified equation by GEP. This test case demonstrates the usefulness of GEP in identifying models with complex function composition and limitation of the expressive and predictive power of the feature library in STRidge.

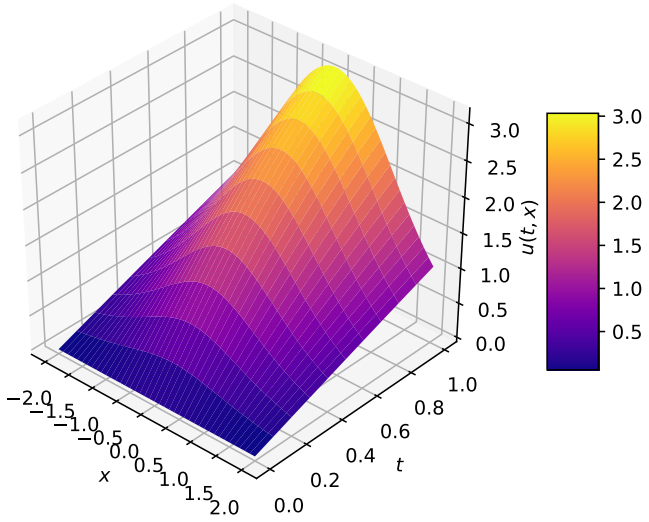

Figure 3.22: Analytical solution of the Sine-Gordon equation.

Table 3.12: Sine-Gordon equation identified by GEP.

	Recovered	**Test error**
True	$u_{2t} = 1.00\ u_{2x} - 1.00\ \sin(u)$	
GEP	$u_{2t} = 0.99\ u_{2x} - 0.99\ \sin(u) - 1.82 \times 10^{-5}$	1.57×10^{-4}

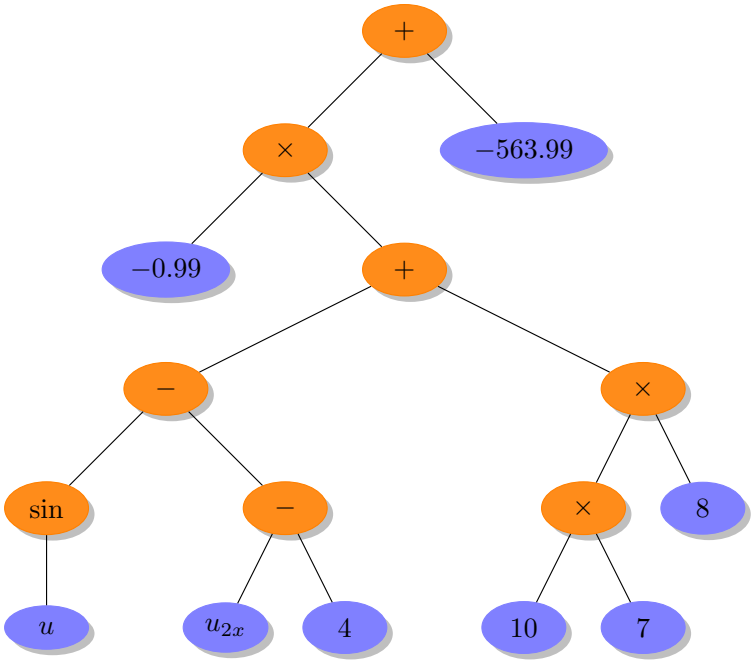

Figure 3.23: Sine-Gordon equation in terms of ET identified by GEP.

CHAPTER IV

Truncation Error Analysis

This section deals with constructing a modified differential equation (MDE) for the Burgers equation. We aim at demonstrating both GEP and STRidge techniques as SR tools in the identification of truncation errors resulting from an MDE of the Burgers nonlinear PDE. MDEs provide valuable insights into discretization schemes along with their temporal and spatial truncation errors. Initially, MDE analysis was developed to connect the stability of nonlinear difference equations with the form of the truncation errorsHirt (1968). In continuation, the symbolic form of MDEs were developed and a key insight was proposed that only the first few terms of the MDE dominate the properties of the numerical discretizationRitchmyer and Norton (1967). These developments of MDE analysis lead to increasing accuracy by eliminating leading order truncation error termsKlopfer and McRae (1983), improving stability of schemes by adding artificial viscosity terms Majda and Osher (1978), preserving symmetriesOzbenli and Vedula (2017b,a), and ultimately sparse identification of truncation errorsThaler et al. (2019). Therefore, MDE analysis plays a prominent role in implicit large eddy simulations (ILES) Adams et al. (2004) as truncation errors are shown to have inherent turbulence modelling capabilitiesMargolin and Rider (2002). Discretization schemes are tuned in the ILES approach as to model the subgrid scale tensor using truncation errors. As the construction of MDEs becomes cumbersome and intractable for complex flow configurations, data driven SR tools such as GEP and STRidge can be exploited for the identification of MDEs by observing the data.

For demonstration purposes, we begin by constructing an MDE of the Burgers

Table 4.1: GEP hyper-parameters selected for identification of truncation error terms of MDEs.

Hyper-parameters	**Burgers eq. (i)**	**Burgers eq. (ii)**
Head length	8	8
Number of genes	5	4
Population size	70	70
Generations	1000	1000
Length of RNC array	20	20
Random constant minimum	1.0×10^{-6}	1.0×10^{-5}
Random constant maximum	0.01	0.01

equation,

$$u_t + uu_x = \nu u_{2x}, \tag{4.1}$$

and discretizing Eq. (4.1) using first order schemes (i.e., forward in time and backward in space approximations for the spatial and temporal derivatives, respectively) and a second order accurate central difference approximation for the second order spatial derivatives. The resulting discretized Burgers PDE is shown below,

$$\frac{u_j^{p+1} - u_j^p}{dt} + u_j^p \frac{u_j^p - u_{j-1}^p}{dx} = \nu \frac{u_{j+1}^p - 2u_j^p + u_{j-1}^p}{dx^2}, \tag{4.2}$$

where temporal and spatial steps are given by dt and dx, respectively. In Eq. 4.2, the spatial location is denoted using subscript index j and the temporal snapshot using superscript index p.

To derive the modified differential equation (MDE) of the Burgers PDE, we

substitute the Taylor approximations for each term,

$$\left.\begin{aligned} u_j^{p+1} &= u_j^p + dt(u_t)_j^p + \frac{dt^2}{2}(u_{2t})_j^p + \frac{dt^3}{6}(u_{3t})_j^p + \dots \\ u_{j+1}^p &= u_j^p + dx((u_x))_j^p + \frac{dx^2}{2}(u_{2x})_j^p + \frac{dx^3}{6}(u_{3x})_j^p + \dots \\ u_{j-1}^p &= u_j^p - dx(u_x)_j^p + \frac{dx^2}{2}(u_{2x})_j^p - \frac{dx^3}{6}(u_{3x})_j^p + \dots \end{aligned}\right\} \tag{4.3}$$

When we substitute these approximations into Eq. 4.2, we obtain the Burgers MDE as follows,

$$(u_t + uu_x - \nu u_{2x})_j^p = -R, \tag{4.4}$$

where R represents truncation error terms of the Burgers MDE given as,

$$R = \frac{dt}{2}(u_{2t})_j^p + \frac{dx}{2}(uu_x)_j^p - \frac{\nu dx^2}{12}(u_{4x})_j^p + O(dt^2, dx^4). \tag{4.5}$$

Furthermore, temporal derivative in Eq. 4.5 is substituted with spatial derivatives resulting in,

$$R = dtuu_x^2 - dt\nu u_x u_{2x} - dt\nu u u_{3x} - \frac{dx}{2}uu_{2x} + \frac{dt}{2}u^2u_{2x} - \frac{\nu dx^2}{12}u_{4x} + O(dt^2, dx^4). \tag{4.6}$$

The truncation error or residual of discretized equation considering $u(t, x)$ as exact solution to the Burgers PDE is equal to the difference between the numerical scheme (Eq. 4.2) and differential equation (Eq. 4.1)Hirsch (2007). This results in discretized equation with residual as shown below,

$$u_j^{p+1} - u_j^p + u_j^p dt\frac{u_j^p - u_{j-1}^p}{dx} - \nu dt\frac{u_{j+1}^p - 2u_j^p + u_{j-1}^p}{dx^2} = Rdt. \tag{4.7}$$

We follow the same methodology for constructing the output data and feature library as discussed in Chapter II for the equation discovery. However, the output or target

data $\mathbf{V(t)}$ is stored with the left hand side of Eq. 4.7 denoted from now as $\mathbf{U_{er}}$. The resulting output and core feature library are shown below,

$$\left.\begin{aligned} \mathbf{V(t)} &= \begin{bmatrix} \mathbf{U_{er}} \end{bmatrix} \\ \widetilde{\mathbf{\Theta}}(\mathbf{U}) &= \begin{bmatrix} \mathbf{U} & \mathbf{U}_x & \mathbf{U}_{2x} & \mathbf{U}_{3x} & \mathbf{U}_{4x} \end{bmatrix} \end{aligned}\right\}. \tag{4.8}$$

The computation of the output data $\mathbf{V(t)}$ in Eq. 4.8 can be obtained using the analytical solution of the Burgers PDE. Furthermore, the derivatives in core feature library $\widetilde{\mathbf{\Theta}}(\mathbf{U})$ are calculated using the finite difference approximations given by Eq. 2.4. We use both analytical solutions listed in Table 3.1 for the Burgers equation (i) and the Burgers equation (ii) to test GEP and STRidge for recovering truncation error terms.

We use the same extended feature library $\widetilde{\mathbf{\Theta}}(\mathbf{U})$ as input to STRidge given in Eq. 2.7, but without the fifth order derivative. However, we add additional third degree interaction of features to $\widetilde{\mathbf{\Theta}}(\mathbf{U})$ to recover the truncation error terms containing third degree nonlinearities. The extra nonlinear features that are added to $\widetilde{\mathbf{\Theta}}(\mathbf{U})$ are given below,

$$\begin{aligned} [&\mathbf{U^2U_x} \quad \mathbf{U^2U_{2x}} \quad \mathbf{U^2U_{3x}} \quad \mathbf{U^2U_{4x}} \\ &\mathbf{UU_x^2} \quad \mathbf{UU_xU_{2x}} \quad \mathbf{UU_xU_{3x}} \quad \mathbf{UU_xU_{4x}}]. \end{aligned}$$

In contrast, GEP uses the core feature $\widetilde{\mathbf{\Theta}}(\mathbf{U})$ as input as it identifies the higher order nonlinear feature interactions automatically. This test case shows the natural feature extraction capability of GEP and need to modify the feature library to increase the expressive power of STRidge. The functional and terminal sets used for truncation error identification are listed in Table 4.2.

Table 4.2: GEP functional and terminal sets used for truncation error term recovery. '?' is a random constant.

