A BAYESIAN APPROACH TO DATA-DRIVEN DISCOVERY OF NONLINEAR DYNAMIC EQUATIONS

A Dissertation presented to
the Faculty of the Graduate School
at the University of Missouri

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

by

JOSHUA S. NORTH

Drs. Christopher K. Wikle and Erin M. Schliep, Dissertation Supervisors

JULY 2022

The undersigned, appointed by the Dean of the Graduate School, have examined the dissertation entitled:

A BAYESIAN APPROACH TO DATA-DRIVEN DISCOVERY OF NONLINEAR DYNAMIC EQUATIONS

presented by Joshua S. North,

a candidate for the degree of Doctor of Philosophy and hereby certify that, in their opinion, it is worthy of acceptance.

Dr. Christopher K. Wikle

Dr. Erin M. Schliep

Dr. Scott H. Holan

Dr. Shih-Kang Chao

Dr. Neil I. Fox

ACKNOWLEDGMENTS

First, I would like to thank my advisors (in no particular order) Drs. Christopher Wikle and Erin Schliep for their constant support, insightful answers, guidance, patience, and passion for statistics. I would also like to thank my committee, Drs. Scott Holan, Shih-Kang Chao, and Neil Fox, for their alternate views, ideas, and knowledge of statistics and atmospheric sciences. To Dr. William Kleiber, whose enjoyment for statistics first inspired me and who suggested I attend Mizzou for graduate school - a very special thank you. To all of my teachers, professors, and instructors along the way, thank you.

I have made some amazing friends during my graduate career and they have made my time at Mizzou and in Columbia special. I have enjoyed my time studying, discussing, relaxing, and hanging out with you and wish you all the best! To my family. I am forever grateful to your love, support, and words of encouragement. And last, but by no means least, my wife Mariah. Your constant support and encouragement got me through the hard days; our meandering walks gave me much needed mental breaks; your love helped me get to the end. Thank you all.

TABLE OF CONTENTS

APPENDIX

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

A Bayesian Approach to Data-Driven Discovery of Nonlinear Dynamic Equations

Joshua S. North

Christopher K. Wikle and Erin M. Schliep, Dissertation Supervisors

ABSTRACT

Dynamic equations parameterized by differential equations are used to represent a variety of real-world processes. The equations used to describe these processes are generally derived based on physical principles and a scientific understanding of the process. Statisticians have embedded these physically-inspired differential equations into a probabilistic framework, providing uncertainty quantification to parameter estimates and model specification. These statistical models typically rely on a predefined differential equation or class of models to represent the dynamics of the system. Recently, methods have been developed to discover the governing equation of complex systems. However, these approaches rarely account for uncertainty in the discovered equations, and when uncertainty is accounted for, it is not for the complete system. This dissertation begins with a statistical model for the seasonal temperature cycle over North America, where the dynamics of the system are parameterized by a specified functional form. The model highlights how the seasonal cycle is changing in space and time, motivating the need to better understand the driving mechanisms of such systems. Then, a statistical approach to data-driven discovery is proposed, where uncertainty is incorporated throughout the complete modeling process. The novelty of the approach is the dynamics are treated as a random process, which has not be considered previously in the data-driven discovery literature. The proposed approach sits at the junction between the statistical approach of incorporating dynamic equations in a probabilistic framework and the data-driven discovery methods proposed in computer science, physics, and applied mathematics. The proposed method is put into context within the broader literature, highlighting its contribution to the field of data-driven discovery.

Chapter 1

Introduction

A reoccurring theme at the intersection of statistical and atmospheric sciences is to provide a more complete understanding to our physical world through scientifically motivated models. Pivotal in that understanding is the quantification of the inherent uncertainty in meteorological processes stemming from our inability to perfectly characterize a process. To complicate the issue, the Earth's climate is changing asynchronously across space and time, altering the manner in which atmospheric systems interact and our predisposition on how the systems are characterized. This necessitates methods that can characterize how complex systems interact and discover the mechanisms driving these systems.

One such system where the mechanisms driving the system are unknown is the seasonal temperature cycle, which is often characterized through harmonic components. That is, the temperature cycle can be decomposed in terms of its annual (e.g., one cycle per year) and semi-annual (e.g., two cycles per year) harmonic components. Higher-order harmonic terms could be included, but the annual and semi-annual harmonic components have been shown to explain the majority of the variation in the temperature cycle. While the annual harmonic can be attributed to Earth's orbital path around the sun, the importance of the semi-annual harmonic is not as obvious. In the Southern latitudes, the semi-annual cycle has been shown to result from differential heating in north-south land-sea contrasts between Antarctica and the Southern Ocean (van Loon, 1967), while the semi-annual cycle in the Northern Hemisphere has been shown to be related to an east-west land-sea contrast (Wikle and Chen, 1996). It is known that the land-sea contrast plays a role in the semi-annual cycle, and it is hypothesized that anthropogenic climate change will lead to asynchronous changes in the semi-annual cycle because the land and ocean have different responses to greenhouse-related heating. However, the specific mechanisms of this changing system and how to properly characterize them remain unknown.

In Chapter 2, we propose a model to jointly quantify the minimum and maximum temperature cycles parameterized by their annual and semi-annual harmonic components. The model enables the identification of spatial and temporal changes in the seasonal temperature cycle, capturing spatial dependence, temporal dynamics, and multivariate dependence of these harmonics through spatially and temporally varying coefficients. The model is applied to minimum and maximum temperature over North America from 1979-2018, with regions experiencing significant shifts in the temperature cycle being highlighted through changes in the two harmonics. Our results provide further insight into how the temperature cycles are shifting when parameterized functionally using harmonics and the importance the semi-annual harmonic has on the seasonal temperature cycle.

The seasonal cycle analysis from Chapter 2 suggests potential hypotheses that may govern the dynamics of this changing process. Current deterministic coupled global ocean-atmosphere models can mimic this behavior and provide projections of how it may change in the future. However, such models are nearly as complex as the real-world system and the specific dynamic mechanisms that are responsible for the asymmetric response of the seasonal cycle to climate change are unclear. This suggests the need to discover the specific dynamic mechanisms in the presence of uncertain observations. Recently, a body of literature has surfaced within the fields of computer science, physics, and applied mathematics that aims to discover the governing equations driving nonlinear systems, generally termed *data-driven discovery of nonlinear dynamic equations*. We provide a substantive literature review of these methods in Chapter 3, highlighting the strengths, weaknesses, similarities, and differences of various approaches and provide parallels to classical statistical modeling of dynamic systems. The general idea behind data-driven discovery is to relate the time derivative of a system to a function of the system. The problem can be parameterized as an ordinary or partial differential equation (ODE/PDE) where the left hand side of the equation is a temporal derivative and the right hand side is some nonlinear function of the system (e.g., interactions, space and/or time derivatives, etc.).