Parameter	**Value**
Function set	$+, -, \times$
Terminal set	$\widetilde{\mathbf{\Theta}}(\mathbf{U})$, ?
Linking function	$+$

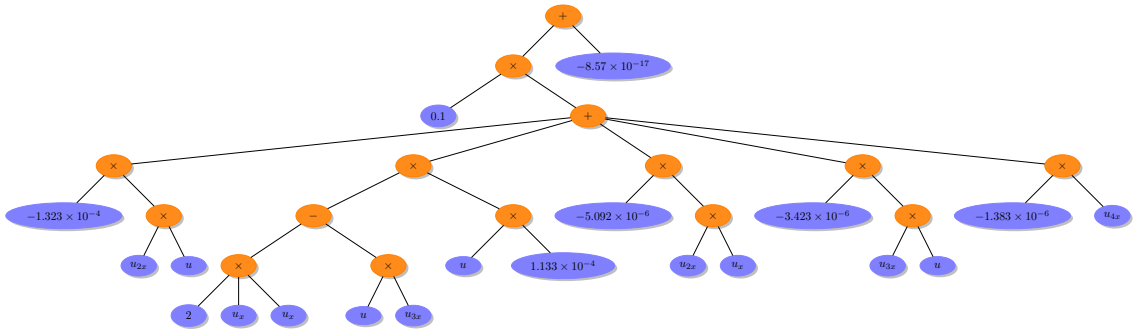

Figure 4.1: Truncation error of the Burgers MDE using analytical solution of the Burgers equation (i) in terms of ET identified by GEP.

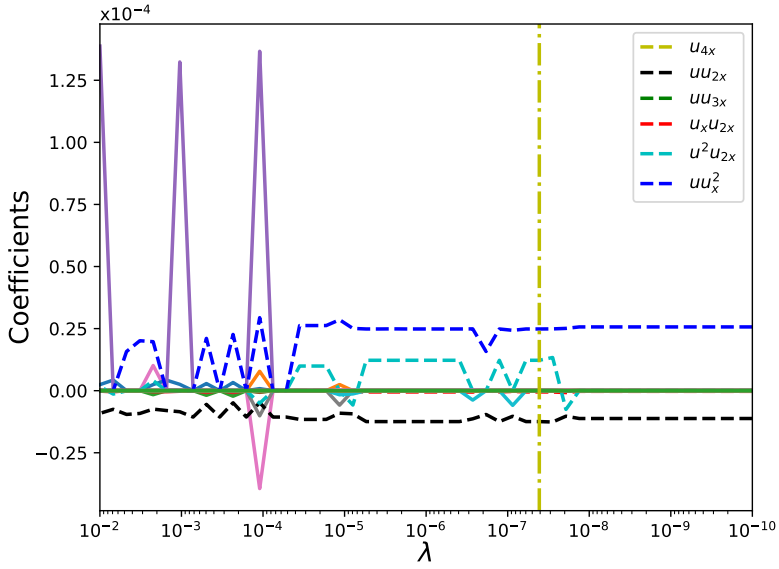

Figure 4.2: STRidge coefficients as a function of regularization parameter λ for truncation error of the Burgers MDE (i).

4.1 Burger Equation (i)

First, we test the recovery of truncation errors using the analytical solution of the Burgers equation (i) with the same spatial and temporal domain listed in Table 3.1. However, we set spatial discretization to be $dx = 0.005$ and temporal discretization to $dt = 0.005$ for storing the analytical solution $u(t, x)$. This test case needs a large population size, bigger head length, more genes and more iterations as given in Table 4.1, as the truncation error terms consist of nonlinear combinations of features and the coefficients of error terms that are generally difficult for GEP to identify. Fig. 4.1 shows the ET form of the identified truncation error terms. The regularization weight λ for STRidge is swept across a range of values as shown in Fig. 4.2. The vertical yellow line in Fig. 4.2 is the value of λ where STRidge identifies the best truncation error model. Table 4.3 shows the recovered error terms by GEP and STRidge along with their coefficients. Both GEP and STRidge perform well in identifying the nonlinear spatial error terms with STRidge predicting the error coefficient better than GEP.

4.2 Burger Equation (ii)

In the second case, we test the recovery of truncation errors using an analytical solution of the Burgers eq. (ii) with the same spatial and temporal domain listed in Table 3.1. We select the spatial discretization $dx = 0.005$ and the temporal discretization $dt = 0.1$ for propagating the analytical solution $u(t, x)$. This test case also follows the previous case where a large population size, bigger head length, more genes, and more iterations are needed as shown in Table 4.1. Fig. 4.3 shows the ET form of identified truncation error terms. The regularization weight λ for STRidge is swept across a range of values as shown in Fig. 4.4. In this test case, the coefficients change rapidly in respect to λ, and the best model is recovered only at the value of λ shown by the vertical yellow line in Fig. 4.4. Table 4.4 shows the recovered error terms by GEP and STRidge

along with their coefficients. Similar to the previous test case, STRidge predicts the truncation error coefficients better than GEP.

Table 4.3: Identified truncation error terms along with coefficients for the Burgers MDE (i) by GEP and STRidge.

	True	**GEP**	**Relative error (%)**	**STRidge**	**Relative error (%)**
uu_x^2	2.5×10^{-5}	2.26×10^{-5}	9.6	2.48×10^{-5}	0.8
$u_x u_{2x}$	-5.0×10^{-7}	-5.09×10^{-7}	1.8	-5.02×10^{-7}	0.4
uu_{3x}	-2.5×10^{-7}	-3.42×10^{-7}	36.8	-2.29×10^{-7}	8.4
$u^2 u_{2x}$	1.25×10^{-5}	1.13×10^{-5}	9.6	1.22×10^{-5}	2.4
u_{4x}	1.25×10^{-9}	1.38×10^{-9}	10.4	1.16×10^{-9}	7.2
uu_{2x}	-1.25×10^{-5}	-1.33×10^{-5}	6.4	-1.24×10^{-5}	0.8

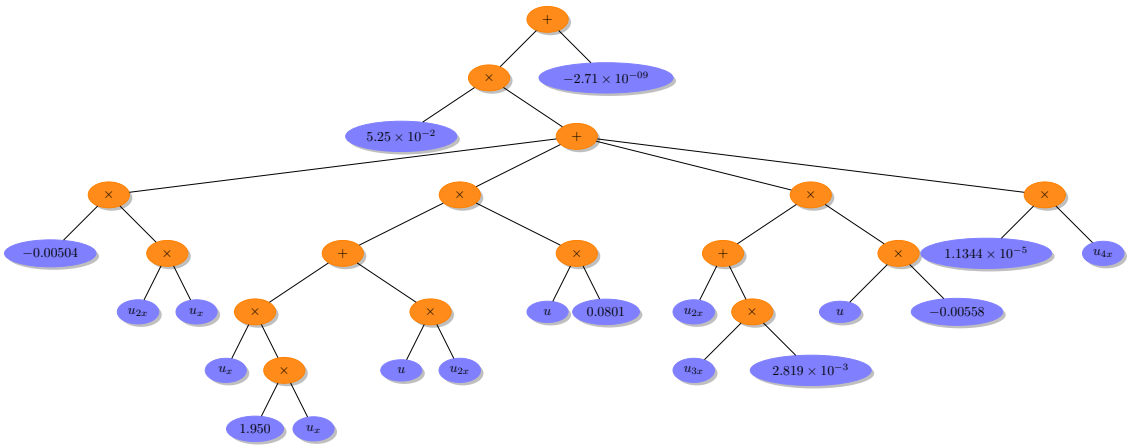

Figure 4.3: Truncation error term of the Burgers MDE using analytical solution of the Burgers equation (ii) in terms of ET identified by GEP.

Table 4.4: Identified truncation error terms along with coefficients for the Burgers MDE (ii) by GEP and STRidge.

	True	**GEP**	**Relative error (%)**	**STRidge**	**Relative error (%)**
uu_x^2	1.0×10^{-2}	8.19×10^{-3}	18.1	9.92×10^{-3}	0.8
$u_x u_{2x}$	-2.0×10^{-4}	-2.64×10^{-4}	32.0	-1.99×10^{-4}	0.5
uu_{3x}	-1.0×10^{-4}	-1.55×10^{-4}	55.0	-9.91×10^{-5}	0.9
$u^2 u_{2x}$	5.0×10^{-3}	4.21×10^{-3}	15.8	5.08×10^{-3}	1.6
u_{4x}	5.0×10^{-7}	5.65×10^{-7}	13.0	4.94×10^{-7}	1.2
uu_{2x}	-2.5×10^{-4}	-2.75×10^{-4}	10	-2.54×10^{-4}	1.6

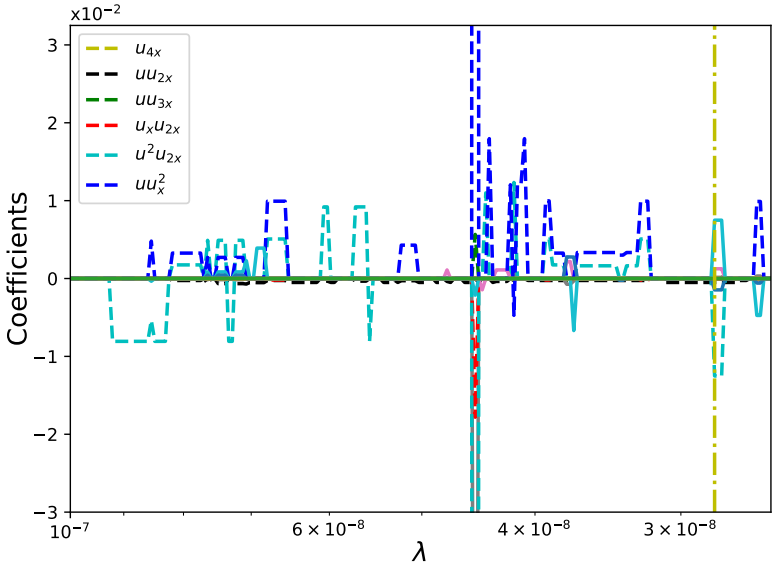

Figure 4.4: STRidge coefficients as a function of regularization parameter λ for truncation error of the Burgers MDE (ii).

CHAPTER V

Hidden Physics Discovery

In this section, we demonstrate the identification of hidden physical laws from sparse data mimicking sensor measurements using GEP and STRidge. Furthermore, we demonstrate the usefulness of GEP as a natural feature extractor that is capable of identifying complex functional compositions. However, STRidge in its current form is limited by its expressive power which depends on its input feature library. Many governing equations of complex systems in the modern world are only partially known or in some cases still awaiting first principle equations. For example, atmospheric radiation models or chemical reaction models might be not fully known in governing equations of environmental systemsKrasnopolsky and Fox-Rabinovitz (2006a,b). These unknown models are generally manifested in the right hand side of the known governing equations (i.e., dynamical core) behaving as a source or forcing term. The recent explosion of rapid data gathering using smart sensorsDhingra et al. (2019) has enabled researchers to collect data that the true physics of complex systems but their governing equations are only known partially. To this end, SR approaches might be able to recover these unknown physical models when exposed to data representing full physics.

To demonstrate the proof of concept for identification of unknown physics, we formulate a 1D advection-diffusion PDE and a 2D vortex-merger problem. These problems include a source term that represents the hidden physical law. We generate synthetic data that contains true physics and substitute this data set in to the known governing equations. This results in an unknown physical model left as a residual that must be recovered by GEP when exposed to a target or output containing the known

part of the underlying processes. Furthermore, both GEP and STRidge are tested to recover eddy viscosity kernels for the 2D Kraichnan turbulence problem. These eddy viscosity kernels are manifested as source terms in the LES equations that model unresolved small scales. Additionally, the value of the ad-hoc free modelling parameter that controls the dissipation in eddy viscosity models is also recovered using GEP and STRidge.

Table 5.1: GEP hyper-parameters selected for identifying source terms for the 1D advection-diffusion and the 2D vortex-merger problem.

Hyper-parameters	**1D advection-diffusion eq.**	**2D vortex-merger problem**
Head length	6	5
Number of genes	2	3
Population size	50	50
Generations	1000	500
Length of RNC array	5	8
Random constant minimum	$\frac{\pi}{4}$	$-\pi$
Random constant maximum	π	π

5.1 1D Advection-Diffusion PDE

In the first test case, we consider a 1D non-homogeneous advection-diffusion PDE which appears in many areas such as fluid dynamics Kumar (1983), heat transfer Isenberg and Gutfinger (1973), and mass transfer Guvanasen and Volker (1983). The non-homogeneous PDE takes the form,

$$u_t + cu_x = \alpha u_{2x} + S(t, x), \tag{5.1}$$

where c $= \frac{1}{3\pi}$, $\alpha = \frac{1}{4}$ and $S(t, x)$ is the source term.

We use an analytical solution $u(t, x)$ for solving Eq. 5.1. The exact solution for

this non-homogeneous PDE is as follows,

$$u(t,x) = \exp\left(\frac{\pi^2 t}{4}\right)\sin(\pi x), \tag{5.2}$$

where the spatial domain $x \in [0,1]$ and the temporal domain $t \in [0,1]$. We discretize the space and time domains with $n = 501$ and $m = 1001$, respectively. Fig. 5.1 shows the corresponding analytical solution $u(t,x)$.

The source term $S(t,x)$, which satisfies Eq. 5.1 for the analytical solution provided by Eq. 5.2, is given as,

$$S(t,x) = \frac{\pi^2}{2}\exp\left(\frac{\pi^2 t}{4}\right)\sin(\pi x) + \frac{1}{3}\exp\left(\frac{\pi^2 t}{4}\right)\cos(\pi x). \tag{5.3}$$

Our goal is to recover this hidden source term once the solution $u(t,x)$ is available either by solving the analytical equation given by Eq. 5.2 or by sensor measurements in real world applications. Furthermore, we select 64 random sparse spatial locations to mimic experimental data collection. After the solution $u(t,x)$ is stored at selected sparse spatial locations, we follow the same procedure for constructing output data and feature building as discussed in Chapter II. The corresponding output data $\mathbf{V}$ and feature library for recovering source term using GEP are given as,

$$\left.\begin{aligned} \mathbf{V} &= \begin{bmatrix} \mathbf{U_t} + c\mathbf{U_x} - \alpha\mathbf{U_{2x}} \end{bmatrix} \\ \widetilde{\mathbf{\Theta}} &= \begin{bmatrix} \mathbf{x} & \mathbf{t} \end{bmatrix} \end{aligned}\right\}. \tag{5.4}$$

The derivatives in the output data $\mathbf{V}$ are calculated using Eq. 2.4. Hence, to calculate spatial derivatives, we also store additional stencil data $u(t,x)$ around the randomly selected sparse locations $(u)_j^p$ i.e, $(u)_{j+1}^p$,$(u)_{j-1}^p$. Table 5.2 gives the functional and terminal sets used by GEP to recover the source term $S(t,x)$ given in Eq. 5.3.

Table 5.1 lists the hyper-parameters used by GEP for recovering source term of the

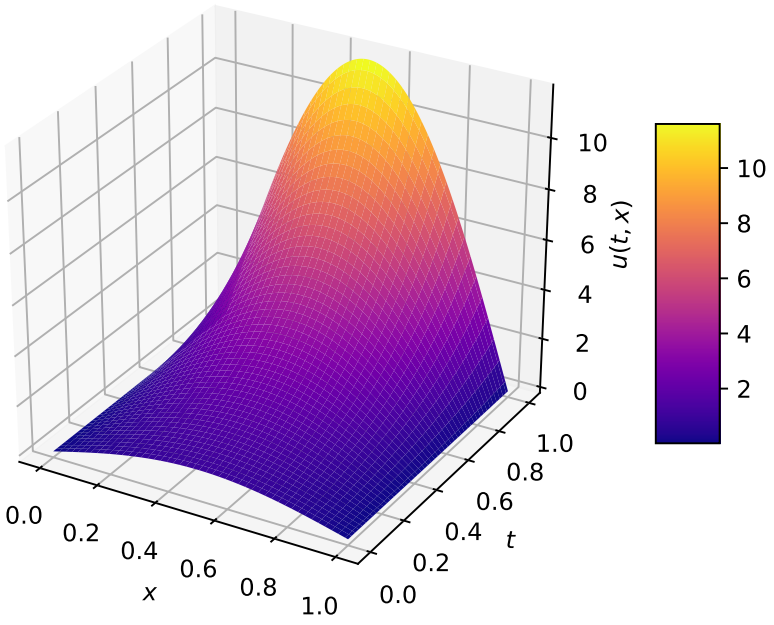

Figure 5.1: Solution to the 1D advection-diffusion PDE with source term.

Table 5.2: GEP functional and terminal sets used for source term identification. '?' is a random constant.

Parameter	**Value**
Function set	$+, -, \times, /$, exp, sin, cos
Terminal set	$\widetilde{\mathbf{\Theta}}$, ?
Linking function	$+$

1D advection-diffusion equation. As the hidden physical law given in Eq. 5.3 consists of complex functional compositions, GEP requires a larger head length, and more generations are required by GEP for identification. The ET form of the source term $S(t, x)$ found by GEP is shown in Fig. 5.2. The identified source term after simplifying the ET form found by GEP is listed in Table 5.3. GEP was able to identify the source term $S(t, x)$ given in Eq. 5.3 from sparse data.

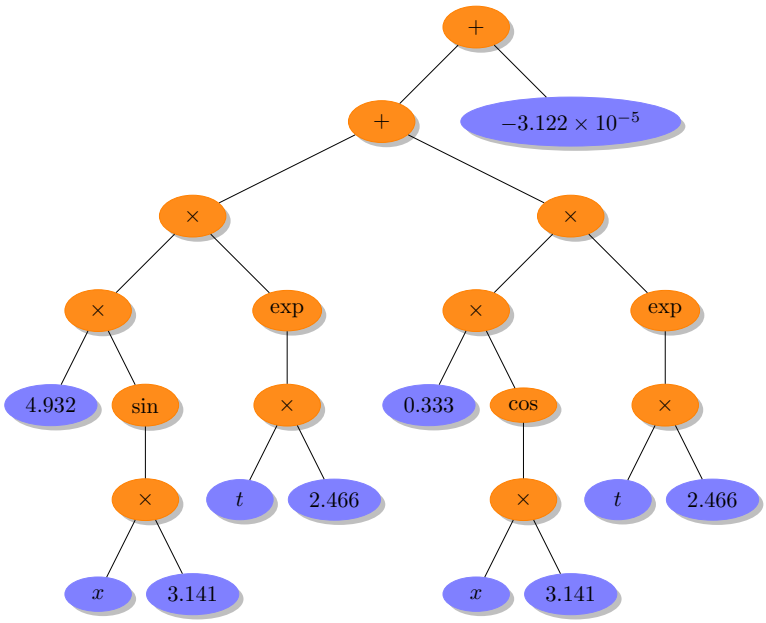

Figure 5.2: Hidden source term of the 1D advection-diffusion PDE in terms of ET identified by GEP.

Table 5.3: Hidden source term (S) of the 1D advection-diffusion PDE identified by GEP.

	Recovered	**Test error**
True	$S = 4.93 \exp(2.47\ t) \sin(3.14\ x) + 0.33 \exp(2.47\ t) \cos(3.14\ x)$	
GEP	$S = 4.93 \exp(2.46\ t) \sin(3.14\ x) + 0.33 \exp(2.46\ t) \cos(3.14\ x) - 3.12 \times 10^{-5}$	3.34×10^{-7}

5.2 2D Vortex-Merger Problem

In this section, we demonstrate the recovery of hidden physical law from the data generated by solving the vortex-merger problem with source terms. The initial two

vortices merge to form a single vortex when they are located within a certain critical distance from each other. This two-dimensional process is one of the fundamental processes of fluid motion and it plays a key role in a variety of simulations, such as decaying two-dimensional turbulenceMeunier et al. (2005); San and Staples (2012) and mixing layersSan and Staples (2013). This phenomenon also occurs in other fields such as astrophysics, meteorology, and geophysicsReinaud and Dritschel (2005). The Vortex-merger problem is simulated by using the 2D incompressible Navier-Stokes equations in the domain with periodic boundary conditions.

We specifically solve the system of PDEs called vorticity-streamfunction formulation. This system of PDEs contains the vorticity transport equation derived from taking the curl of the 2D incompressible Navier-Stokes equations and the Poisson equation representing the kinematic relationship between the streamfunction (ψ) and vorticity (ω). The resulting vorticity-streamfunction formulation with source term is given as,

$$\left.\begin{aligned} \omega_t + J(\omega,\psi) &= \frac{1}{\mathrm{Re}}\nabla^2\omega + S(t,x,y) \\ \nabla^2\psi &= -\omega \end{aligned}\right\} \tag{5.5}$$

where the Reynolds number is set to Re = 2000. In Eq. 5.5, $S(t,x,y)$ is the source term and $J(\omega,\psi)$ is the Jacobian term given as $\psi_y\omega_x - \psi_x\omega_y$. We use the Cartesian domain $(x,y) \in [0,2\pi] \times [0,2\pi]$ with a spatial resolution of 128×128. The initial vorticity field consisting of a co-rotating vortex pair is generated using the superposition of two Gaussian-distributed vortices given by,

$$\begin{aligned} \omega(0,x,y) = \Gamma_1 \exp\left(-\rho\left[(x-x_1)^2 + (y-y_1)^2\right]\right) \\ + \Gamma_2 \exp\left(-\rho\left[(x-x_2)^2 + (y-y_2)^2\right]\right), \end{aligned} \tag{5.6}$$

where the circulation $\Gamma_1 = \Gamma_2 = 1$, the interacting constant $\rho = \pi$ and the intial vortex centers are located near each other with coordinates $(x_1, y_1) = (\frac{3\pi}{4}, \pi)$ and

$(x_2, y_2) = (\frac{5\pi}{4}, \pi)$. We choose the source term $S(t, x)$ as,

$$S(t, x, y) = \Gamma_0 \sin(x)\cos(y)\exp\left(\frac{-4\pi^2}{\mathrm{Re}}t\right), \tag{5.7}$$

where the magnitude of the source term is set to $\Gamma_0 = 0.01$.

The vorticity field ω and streamfunction fieldψ are obtained by solving the Eq. 5.5 numerically. We use a third-order Runge-Kutta scheme for the time integration, and a second order Arakawa scheme Arakawa (1966) for the discretization of the Jacobian term $J(\omega, \psi)$. As we have a periodic domain, we use a fast Fourier transform (FFT) for solving the Poisson equation in Eq. 5.5 to obtain the streamfunction at every time step. Numerical details for solving the vortex-merger problem can be found in San et alSan and Staples (2013); Pawar and San (2019). We integrate the solution from time $t = 0$ to $t = 20$ with a temporal step $dt = 0.01$.