The difficulty of data-driven discovery in this modeling framework is computing derivatives in space or time or both, especially with observation uncertainty. The computed derivatives are used to determine the dynamics of the system, where proper recovery of the derivatives is pivotal to the method being able to discover the dynamics. In Chapter 3, we provide a comprehensive discussion of the numerous methods by which this process can be done. However, these various methods each have some crucial short-comings. These methods are rarely applied to real-world problems, favoring simulations by which to exhibit their contributions to the literature. While there is little to no discussion as to why the methods are rarely applied to real-world systems, it is most likely due to the difficulty real-world data pose. For example, in the simulations, the true equation is known and it is easy to assess model discovery accuracy. In a real-world system, we must rely on the scientific literature to provide a hypothesis as to the correct equation, but there is no guarantee the "true" dynamics abide by these hypotheses. Additionally, the ability to handle observation uncertainty in the form of missing data or measurement noise is also a limitation of these approaches.

To account for the inherent uncertainty in real-world data, a statistical framework to data-driven discovery of nonlinear dynamics can be used. As we will discuss in Chapter 3, and exemplified by Chapters 4 and 5, statistical methods for data-driven discovery provide a framework that can accommodate data imperfections and be applied to a wider range of problems. However, few methods for data-driven discovery consider uncertainty quantification, and those that do treat the systems as true realizations as opposed to random processes. Specifically, these methods rely on denoising the data and computing derivatives *a prior*, treating these derived values as the fixed and known observation. By not properly accounting for the randomness of the observed process, these methods disregard observational uncertainty (missing data and measurement noise). In turn, this means the estimated uncertainty in these approaches is dependent on the method(s) used to denoise and differentiate the data and fail to properly represent the uncertainty of the system.

In Chapters 4 and 5 we will develop a statistical method for data-driven discovery of nonlinear dynamic equations that considers the dynamic system a random process, enabling uncertainty quantification throughout the entire process. Chapter 4 develops a Bayesian approach for data-driven discovery of ordinary differential equations (ODEs), bringing the current work of data-driven discovery to the statistical sciences. In contrast to the prior methods of data-driven discovery with uncertainty, we treat the system as a random process within a Bayesian hierarchical model (BHM) where the dynamic process is modeled as a latent process that is represented using a basis function expansion. Using the BHM framework, where the dynamics are latent and the system is a random process, results in our approach being able to accommodate missing data, properly account for measurement uncertainty, and provide proper uncertainty quantification to the system. To illustrate the advantage of the statistical approach to data-driven discovery, we show the robustness of the model to observational noise and missing data on multiple simulated datasets. We then apply the method to three real-world problems - the historic Hare-Lynx predator-prey system, a motion tracked pendulum, and Pacific sea surface temperature.

In Chapter 5 we extend the model from Chapter 4 to space-time processes and propose a Bayesian approach for data-driven discovery of *partial* differential equations (PDEs). The addition of the spatial dimension requires that the problem be reformulated using higher-order tensors, where the dynamic process is represented as a higher-order basis function expansion. Different from the other approaches used to discover PDEs, our approach again models the system as a random process and can accommodate missing data. In addition, the framework can accommodate systems where the response is dependent on temporal and/or spatio-temporal derivatives of the system. The applicability of the proposed methodology is illustrated on three simulated systems with varying amounts of observational uncertainty and missing data. The method is also applied to a real-world system, where we infer how the vorticity of the streamfunction evolves over time.

The dissertation is concluded in Chapter 6. Specifically, we provide a quick summary and four potential ways to extend and improve on the work presented here.

Chapter 2

On the spatial and temporal shift in the archetypal seasonal temperature cycle as driven by annual and semi-annual harmonics

2.1 Introduction

In ecology, “spatial synchrony” is a concept that describes coincident in time variations in an ecological process across geographically separated populations (Liebhold et al., 2004). In many cases, this synchrony leads to symbiotic relationships that are tied to environmental seasonal cycles. For example, an important climate-driven issue facing forest health concerns native bark beetle infestations, with current beetle outbreaks among the most severe in recorded history (Bentz et al., 2010). Historically, exposure to very cold temperatures is necessary to control the beetle population, but increasing seasonal minimum temperatures in northern latitudes has disrupted this historical synchrony, limiting beetle mortality, and increasing tree mortality. Other examples of spatial synchrony being disrupted by changes in environmental seasonal cycles include ocean primary productivity (Defriez et al., 2016), lake stratification (Kraemer et al., 2015), migration patterns (Usui et al., 2017), and flood hazards (Arnell and Gosling, 2016), among others. The broad extent of these impacts highlight the importance of understanding how spatial patterns in environmental seasonal cycles vary through time.

The seasonal cycle in atmospheric variables is a direct response to the variation in solar insolation due to the Earth's orbital path around the Sun. Specifically, the atmospheric response to the overhead Sun crossing the equator twice a year suggests a more complicated seasonal variation that includes both an annual and a semi-annual harmonic. The annual (first) harmonic is a sinusoid that has one cycle per year and the semi-annual (second) harmonic is a sinusoid that has two cycles per year (see Eqn. 2.1). Harmonic analysis has been used by meteorologists and climatologists to characterize the connection between these harmonics and the observed seasonal cycle since the early 20th century (e.g., see Hsu and Wallace (1976a,b) for a review of this early work). Although the semi-annual harmonic typically contributes less variance to the seasonal cycle than the annual harmonic in the Northern mid-latitudes, its amplitude and phase vary considerably across space, and there are regions in which it can significantly affect the seasonal cycle (e.g., shifting the phase, strengthening the peak, flattening the minimum; see White and Wallace (1978)). One of the first studies to discuss a specific dynamical mechanism behind the semi-annual cycle was van Loon (1967), in which he showed that the semi-annual cycle in the high Southern latitudes was the result of differential heating due to north–south land/sea contrasts between Antarctica and the Southern ocean. Unlike the north–south contrast exhibited in high latitudes, Wikle and Chen (1996) showed evidence that the Northern hemisphere extratropical semi-annual cycle exhibits a strong east–west structure and is governed by the spatio-temporal asymmetries in the seasonal variation of the northern hemisphere stationary eddies (e.g., the wave structure in the atmospheric circulation). Because this variation in stationary eddies is due to east–west differential heating from land/sea contrasts, the fundamental mechanism for both the extra-tropical and high-latitude semi-annual cycles is due to land–sea contrasts and the impact this has on the atmospheric circulation.

It is well-known that atmospheric circulation patterns are varying due to differing responses of land and sea to climate forcing (e.g., Sutton et al. (2007)). This suggests that the annual and semi-annual harmonics are also likely varying. Stine et al. (2009) investigated the change in the annual harmonic component of surface temperature between the years 1900-1953 and 1954-2007 by looking at the lag (difference between temperature and local solar insolation phases) and gain (ratio of temperature and insolation amplitudes). Based on simple t-test comparisons of the lag and gain for the different time periods, they showed that the annual temperature cycle has changed, but asymmetrically across space. Dwyer et al. (2012) also showed that there has been heterogeneous variation in the annual amplitude and phase of the mean surface temperature cycle in response to greenhouse gases. These analyses did not explicitly consider the semi-annual component of the seasonal cycle, multivariate seasonal variation of atmospheric variables, nor did they consider a formal model-based uncertainty quantification framework that could accommodate spatial and temporal variation of the harmonics.