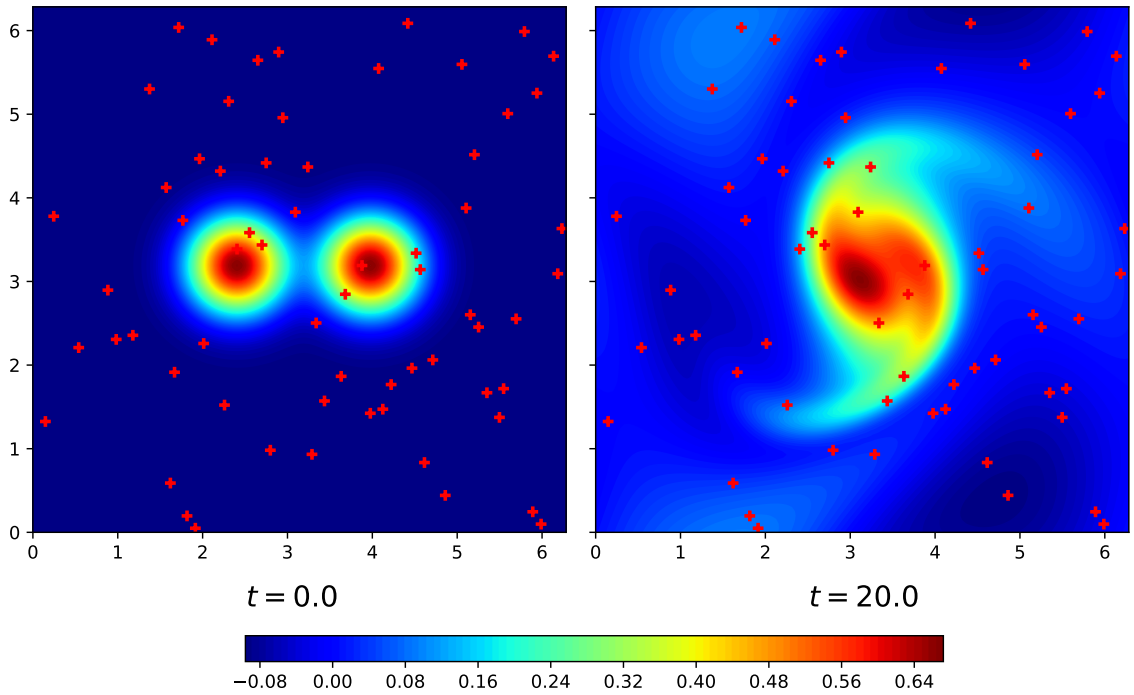

Figure 5.3: The 2D vortex-merger problem with source term at time $t = 0.0$ and $t = 20.0$. The red markers shows 64 random sensor locations used to collect vorticity (ω) and streamfunction (ψ) data for recovering source term $S(t, x, y)$.

Fig. 5.3 shows the merging process of two vortices at the initial and final times. The red markers in Fig. 5.3 are 64 randomly selected sparse locations to collect both streamfunction ψ and vorticity ω data. Once the streamfunction and vorticity data at sparse locations are available, we can construct the target data $\mathbf{V}$ and feature library $\widetilde{\mathbf{\Theta}}$ as discussed in Chapter II. The resulting input-response data is given as,

$$\left.\begin{aligned} \mathbf{V} &= \left[\begin{array}{c} \boldsymbol{\omega_t} + \boldsymbol{J(\omega,\psi)} - \dfrac{1}{\mathrm{Re}}\boldsymbol{\nabla^2\omega} \end{array}\right] \\ \widetilde{\mathbf{\Theta}} &= \left[\begin{array}{ccc} \mathbf{x} & \mathbf{y} & \mathbf{t} \end{array}\right] \end{aligned}\right\}. \tag{5.8}$$

The derivatives in the output data $\mathbf{V(t)}$ are calculated using finite difference approximations similar to Eq. 2.4. As streamfunction $(\psi)^p_{i,j}$ and vorticity $(\omega)^p_{i,j}$ data are selected only at sparse spatial locations, we also store the surrounding stencil, i.e., $(\psi)^p_{i+1,j}$, $(\psi)^p_{i-1,j}$, $(\psi)^p_{i,j+1}$, $(\psi)^p_{i,j-1}$, and $(\omega)^p_{i+1,j}$, $(\omega)^p_{i-1,j}$, $(\omega)^p_{i,j+1}$, $(\omega)^p_{i,j-1}$ in order to calculate the derivatives. The index i represents spatial location in x direction, and j represents spatial location in y direction.

In this test case, we demonstrate the identification of hidden physics which is the source term $S(t,x,y)$ given by Eq. 5.7 from the data obtained at sparse spatial locations using GEP. Table 5.1 lists the hyper-parameters used by GEP to recover the hidden physical law. We use the same function and terminal sets as shown in Table 5.2 but $\times$ is used as a linking function. Fig. 5.4 shows the ET form of hidden physical law (source term) obtained by GEP. Simplification of the ET form shows the identified source term which is close to true source term as shown in Table 5.4.

Table 5.4: Hidden source term (S) of the 2D vortex-merger problem identified by GEP.

	Recovered	**Test error**
True	$S = 0.0100\ \sin(x)\ \cos(y)\ \exp(-0.078\ t)$	
GEP	$S = 0.0099\ \sin(x)\ \cos(y)\ \exp(-0.078\ t) - 1.47 \times 10^{-6}$	1.35×10^{-8}

The 1D advection-diffusion and 2D vortex-merger problem demonstrate the use-

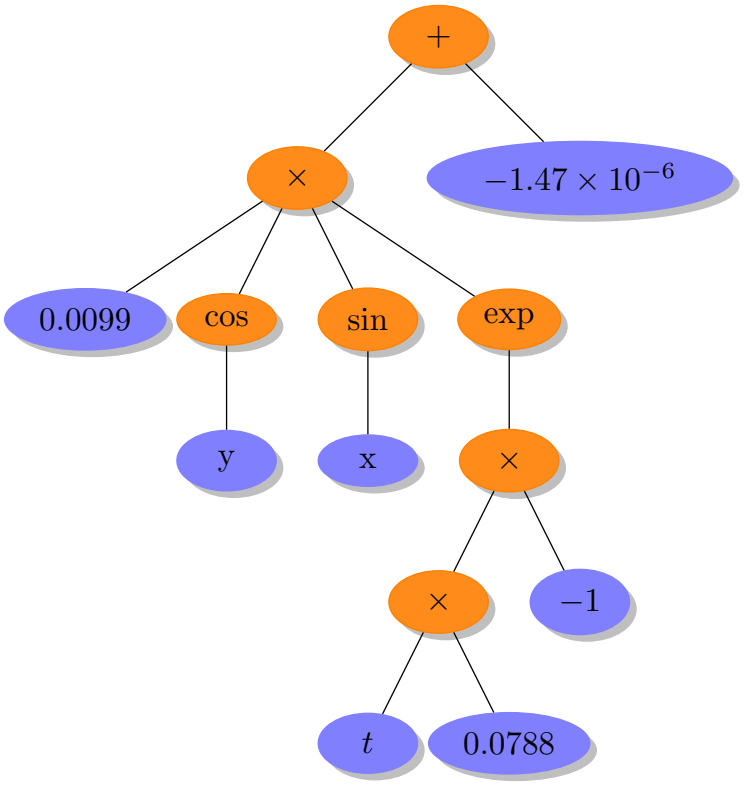

Figure 5.4: Hidden source term of the 2D vortex-merger problem in terms of ET identified by GEP.

fulness of GEP in recovering hidden physics, i.e., a source term that composed of complex functions using randomly selected sparse data. The expressive power of the feature library limits the applications of STRidge for identifying complex composition models. However, STRidge might be able to identify the infinite series approximations of these nonlinear functions Brunton et al. (2016). In the next test case, we use both STRdige and GEP to identify eddy viscosity kernels along with their free modelling coefficient that controls the dissipation of these kernels.

CHAPTER VI

Subgrid Scale Modelling

6.1 2D Kraichnan Turbulence

The concept of two-dimensional turbulence helps in understanding many complex physical phenomenon such as geophysical and astrophysical flowsBoffetta and Musacchio (2010); Boffetta and Ecke (2012). The equations of two-dimensional turbulence can model idealized flow configurations restricted to two-dimensions such as flows in rapidly rotating systems and in thin films over rigid bodies. The physical mechanism associated with the two-dimensional turbulence is explained by the Kraichnan-Batchelor-Leith (KBL) theoryKraichnan (1967); Batchelor (1969); Leith (1971). Generally, large eddy simulation (LES) is performed for both two and three dimensional flows to avoid the fine resolution and thereby computational requirements of direct numerical simulation (DNS) Piomelli (1999); Meneveau and Katz (2000). In LES, the flow variables are decomposed into resolved low wavenumber (or large scale) and unresolved high wavenumber (or small scale). This is achieved by the application of a low pass spatial filter to the flow variables. By arresting high wavenumber content (small scales), we can reduce the high resolution requirement of DNS, and hence faster simulations and reduced storage requirements. However, the procedure of introducing a low pass filtering results in an unclosed term for the LES governing equations representing the finer scale effects in the form of a source term.

Thus the quality of LES depends on the modeling approach used to close the spatial filtered governing equations to capture the effects of the unresolved finer scalesSagaut (2006). This model also called the subgrid scale model is a critical part

of LES computations. A functional or eddy viscosity approach is one of the popular approaches to model this closure term. These approaches propose an artificial viscosity to mimic the dissipative effect of the fine scales. Some of the popular functional models are the Smagorinsky Smagorinsky (1963), Leith Leith (1968), Balwin-LomaxBaldwin and Lomax (1978) and Cebeci-smith modelsSmith and Cebeci (1967). All these models require the specification of a model constant that controls the quantity of dissipation in the simulation, and its value is often set based on the nature of the particular flow being simulated. In this section, we demonstrate the identification of an eddy viscosity kernel (model) along with its ad-hoc model constant from observing the source term of the LES equation using both GEP and STRidge as robust SR tools. To this end, we use the vorticity-streamfunction formulation for two-dimensional fluid flows given in Eq. 5.5. We derive the LES equations for the two dimensional Kraichnan turbulence by applying a low pass spatial filter to the vorticity-streamfunction PDE given in Eq. 5.5. The resulting filtered equation is given as,

$$\bar{\omega}_t + \overline{J(\psi, \omega)} = \frac{1}{\mathrm{Re}} \nabla^2 \bar{\omega}, \tag{6.1}$$

where Re is the Reynolds number of the flow and $J(\omega, \psi)$ is the Jacobian term given as $\psi_y \omega_x - \psi_x \omega_y$. Furthermore the Eq. 6.1 is rearranged as,

$$\bar{\omega}_t + J(\bar{\psi}, \bar{\omega}), = \frac{1}{\mathrm{Re}} \nabla^2 \bar{\omega} + \Pi, \tag{6.2}$$

where the LES source term Π is given as,

$$\Pi = J(\bar{\psi}, \bar{\omega}) - \overline{J(\psi, \omega)}. \tag{6.3}$$

The source term Π in Eq. 6.3 represents the influence of the subgrid scales on larger resolved scales. The term $\overline{J(\psi, \omega)}$ is not available, which necessitates the use of

a closure modelling approach. In functional or eddy viscosity models, the source term of LES equations is represented as,

$$\Pi = \nu_e \nabla^2 \bar{\omega}. \tag{6.4}$$

where eddy viscosity ν_e is given by, but not limited to, the Smagorinsky, Leith, Baldwin-Lomax, and Cebeci-Smith kernels. The choice of these eddy viscosity kernels essentially implies the choice of a certain function of local field variables such as the strain rate or gradient of vorticity as a control parameter for the magnitude of ν_e.

Table 6.1: GEP functional and terminal sets used for identifying eddy viscosity kernel. '?' is a random constant.

Parameter	**Value**
Function set	$+, -, \times, /$
Terminal set	$\widetilde{\mathbf{\Theta}}$, ?
Linking function	$+$

In Smagorisnky model, the eddy viscosity kernel is given by,

$$\nu_e = (c_s \delta)^2 |\bar{S}|, \tag{6.5}$$

where c_s is a free modelling constant that controls the magnitude of the dissipation and δ is a characteristic grid length scale given by the square root of the product of the cell sizes in each direction. The $|\bar{S}|$ is based on the second invariant of the filtered field deformation, and given by,

$$|\bar{S}| = \sqrt{4\bar{\psi}_{xy}^2 + (\bar{\psi}_{2x} - \bar{\psi}_{2y})^2}, \tag{6.6}$$

The Leith model proposes that eddy viscosity kernel is a function of vorticity and

given as,

$$\nu_e = (c_s\delta)^3|\nabla\bar{\omega}|, \tag{6.7}$$

where $|\nabla\bar{\omega}|$ controls the dissipative character of the eddy viscosity as against the resolved strain rate used in the Smagorinsky model. The magnitude of the gradient of vorticity is defined as,

$$|\nabla\bar{\omega}| = \sqrt{\bar{\omega}_x^2 + \bar{\omega}_y^2}. \tag{6.8}$$

Table 6.2: GEP hyper-parameters selected for identification of the eddy viscosity kernel for the Kraichnan turbulence.

Hyper-parameters	**Kraichnan turbulence**
Head length	2
Number of genes	2
Population size	20
Generations	500
Length of RNC array	3
Random constant minimum	-1
Random constant maximum	1

The Baldwin-Lomax is an alternative approach that models the eddy viscosity kernel as,

$$\nu_e = (c_s\delta)^2|\bar{\omega}|, \tag{6.9}$$

where $|\bar{\omega}|$ is the absolute value of the vorticity considered as a measure of the local energy content of the flow at a grid point and also a measure of the dissipation required at that location.

The Cebeci-Smith model was devised for the Reynolds Averaged Navier-Stokes (RANS) applications. The model is modified for LES setting, and is given as,

$$\nu_e = (c_s\delta)^2|\overline{\Omega}|, \tag{6.10}$$

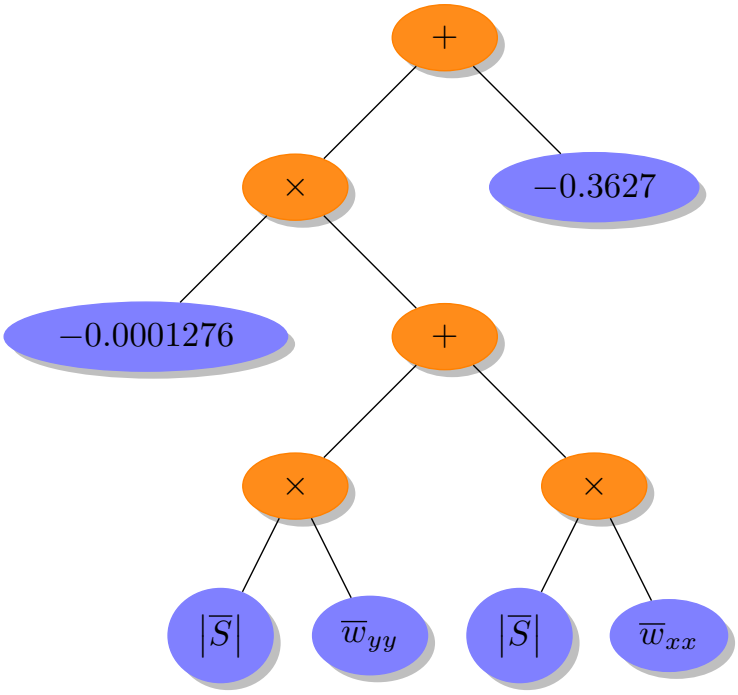

Figure 6.1: Samgorisnsky kernel in terms of ET identified for the two-dimensional Kraichnan turbulence problem by GEP.

where $|\overline{\Omega}|$ is given as,

$$|\overline{\Omega}| = \sqrt{\overline{\psi}_{2x}^2 + \overline{\psi}_{2y}^2}. \tag{6.11}$$

Table 6.3: LES source term (Π) for two-dimensional Kraichnan turbulence problem identified by GEP and STRidge.

	Recovered
GEP	$\Pi = 0.000128 \ \ \|S\| \ \ w_{2x} + 0.000128 \ \ \|S\| \ \ w_{2y} - 0.362$
STRidge	$\Pi = 0.000132 \ \ \|S\| \ \ w_{2x} + 0.000129 \ \ \|S\| \ \ w_{2y}$

High fidelity DNS simulations are performed for Eq. 6.1. We use a square domain of length 2π with periodic boundary conditions in both directions. We simulate homogeneous isotropic decaying turbulence which may be specified by an initial energy spectrum that decays through time. High fidelity DNS simulations are carried out for Re $= 4000$ with 1024×1024 resolution from time $t = 0$ to $t = 4.0$ with time step

0.001. The filtered flow quantities and LES source term Π in Eq. 6.3 are obtained from coarsening the DNS quantities to obtain quantities with a 64×64 resolution. The further details of solver and coarsening can be found in San and StaplesSan and Staples (2012). Once the LES source term Π in Eq. 6.3 and filtered flow quantities are obtained, we build the feature library and output data similar to the discussion in Chapter II. The resulting input-response data is given as,

$$\left.\begin{aligned} \mathbf{V} &= \begin{bmatrix} \boldsymbol{\Pi} \end{bmatrix} \\ \widetilde{\boldsymbol{\Theta}} &= \begin{bmatrix} \bar{\omega}_{2x} & \bar{\omega}_{2y} & |\bar{S}| & |\nabla\bar{\omega}| & |\bar{\omega}| & |\bar{\Omega}| \end{bmatrix} \end{aligned}\right\}. \tag{6.12}$$

GEP uses the output and feature library given in Eq. 6.12 to automatically extract the best eddy viscosity kernel for decaying turbulence problems along with the model's ad-hoc coefficient.

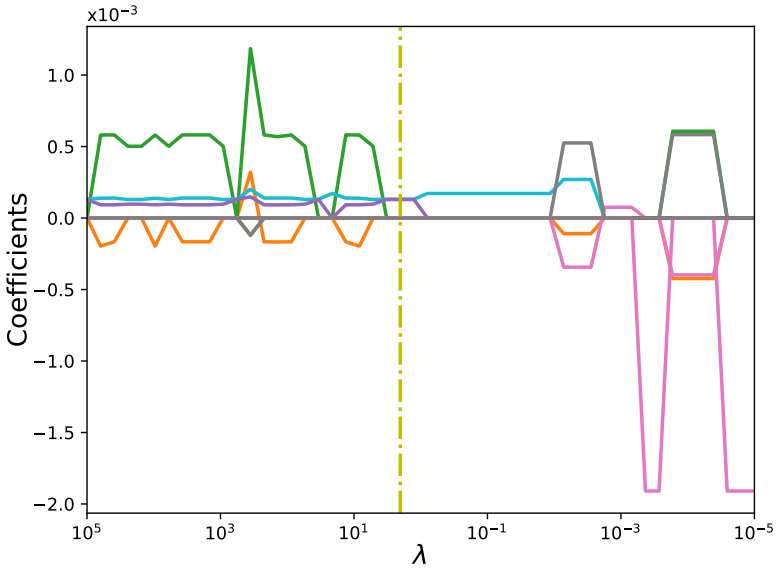

Figure 6.2: STRidge coefficients as a function of regularization parameter λ for the two-dimensional Kraichnan turbulence problem.

The extended feature library is constructed to include nonlinear interactions up to the quadratic degree to expand the expressive power for the STRidge algorithm. The

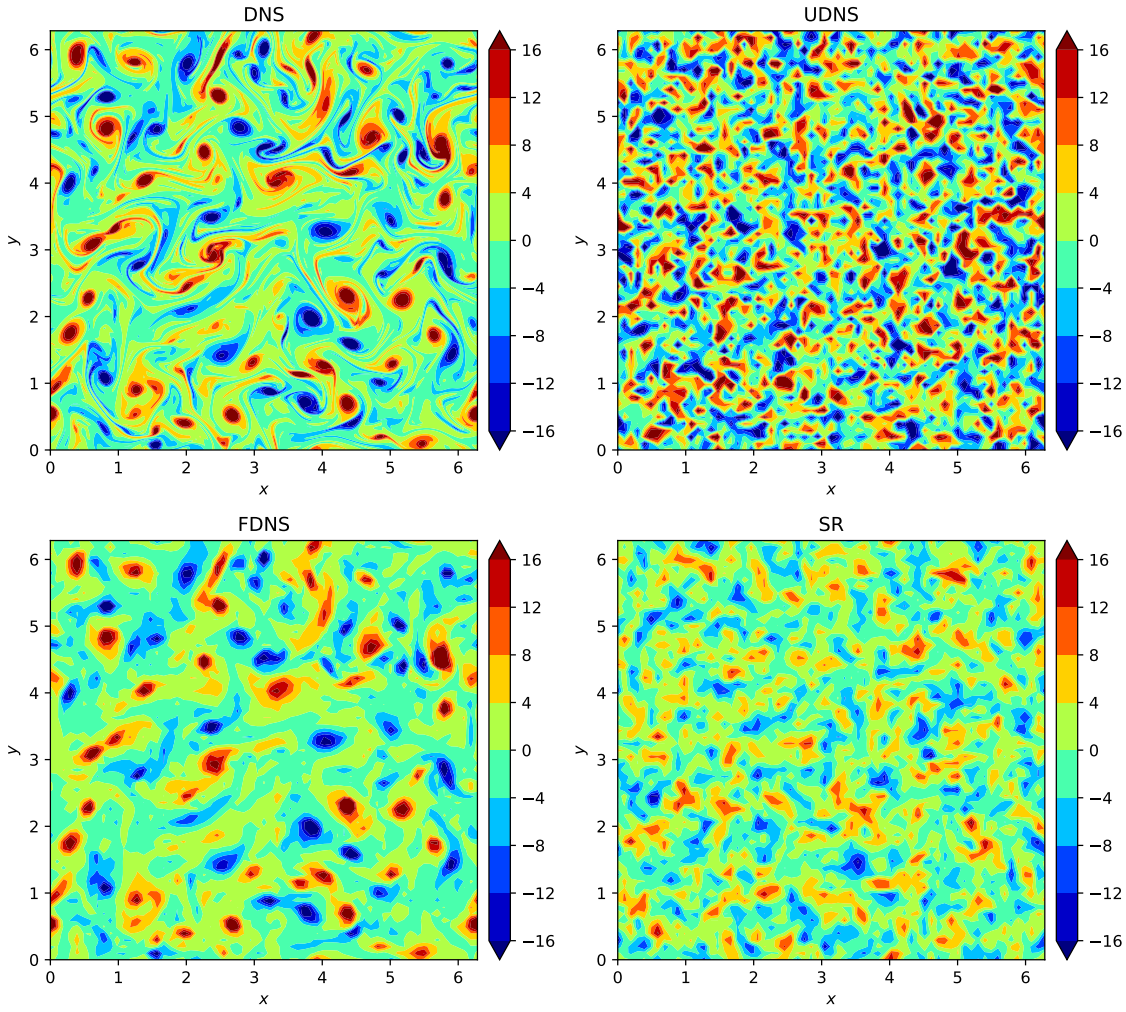

Figure 6.3: Controur plots for the two-dimensional Kraichnan turbulence problem at $t = 4$. SR refers to the identified model of the Smagorinsky kernel with $c_s = 0.12$. UDNS and FDNS refer to the no-model and filtered DNS simulations, respectively.