Dynamic spatio-temporal models (DSTMs) are well established in the literature for modeling complex spatial processes that evolve over time (see Cressie and Wikle (2011) for a collection of references and methods). Statistical DSTMs are able to capture spatial and temporal dependence in the process across different scales, while retaining the ability to capture uncertainty in parameter estimation and prediction. Surface temperatures over land generally can be decomposed into three components: a term to account for trend, a seasonal component, and a "weather" component. We would expect the first two of these components to vary somewhat slowly across time, with near-by locations experiencing similar temperatures and temperature variation through time, whereas the weather component corresponds to a dynamic process observed at finer spatial and temporal scales (Wikle et al., 1998). The most natural way to accommodate slowly varying time variation in trend and seasonal parameters is via the dynamic linear model (DLM) paradigm (e.g., West and Harrison (2006)). Such models are commonly extended to the dynamic evolution of parameters in spatio-temporal and multivariate settings by representing the parameters as spatial fields that vary in time according to a DLM (e.g., see the overviews in Gelfand et al. (2010); Cressie and Wikle (2011); Banerjee et al. (2014); Gelfand et al. (2017)).

The main modeling contribution of this work is the development of a joint statistical framework for time-varying minimum and maximum temperature cycles, which are specified through the annual and semi-annual harmonics, while accounting for spatial and temporal dependence. This model is motivated by the fact that responses to changes in heating are asymmetric in space; thus, we expect that the annual and semi-annual harmonics in temperature are varying in time differently across space, leading to time-varying differences in seasonal cycles. By adopting a Bayesian framework for parameter estimation for the associated DSTM model, we are able to quantify the extent to which regions across North America are experiencing significant shifts in the minimum and maximum temperature cycles. Significant asymmetric shifts in minimum and maximum temperature seasonal cycles may seriously affect biological processes that are synchronously linked to such cycles.

The remainder of this Chapter is structured as follows. In Section 2.2 we describe the data used in the analysis and provide a brief exploratory analysis to motivate the work. Section 2.3 details the joint model specification using the annual and semi-annual harmonics and methods for model inference. Section 2.4 presents the findings of the analysis and Section 2.5 provides a discussion and directions for future work.

2.2 Data and Preliminary Analysis

For the analyses presented here, we consider air temperature (deg C) data at two meters above the surface obtained from the National Center for Environmental Prediction (NCEP) Reanalysis[1]. The data are available at three hour intervals for each day from January 1, 1979 to December 31, 2018, which we summarize as daily minimum and maximum temperature. Since daily minimum and maximum temperature are products of the diurnal cycle, not extremes in the context of block maximum (Cooley and Sain, 2010) or exceedances over threshold (Chavez-Demoulin and Davison, 2005), we do not consider them within the extreme value theory framework. The data are on a 349×277 Northern Lambert Conformal Conic grid, with corners at approximately (1.000N, 145.500W), (0.898N, 68.320W), (46.354N, 2.570W), and (46.634N, 148.642E). All data exploration was conducted on a reduced spatial domain with corners at approximately (16.103N, 140.543W), (15.997N, 73.229W), (57.601N, 22.274W), and (57.856N, 168.499E).

Let z_t denote temperature (say, minimum or maximum) on day t where $t = 1, \ldots, T$, and T is the number of days in the year (365 or 366 for leap years). The discrete Fourier

[1] https://www.esrl.noaa.gov/psd/

series representation of the time series, expressed in amplitude-phase form, is given as

$$z_t = a_0 + \sum_{h=1}^{\lfloor T/2 \rfloor} A_h \cos\left(\frac{2\pi ht}{T} + \varphi_h\right), \tag{2.1}$$

where A_h and φ_h are the amplitude and phase, respectively, for the h^{th} harmonic component, and $\lfloor \ \rfloor$ is the "floor function" that returns the largest integer value of the argument. Reparameterizing Eqn. 2.1 in terms of its Fourier coefficients results in

$$z_t = a_0 + \sum_{h=1}^{\lfloor T/2 \rfloor} a_h \cos\left(\frac{2\pi th}{T}\right) + b_h \sin\left(\frac{2\pi th}{T}\right), \tag{2.2}$$

where a_h and b_h are the Fourier coefficients for the h^{th} harmonic, related to the amplitude by $A_h = \sqrt{a_h^2 + b_h^2}$, $A_h \in [0, \infty)$, and the phase by $\varphi_h = \tan^{-1}(-b_h/a_h)$, $\varphi_h \in [-\pi/h, \pi/h]$. As discussed in the Section 2.1, higher order harmonics lack a clear physical interpretation, so we restrict our estimation to the first two harmonics (i.e., $h = 1, 2$, where $h = 1$ and $h = 2$ correspond to the annual and semi-annual harmonic, respectively), and refer to a "cycle" as the sum of the first and second harmonics hereafter.

To investigate the possible relative importance of the semi-annual harmonic in North American minimum and maximum temperature cycles, Figure 2.1 shows the estimated seasonal cycles for minimum temperature when both the annual and semi-annual harmonic components are included (solid) compared to those in which only the annual component is considered (dashed). These estimated cycles are shown for two different years, 1979 and 1999, and two different locations, one in central Texas and the other in Kings Canyon National Park, California. The top panel shows the estimated cycle for the year 1979, the middle panel for the year 1999, and the bottom panel shows the difference between the two cycles, with the estimates from the 1979 cycle subtracted from the 1999 cycle. These plots

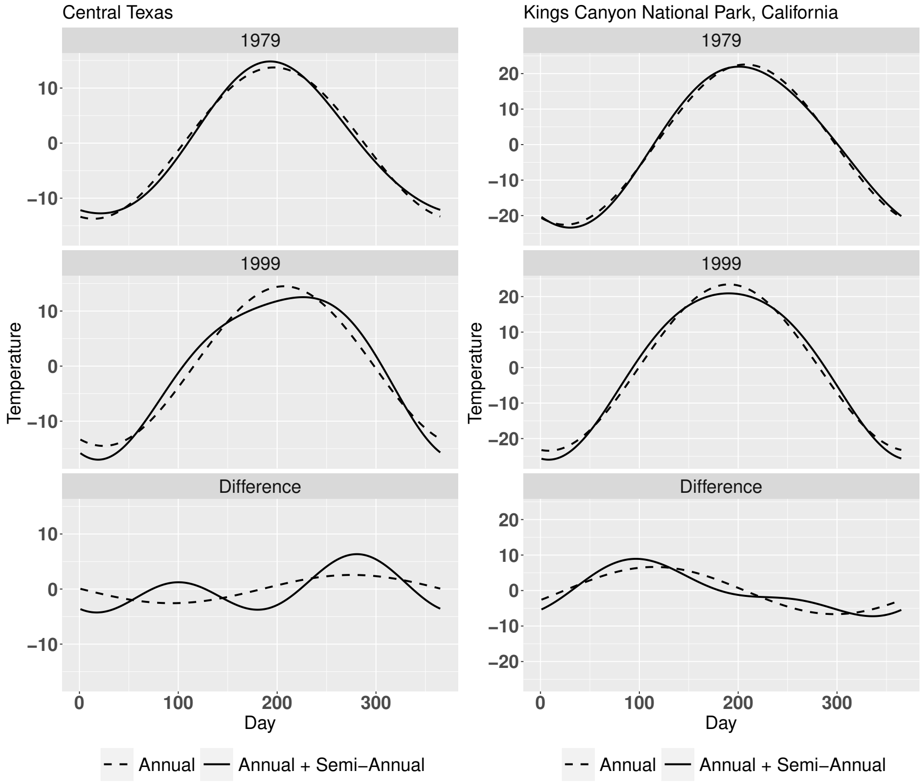


Figure 2.1: Comparison of estimated seasonal cycle for minimum temperature using the first and second Fourier harmonics (solid line) compared to just the first Fourier harmonic (dashed line) in central Texas and Kings Canyon National Park, California. The top plot is the estimated cycle for 1979, the middle for 1999, and the bottom showing the difference (1999 year - 1979 year).