resulting extended feature library is given as,

$$\boldsymbol{\Theta} = \left[\begin{array}{ccccccc} \mathbf{1} & \bar{\boldsymbol{\omega}}_{\boldsymbol{2x}} & \bar{\boldsymbol{\omega}}_{\boldsymbol{2x}}{}^2 & \bar{\boldsymbol{\omega}}_{\boldsymbol{2y}} & \bar{\boldsymbol{\omega}}_{\boldsymbol{2x}}\bar{\boldsymbol{\omega}}_{\boldsymbol{2y}} & \bar{\boldsymbol{\omega}}_{\boldsymbol{2y}}{}^2 & \ldots & |\bar{\boldsymbol{\Omega}}|^{\mathbf{2}} \end{array} \right]. \tag{6.13}$$

The function and terminal sets used for identification of eddy viscosity kernel by GEP are listed in Table 6.1. Furthermore, the hyper-parameters of GEP are listed in Table 6.2. Both GEP and STRidge identify the Smagorinsky kernel with approximately the same coefficients as shown in Table 6.3. The ET form of the Smagorinsky kernel found by GEP is shown in Fig. 6.1. The regularization weight λ is varied to recover multiple models of different complexity as shown in Fig. 6.2. The yellow line in Fig. 6.2 corresponds to the value of λ where STRidge identifies the Smagorinsky kernel. We can take the average coefficient from both SR tools and derive the value of the free modelling constant identified by SR approaches. The average model of both approaches is given by,

$$\Pi = 0.000129\ (|S|\ w_{2x} + |S|\ w_{2y}). \tag{6.14}$$

By comparing with Eq. 6.4 and Eq. 6.5 and using the spatial cell size $\delta = \frac{2\pi}{64}$, the value of the free modelling constant is retrieved as $c_s = 0.12$.

The SR identified Smagorinsky kernel with $c_s = 0.12$ is plugged into the LES source term Π in Eq. 6.2 and a forward LES simulation is run for the 2D decaying turbulence problem. Fig. 6.3 shows the vorticity fields at time $t = 4.0$ for the DNS, under-resolved no-model simulation (UDNS), filtered DNS (FDNS), and LES with SR retrieved Smagorinsky kernel. Energy spectra at time $t = 4.0$ are showed in Fig. 6.4. We can observe that SR approaches satisfactorily identify the value of the modelling constant c_s, which controls reasonably well the right amount of dissipation needed to account for the unresolved small scales. We also highlight that several deep learning frameworks such as ANNs have been exploited for subgrid scale modelling for 2D

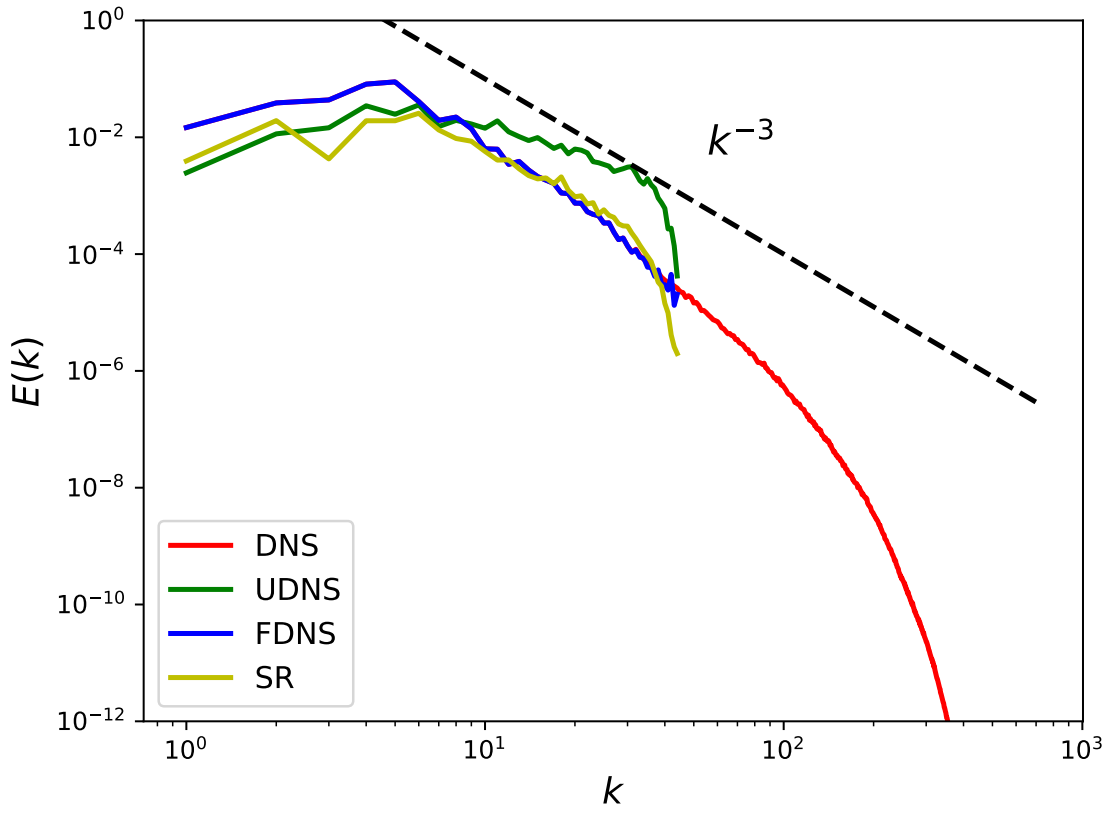

Figure 6.4: Energy spectra for the two-dimensional Kraichnan turbulence problem at $t = 4$. SR refers to the identified model of the Smagorinsky kernel with $c_s = 0.12$. UDNS and FDNS refer to the no-model and filtered DNS simulations, respectively.

Kraichnan turbulenceMaulik et al. (2019b, 2018, 2019a). The importance of feature selection can be seen in these works where different invariant kernels, like those listed in the feature library given in Eq. 6.12, are used as inputs to improve the ANN's predictive performance. The authors compared a posteriori results with different free modelling coefficients of the Smagorinsky and Leith models. Furthermore, it is evident from the energy spectrum comparisons in their studies that the appropriate addition of dissipation with the right tuning of the free modelling coefficient can lead to better predictions of the energy spectrum. To this end, SR approaches automatically distill traditional models along with the right values for the ad-hoc free modelling coefficients. Although the present study establishes a modular regression approach for discovering the relevant free parameters in LES models, we highlight that it can be extended easily to a dynamic closure modelling framework reconstructed automatically by sparse data on the fly based on the flow evolution, a topic we would like to address in future.

CHAPTER VII

Conclusion and Future Work

7.1 Conclusion

Data driven symbolic regression tools can be extremely useful for researchers for inferring complex models from sensor data when the underlying physics is partially or completely unknown. Sparse optimization techniques are envisioned as an SR tool that is capable of recovering hidden physical laws in a highly efficient computational manner. Popular sparse optimization techniques such as LASSO, ridge, and elastic-net are also known as feature selection methods in machine learning. These techniques are regularized variants of least squares regression adapted to reduce overfitting and promote sparsity. The model prediction ability of sparse regression methods is primarily dependent on the expressive power of its feature library which contains exhaustive combinations of nonlinear basis functions that might represent the unknown physical law. This limits the identification of physical models that are represented by complex functional compositions. GEP is an evolutionary optimization algorithm widely adapted for the SR approach. This genotype-phenotype algorithm takes advantage of the simple chromosome representations of GA and the free expansion of complex chromosomes of GP. GEP is a natural feature extractor that may not need a priori information of nonlinear bases other than the basic features as a terminal set. Generally, with enough computational time, GEP may recover unknown physical models that are represented by complex functional compositions by observing the input-response data.

In this work, we demonstrate that the sparse regression technique STRidge and the

evolutionary optimization algorithm GEP are effective SR tools for identifying hidden physical laws from observed data. We first identify various canonical PDEs using both STRidge and GEP. We demonstrate that STRidge is limited by its feature library for identifying the Sine-Gordon PDE. Following equation discovery, we demonstrate the power of both algorithms in identifying the leading truncation error terms for the Burgers MDE. While both algorithms find the truncation terms, coefficients found by STRidge were more accurate than coefficients found by GEP. We note that, when the feature library is capable of expressing the underlying physical model, the application of STRidge is suitable due to its fewer hyper-parameters and lower computational overhead. Next, we illustrate the recovery of hidden physics that is supplied as the source or forcing term of a PDE. We use randomly selected sparse measurements that mimic real world data collection. STRdige is not applied in this setting as the feature library was limited to represent the unknown physical model that consists of complex functional compositions. GEP was able to identify the source term for both 1D advection-diffusion PDE and 2D vortex-merger problem using sparse measurements. Finally, both STRdige and GEP were applied to discover the eddy viscosity kernel along with its ad-hoc modelling coefficient as a subgrid scale model for the LES equations simulating the 2D Kraichnan turbulence problem. This particular example demonstrates the capability of inverse modelling or parametric estimation for turbulence closure models using SR approaches.

7.2 Future Work

Major follow up research can be conducted taking into account the outcome of the current study. Some of them are as follows:

- Commonly used RANS/LES turbulence modelling is based on linear stress-strain relationship i.e, Boussinesq approximation which relates anisotropy a_{ij} of the

Reynolds stress τ_{ij} to mean strain rates.

$$\begin{aligned} \tau_{ij} &= \frac{2}{3}\rho k\delta_{ij} - a_{ij}, \\ a_{ij} &= 2\mu_t S_{ij}, \end{aligned} \tag{7.1}$$

Where ρ denotes density, k is turbulent kinetic energy and μ_t represents turbulent viscosity. The linear relation in Eq. 7.1 is known to questionable prediction for flows with separation, boundary layers over curved surfaces and flow over complex topologies. Pope (1975) proposed nonlinear eddy viscosity based on stress tensor decomposition where anisotropic stress tensor a_{ij} is linear combination of basis tensors and scalar invariants.

$$a_{ij}(S_{ij}, \Omega_{ij}) = \sum_{n=1}^{10} T_{ij}^{(n)} \alpha_n(I_1, \ldots, I_5), \tag{7.2}$$

in which the coefficients α_n are function of five invariants $I_1, \ldots, I_5$ and ten tensor basis $T_{ij}^1, T_{ij}^2, \ldots, T_{ij}^{10}$. The first four base tensors T_{ij}^n and two invariants I_m in Eq. 7.2 read,

$$\begin{aligned} T_{ij}^1 = S_{ij}, T_{ij}^2 &= S_{ij}\Omega_{kj}, \\ T_{ij}^3 &= S_{ik}S_{kj} - \frac{1}{3}\delta_{ij}S_{mn}S_{nm}, \\ T_{ij}^4 &= \Omega_{ik}\Omega_{kj} - \frac{1}{3}\delta_{ij}\Omega_{mn}\Omega_{nm}, \\ I_1 = S_{mn}S_{nm}, I_2 &= \Omega_{mn}\Omega_{nm}. \end{aligned} \tag{7.3}$$

where S_{ij} and Ω_{ij} in Eq. 7.3 are mean strain rate and mean rotation rate respectively. SR tools can be successfully used to find the anisotropic tensor given in Eq. 7.2 .

- Various SR tools can be exploited for the identification of nonlinear truncation error terms of MDEs for implicit LES approaches that can be exploited for modelling turbulent flows without the need for explicit subgrid scale models.

References

Adams, N., Hickel, S., and Franz, S. (2004). Implicit subgrid-scale modeling by adaptive deconvolution. *Journal of Computational Physics*, 200(2):412–431.

Arakawa, A. (1966). Computational design for long-term numerical integration of the equations of fluid motion: Two-dimensional incompressible flow. part i. *Journal of Computational Physics*, 1(1):119–143.

Aronson, D. G. and Weinberger, H. F. (1978). Multidimensional nonlinear diffusion arising in population genetics. *Advances in Mathematics*, 30(1):33–76.

Baldwin, B. and Lomax, H. (1978). Thin-layer approximation and algebraic model for separated turbulentflows. In *16th aerospace sciences meeting*, page 257. AIAA Meeting Paper.

Baraniuk, R. G. (2007). Compressive sensing. *IEEE Signal Processing Magazine*, 24(4):118–124.

Barone, A., Esposito, F., Magee, C., and Scott, A. (1971). Theory and applications of the sine-gordon equation. *La Rivista del Nuovo Cimento (1971-1977)*, 1(2):227–267.

Batchelor, G. K. (1969). Computation of the energy spectrum in homogeneous two-dimensional turbulence. *The Physics of Fluids*, 12(12):II–233.

Bateman, H. (1915). Some recent researches on the motion of fluids. *Monthly Weather Review*, 43(4):163–170.

Boffetta, G. and Ecke, R. E. (2012). Two-dimensional turbulence. *Annual Review of Fluid Mechanics*, 44:427–451.

Boffetta, G. and Musacchio, S. (2010). Evidence for the double cascade scenario in two-dimensional turbulence. *Physical Review E*, 82(1):016307.

Bongard, J. and Lipson, H. (2007). Automated reverse engineering of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 104(24):9943–9948.

Brameier, M. F. and Banzhaf, W. (2007). *Linear genetic programming*. Springe-Verlag, New York.

Brunton, S. L. and Noack, B. R. (2015). Closed-loop turbulence control: progress and challenges. *Applied Mechanics Reviews*, 67(5):050801.

Brunton, S. L., Proctor, J. L., and Kutz, J. N. (2016). Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937.

Cai, J.-F., Dong, B., Osher, S., and Shen, Z. (2012). Image restoration: total variation, wavelet frames, and beyond. *Journal of the American Mathematical Society*, 25(4):1033–1089.

Çanakcı, H., Baykasoğlu, A., and Güllü, H. (2009). Prediction of compressive and tensile strength of Gaziantep basalts via neural networks and gene expression programming. *Neural Computing and Applications*, 18(8):1031.

Candes, E. J., Romberg, J. K., and Tao, T. (2006). Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 59(8):1207–1223.

Candes, E. J. and Wakin, M. B. (2008). An introduction to compressive sampling. *IEEE Signal Processing Magazine*, 25(2):21–30.

Candes, E. J., Wakin, M. B., and Boyd, S. P. (2008). Enhancing sparsity by reweighted ℓ 1 minimization. *Journal of Fourier Analysis and Applications*, 14(5-6):877–905.

Chen, C., Luo, C., and Jiang, Z. (2017). Elite bases regression: a real-time algorithm for symbolic regression. In *2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, pages 529–535. IEEE.

Ciregan, D., Meier, U., and Schmidhuber, J. (2012). Multi-column deep neural networks for image classification. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3642–3649.

Cordier, L., Noack, B. R., Tissot, G., Lehnasch, G., Delville, J., Balajewicz, M., Daviller, G., and Niven, R. K. (2013). Identification strategies for model-based control. *Experiments in Fluids*, 54(8):1580.

Debien, A., Von Krbek, K. A., Mazellier, N., Duriez, T., Cordier, L., Noack, B. R., Abel, M. W., and Kourta, A. (2016). Closed-loop separation control over a sharp edge ramp using genetic programming. *Experiments in Fluids*, 57(3):40.

Dehghan, M. and Fakhar-Izadi, F. (2011). Pseudospectral methods for nagumo equation. *International Journal for Numerical Methods in Biomedical Engineering*, 27(4):553–561.

Dhingra, S., Madda, R. B., Gandomi, A. H., Patan, R., and Daneshmand, M. (2019). Internet of things mobile-air pollution monitoring system (IoT-Mobair). *IEEE Internet of Things Journal*, 6:5577 – 5584.

Dong, B., Jiang, Q., and Shen, Z. (2017). Image restoration: Wavelet frame shrinkage, nonlinear evolution PDEs, and beyond. *Multiscale Modeling & Simulation*, 15(1):606–660.

Duriez, T., Parezanović, V., von Krbek, K., Bonnet, J.-P., Cordier, L., Noack, B. R., Segond, M., Abel, M., Gautier, N., Aider, J.-L., et al. (2015). Feedback control of turbulent shear flows by genetic programming. *arXiv preprint arXiv:1505.01022*.

Faradonbeh, R. S. and Monjezi, M. (2017). Prediction and minimization of blast-induced ground vibration using two robust meta-heuristic algorithms. *Engineering with Computers*, 33(4):835–851.

Faradonbeh, R. S., Salimi, A., Monjezi, M., Ebrahimabadi, A., and Moormann, C. (2017). Roadheader performance prediction using genetic programming (GP) and gene expression programming (GEP) techniques. *Environmental Earth Sciences*, 76(16):584.

Ferariu, L. and Patelli, A. (2009). Elite based multiobjective genetic programming for nonlinear system identification. In *International Conference on Adaptive and Natural Computing Algorithms*, pages 233–242. Springer.

Ferreira, C. (2001). Gene expression programming: a new adaptive algorithm for solving problems. *arXiv preprint cs/0102027*.

Ferreira, C. (2002). Gene expression programming in problem solving. In *Soft Computing and Industry*, pages 635–653. Springer.

Ferreira, C. (2006). *Gene expression programming: mathematical modeling by an artificial intelligence*, volume 21. Springer.

Fortin, F.-A., De Rainville, F.-M., Gardner, M.-A., Parizeau, M., and Gagné, C. (2012). DEAP: Evolutionary algorithms made easy. *Journal of Machine Learning Research*, 13:2171–2175.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22.

Gautier, N., Aider, J.-L., Duriez, T., Noack, B., Segond, M., and Abel, M. (2015). Closed-loop separation control using machine learning. *Journal of Fluid Mechanics*, 770:442–457.

Gregorčič, G. and Lightbody, G. (2008). Nonlinear system identification: From multiple-model networks to Gaussian processes. *Engineering Applications of Artificial Intelligence*, 21(7):1035–1055.

Guvanasen, V. and Volker, R. (1983). Numerical solutions for solute transport in unconfined aquifers. *International Journal for Numerical Methods in Fluids*, 3(2):103–123.

Hirsch, C. (2007). *Numerical computation of internal and external flows: The fundamentals of computational fluid dynamics*. Elsevier, Burlington, MA.

Hirt, C. W. (1968). Heuristic stability theory for finite-difference equations. *Journal of Computational Physics*, 2(4):339–355.

Holland, J. H. (1992). Adaptation in natural and artificial systems. 1975. *Ann Arbor, MI: University of Michigan Press and*.

Hoseinian, F. S., Faradonbeh, R. S., Abdollahzadeh, A., Rezai, B., and Soltani-Mohammadi, S. (2017). Semi-autogenous mill power model development using gene expression programming. *Powder Technology*, 308:61–69.

Hunter, J. K. and Scheurle, J. (1988). Existence of perturbed solitary wave solutions to a model equation for water waves. *Physica D: Nonlinear Phenomena*, 32(2):253–268.

Isenberg, J. and Gutfinger, C. (1973). Heat transfer to a draining film. *International Journal of Heat and Mass Transfer*, 16(2):505–512.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An introduction to statistical learning*, volume 112. Springer Science+Business Media, New York.

Jordan, M. I. and Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260.

Karpathy, A. and Fei-Fei, L. (2015). Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3128–3137.

Kawahara, T. (1972). Oscillatory solitary waves in dispersive media. *Journal of the Physical Society of Japan*, 33(1):260–264.

Kawahara, T., Sugimoto, N., and Kakutani, T. (1975). Nonlinear interaction between short and long capillary-gravity waves. *Journal of the Physical Society of Japan*, 39(5):1379–1386.

Klopfer, G. and McRae, D. S. (1983). Nonlinear truncation error analysis of finite difference schemes forthe euler equations. *AIAA Journal*, 21(4):487–494.

Kocijan, J., Girard, A., Banko, B., and Murray-Smith, R. (2005). Dynamic systems identification with Gaussian processes. *Mathematical and Computer Modelling of Dynamical Systems*, 11(4):411–424.

Korteweg, D. J. and de Vries, G. (1895). On the change of form of long waves advancing in a rectangular canal, and on a new type of long stationary waves. *Philosophical Magazine*, 39(240):422–443.

Koza, J. R. (1992). *Genetic programming: on the programming of computers by means of natural selection*, volume 1. MIT Press, Cambridge, MA, USA.

Kraichnan, R. H. (1967). Inertial ranges in two-dimensional turbulence. *The Physics of Fluids*, 10(7):1417–1423.

Krasnopolsky, V. M. and Fox-Rabinovitz, M. S. (2006a). Complex hybrid models

combining deterministic and machine learning components for numerical climate modeling and weather prediction. *Neural Networks*, 19(2):122–134.

Krasnopolsky, V. M. and Fox-Rabinovitz, M. S. (2006b). A new synergetic paradigm in environmental numerical modeling: Hybrid models combining deterministic and machine learning components. *Ecological Modelling*, 191(1):5–18.

Kumar, N. (1983). Unsteady flow against dispersion in finite porous media. *Journal of Hydrology*, 63(3-4):345–358.

Lamb Jr, G. L. (1980). *Elements of soliton theory*. Wiley-Interscience, New York.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436.

Leith, C. (1971). Atmospheric predictability and two-dimensional turbulence. *Journal of the Atmospheric Sciences*, 28(2):145–161.

Leith, C. E. (1968). Diffusion approximation for two-dimensional turbulence. *The Physics of Fluids*, 11(3):671–672.

Loiseau, J.-C., Noack, B. R., and Brunton, S. L. (2018). Sparse reduced-order modelling: sensor-based dynamics to full-state estimation. *Journal of Fluid Mechanics*, 844:459–490.

Long, Z., Lu, Y., and Dong, B. (2019). PDE-Net 2.0: Learning PDEs from data with a numeric-symbolic hybrid deep network. *Journal of Computational Physics*, 399:108925.

Long, Z., Lu, Y., Ma, X., and Dong, B. (2018). PDE-net: Learning PDEs from data. In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3208–3216, Stockholmsmässan, Stockholm Sweden. PMLR.