illustrate the impact the semi-annual component can have on the temperature cycle, how the impact changes through time, and how these features vary across space. The estimated cycles for 1979 are very similar for both locations, suggesting the semi-annual harmonic had little influence on the minimum temperature cycle at these locations. Conversely, the temperature cycles for 1999 are much more dissonant for both locations, implying the semi-annual harmonic had a greater impact on the temperature cycle in 1999 than in 1979. The impact of the semi-annual harmonic of minimum temperature can be seen clearly in the bottom panel, where the difference between the estimated cycles between the two years

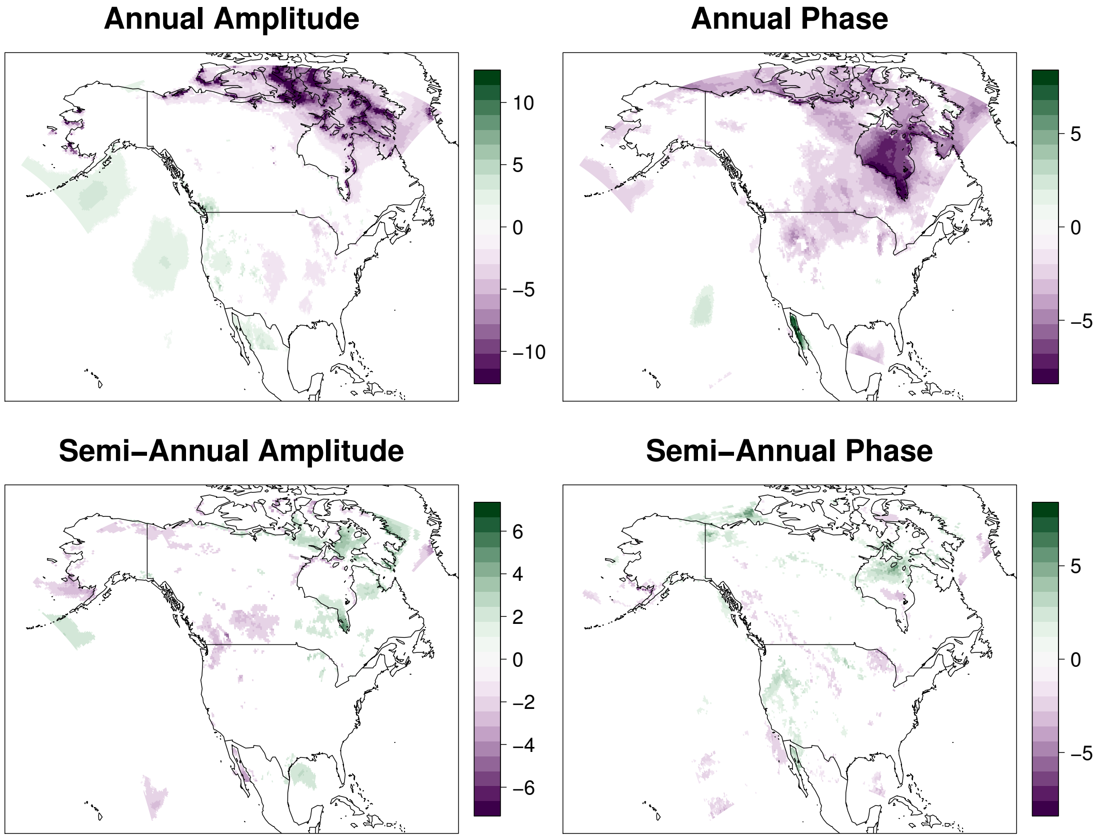


Figure 2.2: Computed t-statistics of the Fourier coefficients for the minimum temperature cycle. Locations with 95% point-wise confidence intervals not including 0 are shown, with the color corresponding to the t-statistic value.

using both annual and semi-annual components in central Texas shows a cyclical deviation from the difference in cycle estimates using only the annual component. At Kings Canyon, the cycle with the annual and semi-annual components oscillates about the difference using only the annual component. This suggests that the impact of the semi-annual component is spatially and temporally varying, and is important in capturing shifts in the temperature cycle through time.

To identify differential change in space through time in the annual and semi-annual harmonics, we investigate daily temperature data for each location for two separated 15-