Luo, C., Hu, Z., Zhang, S.-L., and Jiang, Z. (2015). Adaptive space transformation: An invariant based method for predicting aerodynamic coefficients of hypersonic vehicles. *Engineering Applications of Artificial Intelligence*, 46:93–103.

Luo, C. and Zhang, S.-L. (2012). Parse-matrix evolution for symbolic regression. *Engineering Applications of Artificial Intelligence*, 25(6):1182–1193.

Majda, A. and Osher, S. (1978). A systematic approach for correcting nonlinear instabilities. *Numerische Mathematik*, 30(4):429–452.

Maleewong, M. and Sirisup, S. (2011). On-line and Off-line POD assisted projective integral for non-linear problems: A case study with Burgers' equation. *International Journal of Mathematical, Computational, Physical, Electrical and Computer Engineering*, 5(7):984–992.

Mangan, N. M., Brunton, S. L., Proctor, J. L., and Kutz, J. N. (2016). Inferring biological networks by sparse identification of nonlinear dynamics. *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, 2(1):52–63.

Mangan, N. M., Kutz, J. N., Brunton, S. L., and Proctor, J. L. (2017). Model selection for dynamical systems via sparse regression and information criteria. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2204):20170009.

Margolin, L. G. and Rider, W. J. (2002). A rationale for implicit turbulence modelling. *International Journal for Numerical Methods in Fluids*, 39(9):821–841.

Marx, V. (2013). Biology: The big challenges of big data. *Nature*, 498:255–260.

Maulik, R., San, O., Jacob, J. D., and Crick, C. (2019a). Sub-grid scale model classification and blending through deep learning. *Journal of Fluid Mechanics*, 870:784–812.

Maulik, R., San, O., Rasheed, A., and Vedula, P. (2018). Data-driven deconvolution for large eddy simulations of kraichnan turbulence. *Physics of Fluids*, 30(12):125109.

Maulik, R., San, O., Rasheed, A., and Vedula, P. (2019b). Subgrid modelling for two-dimensional turbulence using neural networks. *Journal of Fluid Mechanics*, 858:122–144.

McConaghy, T. (2011). FFX: Fast, scalable, deterministic symbolic regression technology. In *Genetic Programming Theory and Practice IX*, pages 235–260. Springer.

Meneveau, C. and Katz, J. (2000). Scale-invariance and turbulence models for large-eddy simulation. *Annual Review of Fluid Mechanics*, 32(1):1–32.

Meunier, P., Le Dizès, S., and Leweke, T. (2005). Physics of vortex merging. *Comptes Rendus Physique*, 6(4-5):431–450.

Mitchell, M. (1998). *An introduction to genetic algorithms*. MIT press.

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT Press, Cambridge, MA, USA.

Nagumo, J., Arimoto, S., and Yoshizawa, S. (1962). An active pulse transmission line simulating nerve axon. *Proceedings of the IRE*, 50(10):2061–2070.

Ozbenli, E. and Vedula, P. (2017a). High order accurate finite difference schemes based on symmetry preservation. *Journal of Computational Physics*, 349:376–398.

Ozbenli, E. and Vedula, P. (2017b). Numerical solution of modified differential equations based on symmetry preservation. *Physical Review E*, 96(6):063304.

Ozis, T. and Ozer, S. (2006). A simple similarity-transformation-iterative scheme applied to Korteweg–de Vries equation. *Applied Mathematics and Computation*, 173(1):19–32.

Pawar, S. and San, O. (2019). CFD Julia: A learning module structuring an introductory course on computational fluid dynamics. *Fluids*, 4(3):159.

Perring, J. and Skyrme, T. (1962). A model unified field equation. *Nuclear Physics*, 31:550–555.

Piomelli, U. (1999). Large-eddy simulation: achievements and challenges. *Progress in Aerospace Sciences*, 35(4):335–362.

Pope, S. (1975). A more general effective-viscosity hypothesis. *Journal of Fluid Mechanics*, 72(2):331–340.

Quade, M., Abel, M., Shafi, K., Niven, R. K., and Noack, B. R. (2016). Prediction of dynamical systems by symbolic regression. *Physical Review E*, 94(1):012214.

Raissi, M. and Karniadakis, G. E. (2018). Hidden physics models: Machine learning of nonlinear partial differential equations. *Journal of Computational Physics*, 357:125–141.

Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2018). Numerical gaussian processes for time-dependent and nonlinear partial differential equations. *SIAM Journal on Scientific Computing*, 40(1):A172–A198.

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

Rauhut, H. (2010). *Compressive sensing and structured random matrices*, volume 9. Walter de Gruyter GmbH & Co. KG, Berlin.

Reinaud, J. N. and Dritschel, D. G. (2005). The critical merger distance between two co-rotating quasi-geostrophic vortices. *Journal of Fluid Mechanics*, 522:357–381.

Ritchmyer, R. D. and Norton, K. (1967). *Difference methods for initial value problems*. Jonh Wiley & Sons, New York.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386.

Rudy, S. H., Brunton, S. L., Proctor, J. L., and Kutz, J. N. (2017). Data-driven discovery of partial differential equations. *Science Advances*, 3(4):e1602614.

Sagaut, P. (2006). *Large eddy simulation for incompressible flows: an introduction*. Springer Science & Business Media.

Sallab, A. E., Abdou, M., Perot, E., and Yogamani, S. (2017). Deep reinforcement learning framework for autonomous driving. *Electronic Imaging*, 2017(19):70–76.

San, O. and Staples, A. E. (2012). High-order methods for decaying two-dimensional homogeneous isotropic turbulence. *Computers & Fluids*, 63:105–127.

San, O. and Staples, A. E. (2013). A coarse-grid projection method for accelerating incompressible flow computations. *Journal of Computational Physics*, 233:480–508.

Schaeffer, H. (2017). Learning partial differential equations via data discovery and sparse optimization. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2197):20160446.

Schaeffer, H., Caflisch, R., Hauck, C. D., and Osher, S. (2013). Sparse dynamics for partial differential equations. *Proceedings of the National Academy of Sciences*, 110(17):6634–6639.

Schaeffer, H., Tran, G., and Ward, R. (2018). Extracting sparse high-dimensional dynamics from limited data. *SIAM Journal on Applied Mathematics*, 78(6):3279–3295.

Schmelzer, M., Dwight, R., and Cinnella, P. (2018). Data-driven deterministic symbolic regression of nonlinear stress-strain relation for rans turbulence modelling. In *2018 Fluid Dynamics Conference*, page 2900. AIAA Aviation Forum.

Schmelzer, M., Dwight, R. P., and Cinnella, P. (2019). Machine learning of algebraic stress models using deterministic symbolic regression. *arXiv preprint arXiv:1905.07510*.

Schmidt, M. and Lipson, H. (2009). Distilling free-form natural laws from experimental data. *Science*, 324(5923):81–85.

Schoepplein, M., Weatheritt, J., Sandberg, R., Talei, M., and Klein, M. (2018). Application of an evolutionary algorithm to les modelling of turbulent transport in premixed flames. *Journal of Computational Physics*, 374:1166–1179.

Scott, A. (1963). Neuristor propagation on a tunnel diode loaded transmission line. *Proceedings of the IEEE*, 51(1):240–240.

Shuhua, G. (2019). geppy: a gene expression programming framework in python. `https://github.com/ShuhuaGao/geppy`.

Sirendaoreji (2004). New exact travelling wave solutions for the Kawahara and modified Kawahara equations. *Chaos Solitons & Fractals*, 19(1):147–150.

Smagorinsky, J. (1963). General circulation experiments with the primitive equations: I. the basic experiment. *Monthly Weather Review*, 91(3):99–164.

Smith, A. and Cebeci, T. (1967). Numerical solution of the turbulent-boundary-layer equations. Technical Report DAC 33735, DTIC.

Thaler, S., Paehler, L., and Adams, N. A. (2019). Sparse identification of truncation errors. *Journal of Computational Physics*, 397:108851.

Tibshirani, R. (1996). Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society: Series B*, 58(1):267–288.

Tibshirani, R., Wainwright, M., and Hastie, T. (2015). *Statistical learning with sparsity: the LASSO and generalizations*. Chapman and Hall/CRC, Florida, USA.

Tran, G. and Ward, R. (2017). Exact recovery of chaotic systems from highly corrupted data. *Multiscale Modeling & Simulation*, 15(3):1108–1129.

Vaddireddy, H., Rasheed, A., Staples, A. E., and San, O. (2020). Feature engineering and symbolic regression methods for detecting hidden physics from sparse sensor observation data. *Physics of Fluids*, 32(1):015113.

Vaddireddy, H. and San, O. (2019). Equation discovery using fast function extraction: a deterministic symbolic regression approach. *Fluids*, 4(2):111.

Wang, Z., Xiao, D., Fang, F., Govindan, R., Pain, C. C., and Guo, Y. (2018). Model identification of reduced order fluid dynamics systems using deep learning. *International Journal for Numerical Methods in Fluids*, 86(4):255–268.

Wazzan, L. (2009). A modified tanh–coth method for solving the KdV and the KdV–Burgers' equations. *Communications in Nonlinear Science and Numerical Simulation*, 14(2):443–450.

Weatheritt, J. and Sandberg, R. (2016). A novel evolutionary algorithm applied to algebraic modifications of the rans stress–strain relationship. *Journal of Computational Physics*, 325:22–37.

Weatheritt, J. and Sandberg, R. D. (2017). Hybrid reynolds-averaged/large-eddy simulation methodology from symbolic regression: formulation and application. *AIAA Journal*, pages 5577 – 5584.

Whitham, G. B. (2011). *Linear and nonlinear waves*, volume 42. John Wiley & Sons.

Worm, T. and Chiu, K. (2013). Prioritized grammar enumeration: symbolic regression by dynamic programming. In *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*, pages 1021–1028. ACM.

Yang, Y., Wang, C., and Soh, C. (2005). Force identification of dynamic systems using genetic programming. *International Journal for Numerical Methods in Engineering*, 63(9):1288–1312.

Zhang, T. (2009). Adaptive forward-backward greedy algorithm for sparse learning with linear models. In *Advances in Neural Information Processing Systems*, pages 1921–1928. Neural Information Processing Systems Foundation, Inc.

Zheng, P., Askham, T., Brunton, S. L., Kutz, J. N., and Aravkin, A. Y. (2018). A unified framework for sparse relaxed regularized regression: SR3. *IEEE Access*, 7:1404–1423.

Zhi-Xiong, C. and Ben-Yu, G. (1992). Analytic solutions of the nagumo equation. *IMA Journal of Applied Mathematics*, 48(2):107–115.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320.

VITA

Harsha Vaddireddy

Candidate for the Degree of

Master of Science

Thesis: IDENTIFICATION OF PHYSICAL PROCESSES VIA DATA DRIVEN METHODS

Major Field: Mechanical & Aerospace Engineering

Biographical:

Education:

Completed the requirements for the degree of Master of Science with a major in Mechanical & Aerospace Engineering at Oklahoma State University in May 2020.

Completed Masters of Technology in Aerodynamics and Flight Mechanics at Indian Institute of Space Science and Technology, Trivandrum, India in June 2016.

Completed Bachelors of Engineering in Aerospace Engineering at Hindustan University, Chennai, India in June 2014.

Experience:

Project Assitant-3, Computational and Theoretical Fluid Dynamics division, National Aerospace Laboratories, Bangalore, India from Dec 2016-Jun-2018.