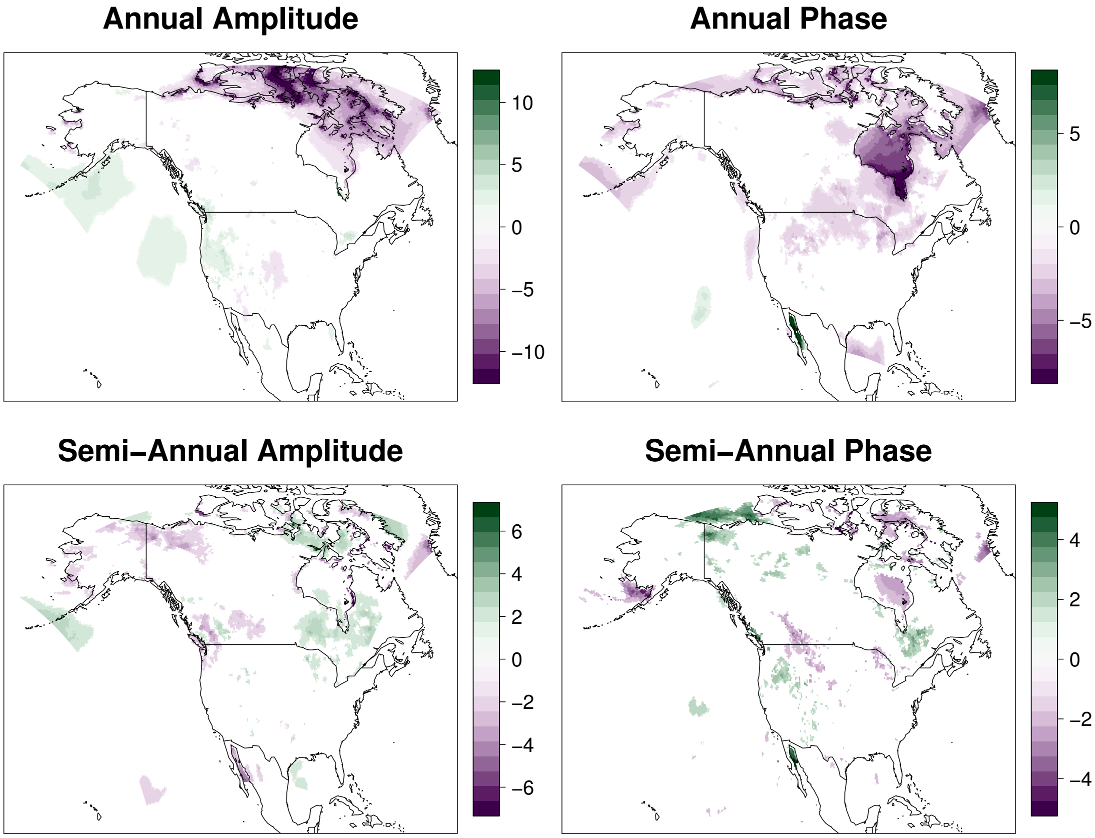


Figure 2.3: Computed t-statistics of the Fourier coefficients for the maximum temperature cycle. Locations with 95% point-wise confidence intervals not including 0 are shown, with the color corresponding to the t-statistic value.

year time periods; period one from 1979 - 1994 and period two from 2003 - 2018. From the annual and semi-annual harmonic estimates, we calculate the phase and amplitude for each year for each of the two time periods. As an exploratory comparison of the two periods at each location, we compute t-statistics of the difference (second period - first period) for both the annual and semi-annual phase and amplitude. Figures 2.2 and 2.3 show the t-statistics over the region for all four components (A_1, φ_1, A_2, φ_2) for minimum and maximum temperature, respectively. T-statistics that are large (in magnitude) suggest possible shifts in temperature cycles. For example, in both the minimum and maximum temperature annual phase, a large area in the northeast of the continent has experienced a negative shift, implying the peak of the wave is occurring earlier in the recent years. Additionally, the shift in the annual amplitude for both the minimum and maximum cycles are alike, with areas over the Pacific, central United States, and northern Canada having similar spatial patterns and values. The shifts in the semi-annual amplitude and phase of both cycles have similar spatial patterns, with areas in the northwest United States and northern Canada most closely resembling each other. These results for the annual amplitude and phase closely resemble the results over North America reported in Stine et al. (2009), however our findings are purely exploratory as they do not account for any spatial, temporal, or process dependence, which could have important effects on reported regions of significant change.

These preliminary analyses identified possible shifts in the annual and semi-annual harmonic components of minimum and maximum temperature over North America, which suggest changes in the cycles themselves, and that these changes might vary considerably across space. Our aim is to obtain full probabilistic inference with regard to these changes in minimum and maximum temperature cycles across the region. Specifically, we propose a multivariate statistical DSTM that captures the relationship between the minimum and maximum temperature cycles, as well as spatial and temporal dependence. This model will be used to quantify the uncertainty associated with potential seasonal cycle changes over space and time.

2.3 Bayesian Hierarchical Formulation

2.3.1 Data model

Let $\mathbf{z}_{1\ell}(\mathbf{s}) = [z_{11}(\mathbf{s}),...,z_{1T_\ell}(\mathbf{s})]'$ and $\mathbf{z}_{2\ell}(\mathbf{s}) = [z_{21}(\mathbf{s}),...,z_{2T_\ell}(\mathbf{s})]'$ denote the centered, by year, minimum and maximum temperature, respectively, for year ℓ, $\ell = 1,\ldots,L$ at location $\mathbf{s}$, $\mathbf{s} \in \{\mathbf{s}_1,...,\mathbf{s}_n\}$ where T_ℓ is the number of days in year ℓ. The linear model for temperature at location $\mathbf{s}$, year ℓ, and variable j can be specified in terms of the Fourier coefficients (e.g., (2.2)) by

$$\mathbf{z}_{j\ell}(\mathbf{s}) = \mathbf{X}_\ell \widetilde{\boldsymbol{\beta}}_{j\ell}(\mathbf{s}) + \widetilde{\boldsymbol{\epsilon}}_{j\ell}(\mathbf{s}) \qquad j = 1,2, \tag{2.3}$$

where $\mathbf{X}_\ell = [\boldsymbol{\rho}_1, \boldsymbol{\psi}_1, \boldsymbol{\rho}_2, \boldsymbol{\psi}_2]$, and $\widetilde{\boldsymbol{\beta}}_{j\ell}(\mathbf{s}) = [a_1(\mathbf{s}), b_1(\mathbf{s}), a_2(\mathbf{s}), b_2(\mathbf{s})]'$, with the t^{th} element of $\boldsymbol{\rho}_h$ and $\boldsymbol{\psi}_h$ equal to $\rho_{ht} = \cos(2\pi h(t-1)/T_\ell)$ and $\psi_{ht} = \sin(2\pi h(t-1)/T_\ell)$, respectively, for $h = 1,2$, and $\widetilde{\boldsymbol{\epsilon}}_{j\ell}(\mathbf{s}) \overset{iid}{\sim} N(0, \widetilde{\sigma}^2_{\varepsilon_j}(\mathbf{s})\mathbf{I}_{T_\ell})$. Here, $\widetilde{\sigma}^2_{\varepsilon_j}(\mathbf{s})$ is the variance for the j^{th} variable at location $\mathbf{s}$, which is assumed constant over years for each cycle and location. Although the simplifying assumption of *i.i.d.* errors assumes no residual temporal autocorrelation as would be present in "weather" processes, preliminary analyses that accounted for this correlation through a daily random effect did not substantially impact parameter inference. As such, the model with daily random effects was not considered further due to the added computational complexity.

Spatial and temporal dependence is modeled using spatially-varying harmonic coefficients and with a random walk time structure. Specifically, the harmonic coefficients, $\widetilde{\boldsymbol{\beta}}_\ell$, are spatial processes and the spatial field evolves according to a random walk. Letting $p = 4$ denote the number of Fourier coefficients per cycle, the $2p$-vector of spatially-varying coefficients, $\widetilde{\boldsymbol{\beta}}_\ell(\mathbf{s}) = \left[\widetilde{\boldsymbol{\beta}}_{1\ell}(\mathbf{s})', \widetilde{\boldsymbol{\beta}}_{2\ell}(\mathbf{s})'\right]'$, for year ℓ, location $\mathbf{s}$ is modeled by

$$\widetilde{\boldsymbol{\beta}}_\ell(\mathbf{s}) | \widetilde{\boldsymbol{\beta}}_{\ell-1}(\mathbf{s}), \widetilde{\mathbf{w}}_\ell(\mathbf{s}) \sim N\left(\widetilde{\boldsymbol{\beta}}_{\ell-1}(\mathbf{s}) + \widetilde{\mathbf{w}}_\ell(\mathbf{s}), \widetilde{\boldsymbol{\Sigma}}_\beta\right) \tag{2.4}$$

where $\widetilde{\mathbf{w}}_\ell(\mathbf{s}) = \left[\widetilde{\mathbf{w}}_{1\ell}(\mathbf{s})', \widetilde{\mathbf{w}}_{2\ell}(\mathbf{s})'\right]'$ with $\widetilde{\mathbf{w}}_{j\ell}(\mathbf{s}) = [\widetilde{w}_{j1\ell}(\mathbf{s}), ..., \widetilde{w}_{jp\ell}(\mathbf{s})]'$, and

$$\widetilde{w}_{jk\ell}(\mathbf{s}) \overset{\text{ind.}}{\sim} GP\left(0, C\left(\cdot; \boldsymbol{\theta}_{jk}\right)\right),$$

where $k = 1, ..., p$, $\widetilde{\boldsymbol{\Sigma}}_\beta$ is a $2p \times 2p$ unstructured covariance matrix, and $GP(0, C(\cdot; \boldsymbol{\theta}_{jk}))$ denotes a Gaussian process over the spatial domain (Cressie, 1993). Each spatial process $\widetilde{\mathbf{w}}_\ell(\mathbf{s})$ accounts for the residual spatial variation in the Fourier coefficients between year $\ell - 1$ and ℓ. We assume an exponential covariance function, where $C(\mathbf{s}, \mathbf{s}'; \boldsymbol{\theta}_{jk}) = \sigma^2_{jk} \exp\{-||\mathbf{s} - \mathbf{s}'||/\phi_{jk}\}$, $||\mathbf{s} - \mathbf{s}'||$ is the Euclidean distance between locations $\mathbf{s}$ and $\mathbf{s}'$, and $\boldsymbol{\theta}_{jk} = \{\sigma^2_{jk}, \phi_{jk}\}$ consists of the spatial variance and decay parameters, respectively, for process $k = 1, ..., p$. Note that the spatial covariance is process and parameter specific, but assumed constant across years.

Based on the similarities in the parameter estimates discussed in Section 2.2, the harmonic coefficients for both the minimum and maximum cycles are modeled jointly to borrow strength. Dependence between the minimum and maximum temperature cycles is captured through the covariance structure in the coefficients, $\widetilde{\boldsymbol{\Sigma}}_\beta$. If dependence between the minimum and maximum temperature cycles is present, we expect the $p \times p$ off-diagonal sub-matrices of $\widetilde{\boldsymbol{\Sigma}}_\beta$ to be non-zero. Lastly, we let $\widetilde{\beta}_0(\mathbf{s}) | \boldsymbol{\mu}_0 \sim N(\boldsymbol{\mu}_0, \Sigma_0)$, completing the model specification.

2.3.2 Predictive Process

Model inference presents computational challenges due to the number of spatial locations, years, harmonics, and processes. For example, a single draw from the conditional distribution of $\widetilde{\boldsymbol{\beta}}_\ell(\mathbf{s})|\widetilde{\boldsymbol{\beta}}_{\ell-1}(\mathbf{s}), \widetilde{\mathbf{w}}_\ell(\mathbf{s})$ requires matrix operations on a $2pn \times 2pn$ matrix, which is computationally prohibitive for even modest sized data sets. Therefore, we propose using spatio-temporal predictive processes (Finley et al., 2012) to enhance computational efficiency.

Let $\mathbb{S} = \{\mathbf{s}_1, ..., \mathbf{s}_n\}$ be the locations where minimum and maximum temperature data are available. Next, define knot locations $\mathbb{S}^* = \{\mathbf{s}_1^*, ..., \mathbf{s}_m^*\}$ located inside the domain of interest where $m \ll n$. For cycle j, at location $\mathbf{s}$, year ℓ, for process k, we define the predictive process (Finley et al., 2012) as

$$w_{jk\ell}(\mathbf{s}) = E(\widetilde{w}_{jk\ell}(\mathbf{s})|\widetilde{\mathbf{w}}^*_{jk\ell}) = \mathbf{c}(\mathbf{s};\boldsymbol{\theta}_{jk})'\mathbf{C}^*(\boldsymbol{\theta}_{jk})^{-1}\widetilde{\mathbf{w}}^*_{jk\ell}, \tag{2.5}$$

where $\widetilde{\mathbf{w}}^*_{jk\ell} = [\widetilde{w}_{jk\ell}(\mathbf{s}_1^*), ..., \widetilde{w}_{jk\ell}(\mathbf{s}_m^*)]'$, $\mathbf{c}(\mathbf{s};\boldsymbol{\theta}_{jk})'$ is a $1 \times m$ vector whose a^{th} element is $C(\mathbf{s}, \mathbf{s}_a^*; \boldsymbol{\theta}_{jk})$, and $\mathbf{C}^*(\boldsymbol{\theta}_{jk})$ is the $m \times m$ matrix with element (a,b) given by $C(\mathbf{s}_a^*, \mathbf{s}_b^*; \boldsymbol{\theta}_{jk})$. Let $\boldsymbol{\beta}_\ell(\mathbf{s}) = \left[\boldsymbol{\beta}'_{1\ell}(\mathbf{s}), \boldsymbol{\beta}'_{2\ell}(\mathbf{s})\right]'$ and $\mathbf{w}_\ell(\mathbf{s}) = \left[w_{11\ell}(\mathbf{s}), ..., w_{1p\ell}(\mathbf{s}), w_{21\ell}(\mathbf{s}), ..., w_{2p\ell}(\mathbf{s})\right]'$, then the predictive process for the coefficients at year ℓ, $\boldsymbol{\beta}_\ell(\mathbf{s})$, is predicated on all previous predictive processes,

$$\boldsymbol{\beta}_\ell(\mathbf{s}) = \sum_{r=1}^{\ell} \mathbf{w}_r(\mathbf{s}) + \boldsymbol{\eta}_r,$$

where $\boldsymbol{\eta}_r \sim N(0, \boldsymbol{\Sigma}_\beta)$ and $\boldsymbol{\Sigma}_\beta$ is defined as in (2.4). The resulting distribution of the coefficients is

$$\boldsymbol{\beta}_\ell(\mathbf{s})|\boldsymbol{\beta}_{\ell-1}(\mathbf{s}),\mathbf{w}_\ell(\mathbf{s}) \sim N\left(\boldsymbol{\beta}_{\ell-1}(\mathbf{s})+\mathbf{w}_\ell(\mathbf{s}),\boldsymbol{\Sigma}_\beta\right),$$

and the data model (equation 2.3) can be rewritten in the form of the predictive process.

$$\mathbf{z}_{j\ell}(\mathbf{s}) = \mathbf{X}_\ell\boldsymbol{\beta}_{j\ell}(\mathbf{s})+\boldsymbol{\epsilon}_{j\ell}(\mathbf{s}), \qquad j=1,2,$$

where $\boldsymbol{\epsilon}_{j\ell}(\mathbf{s}) \overset{iid}{\sim} N(0,\sigma^2_{\varepsilon_j}(\mathbf{s})\,\mathbf{I}_{T_\ell})$ is defined the same as in Eqn. 2.3. Therefore, conditioned on the predictive process, the coefficient process is spatially independent, and draws from the full conditional of $\boldsymbol{\beta}_\ell(\mathbf{s})$ can be obtained univariately.

2.3.3 Parameter Models

To fully specify the Bayesian hierarchical model, we assign prior distributions to all remaining parameters. Conjugate, non-informative priors were chosen when available to ease computational burden. For $\sigma^2_{\varepsilon_\ell}(\mathbf{s})$, the variance for location $\mathbf{s}$ that is shared across time is modeled $\sigma^2_{\varepsilon_j}(\mathbf{s}) \sim$ Inv-Gamma(a,b). For the $2p\times 2p$ covariance matrix of the β parameters we assign $\boldsymbol{\Sigma}_\beta \sim$ Inv-Wishart$(\mathbf{V},\xi)$. Lastly, the spatial variance for the k^{th} spatial process is modeled $\sigma^2_{jk} \sim$ Inv-Gamma(a_{jk},b_{jk}). All hyperpriors were chosen such that the priors have finite first moments, specifically $\mathbf{V}=\mathbf{I}_8$, $\xi=11$, and $a=b=a_k=b_k=2$ for all k. Preliminary analyses with a uniform prior distribution for the spatial decay parameter, ϕ_{jk} indicated that this parameter had little impact on the inference of the parameters of interest. Therefore, we set $\phi_{jk}=\phi$ to a fixed value in our analysis.

The full hierarchical model can be written

$$\begin{aligned}
&\prod_{s=1}^{n}\prod_{\ell=1}^{L}\prod_{j=1}^{2}\left[\mathbf{z}_{j\ell}(\mathbf{s})|\boldsymbol{\beta}_{j\ell}(\mathbf{s}),\sigma^2_{\varepsilon_j}(\mathbf{s})\right]\prod_{s=1}^{n}\prod_{\ell=1}^{L}\left[\boldsymbol{\beta}_{\ell}(\mathbf{s})|\boldsymbol{\beta}_{\ell-1}(\mathbf{s}),\mathbf{w}_{\ell}(\mathbf{s}),\boldsymbol{\Sigma}_{\beta},\boldsymbol{\theta}_{jk}\right] \\
&\prod_{s=1}^{n}\left[\boldsymbol{\beta}_0(\mathbf{s})|\boldsymbol{\mu}_0,\boldsymbol{\Sigma}_0\right]\prod_{s=1}^{n}\prod_{\ell=1}^{L}\left[\widetilde{\mathbf{w}}^*_{\ell}(\mathbf{s})|\boldsymbol{\theta}_{jk}\right]\left[\boldsymbol{\Sigma}_{\beta}\right]\prod_{s=1}^{n}\prod_{j=1}^{2}\left[\sigma^2_{\varepsilon_j}(\mathbf{s})\right]\prod_{j=1}^{2}\prod_{k=1}^{p}\left[\sigma^2_{jk}\right],
\end{aligned} \tag{2.6}$$

where $\mathbf{w}_{\ell}(\mathbf{s})$ is a deterministic composition of $\widetilde{\mathbf{w}}^*_{\ell}$, as shown in (2.5).

2.3.4 Model Inference

Recall from Section 2.2 that the motivation for this modeling effort is purely inferential. Specifically, we are interested in inference with respect to the spatial processes of harmonic coefficients, $\boldsymbol{\beta}_{\ell}(\mathbf{s})$ for $\ell \in \{1,...,L\}$. We obtain samples from the joint posterior distribution using a Gibbs sampling algorithm, the details of which are given in Appendix A. Each of the parameters described above have conjugate full-conditional distributions. To improve computational efficiency of our sampling algorithm, we also took advantage of parallel computation when possible. Specifically, conditioned on the predictive process, $\mathbf{w}_{\ell}$, the parameters $\boldsymbol{\beta}_{\ell}$ are spatially independent and can be updated in parallel.

Posterior inference will focus on the comparison of the amplitude and phase of the annual and semi-annual harmonics across all years and spatial locations. Visualizing their spatial evolution over time provides insight into how the temperature cycle changes across years. Using samples from the posterior distribution of $\boldsymbol{\beta}_{\ell}(\mathbf{s})$, we can obtain full posterior inference for the phase and amplitude of the annual and semi-annual harmonics using composition sampling. We can also compute important characteristics of these cycles, such as the day at which the cycle reached its peak or trough. These peak and trough days can be compared across time to quantify shifts in temperature cycles which may have important impacts on spatial synchrony. In addition, we can identify similarities and differences in shifts in minimum and maximum temperature cycles.

2.4 Results

We fitted the model to the NCEP Reanalysis temperature data introduced in Section 2.2. The spatial domain of interest spanned the continental United States and portions of Mexico and Canada, with corners at approximately (25.039N, 120.243W), (21.399N, 79.588W), (46.471N, 64.321W), and (52.7418N, 128.744W). We thinned the data spatially to reduce the overall dimension, keeping every other location in both the longitudinal and latitudinal directions. This resulted in daily minimum and maximum temperature values at 3621 spatial locations over 40 years, for a total of over 1.06×10^8 data points. For the predictive process outlined in Section 2.3.2, we chose 144 knot locations evenly spaced across the domain of interest. All temperature time series were centered since the focus of inference is on the harmonics and change in harmonics as opposed to raw temperature.

Using MCMC and the Gibbs sampling algorithm (see Appendix A), we obtained 5000 samples from the joint posterior distribution. The first 1000 samples were discarded as burn-in and the remaining 4000 samples were retained for posterior inference. All computation and posterior inference was performed on a high performance computing infrastructure[2] due to the dimensionality of the data and Bayesian inference output. Convergence of model parameters was assessed visually via trace plots, with no issues detected.

Posterior distributions of the annual and semi-annual phase and amplitude of minimum and maximum temperature cycles were obtained for each year using composition sampling.

[2] Computation was performed on a Linux workstation using an Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40GHz processor utilizing 24 cores.

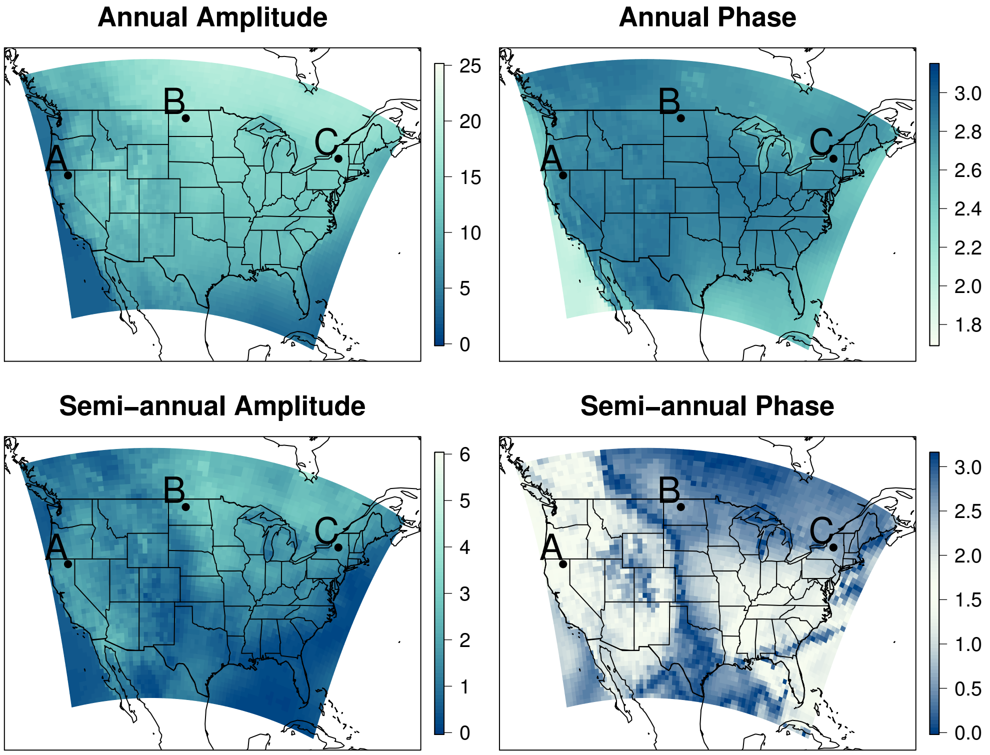


Figure 2.4: Posterior mean estimates of the minimum temperature annual and semi-annual harmonics for 2004. The left two panels show the annual and semi-annual amplitude (deg C), and the right two panels show the annual and semi-annual phase (radians). Location A corresponds to the top plot in Figure 2.6; Location B corresponds to the middle plot in Figure 2.6; Location C corresponds to the bottom plot in Figure 2.6.

Posterior mean estimates of these four cycle quantities for minimum and maximum temperature for the year 2004 are shown in Figures 2.4 and 2.5, respectively. For both figures, estimates of the annual and semi-annual harmonics are shown on the top and bottom panels, respectively, while the amplitude and phase estimates are shown on the left and right, respectively. A prominent spatial feature of the temperature harmonics is the wave-like pattern that appears in the semi-annual amplitude and phase. This wave can been seen most prominently in the bottom-right panel of Figure 2.4 where a band of semi-annual phase values close to 0 (or π) spans from the north-west United States down through the center

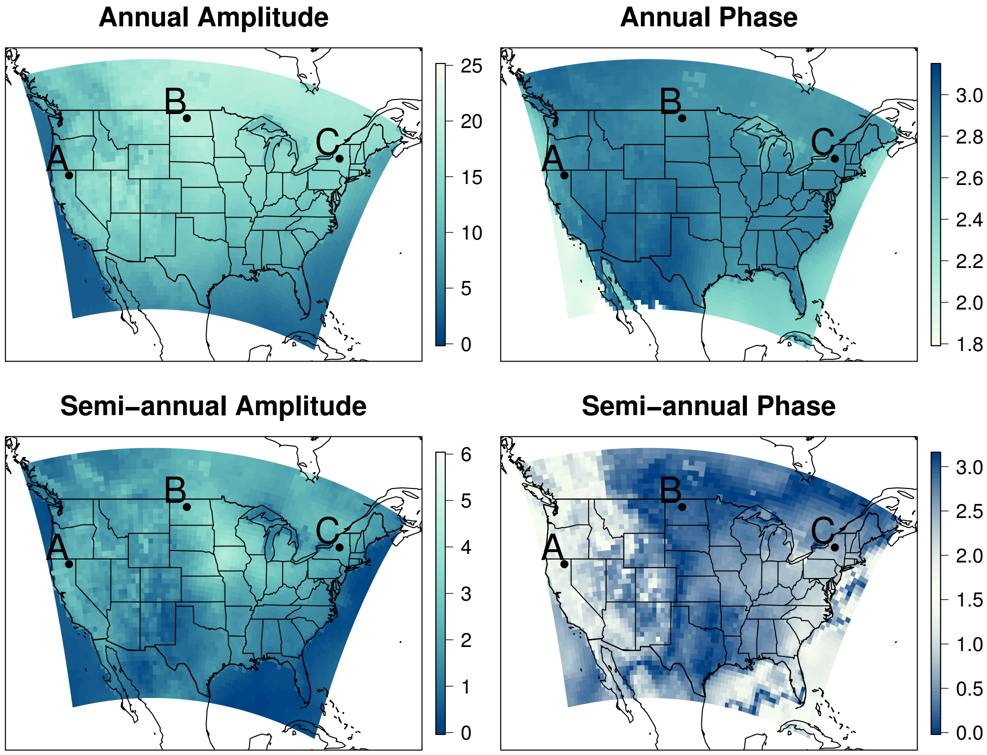


Figure 2.5: Posterior mean estimates of the maximum temperature annual and semi-annual harmonics for 2004. The left two panels show the annual and semi-annual amplitude (deg C), and the right two panels show the annual and semi-annual phase (radians). Location A corresponds to the top plot in Figure 2.6; Location B corresponds to the middle plot in Figure 2.6; Location C corresponds to the bottom plot in Figure 2.6.

of the United States. The semi-annual phase estimates on either side of this band deviate from 0 (or π). The same spatial pattern appears in the semi-annual amplitude, where there is a band of smaller amplitudes following approximately the same path.

These same spatial patterns can are seen in Figure 2.5 for the semi-annual components of the maximum temperature cycle. In the bottom-right panel, the western United States have a contiguous area of lighter colored values close to 1.5 that are bordered by values close to 0 (or π). While less prominent than the minimum semi-annual amplitude, the maximum semi-annual amplitude has a band of smaller amplitudes that spans from the southern

United States up through the center of the United States. These spatial patterns likely arise because of the east-west structure of the semi-annual harmonic due to the land/sea contrast as discussed in the introduction and by Wikle and Chen (1996).

To investigate the temporal variation in minimum and maximum temperature cycles throughout the 40 year period, we produced an animation of the posterior mean estimates of amplitude and phase for the first and second harmonics[3]. The animation illustrates the slow evolution of the annual components and the temporal volatility of the semi-annual components. While the wave-like patterns seen in 2004 (Figures 2.4 and 2.5) are the most common, variations of these spatial patterns appear in both the maximum and minimum semi-annual amplitude and phase in other years. Similar wave-like spatial patterns have been detected for geopotential height (Wallace et al., 1993; Thompson and Wallace, 1998; Thiébaux et al., 1986; Wikle and Chen, 1996) as discussed in the Section 2.1.

To illustrate the component-wise difference in minimum and maximum temperature cycles, Figure 2.6 shows posterior estimates of the annual and semi-annual amplitudes (height of the point on the y-axis) and phase (angular direction of the arrow) simultaneously for minimum and maximum temperature at three different spatial locations. The locations, denoted "A", "B", and "C" in Figures 2.4 and 2.5, are in the west, north central, and northeast, respectively. The relationship between the minimum and maximum annual amplitudes differ across space, which could be attributed to climatological variations. Specifically, for locations "A" and "B", the range between the minimum and maximum annual amplitudes is greater than for location "C", and the interannual variability for the annual amplitudes is much greater for location "B" than locations "A" and "C". The minimum and maximum semi-annual phase are the same for most years (i.e., phase locked), with relatively

[3] https://joshuanorth.shinyapps.io/harmonics_application/

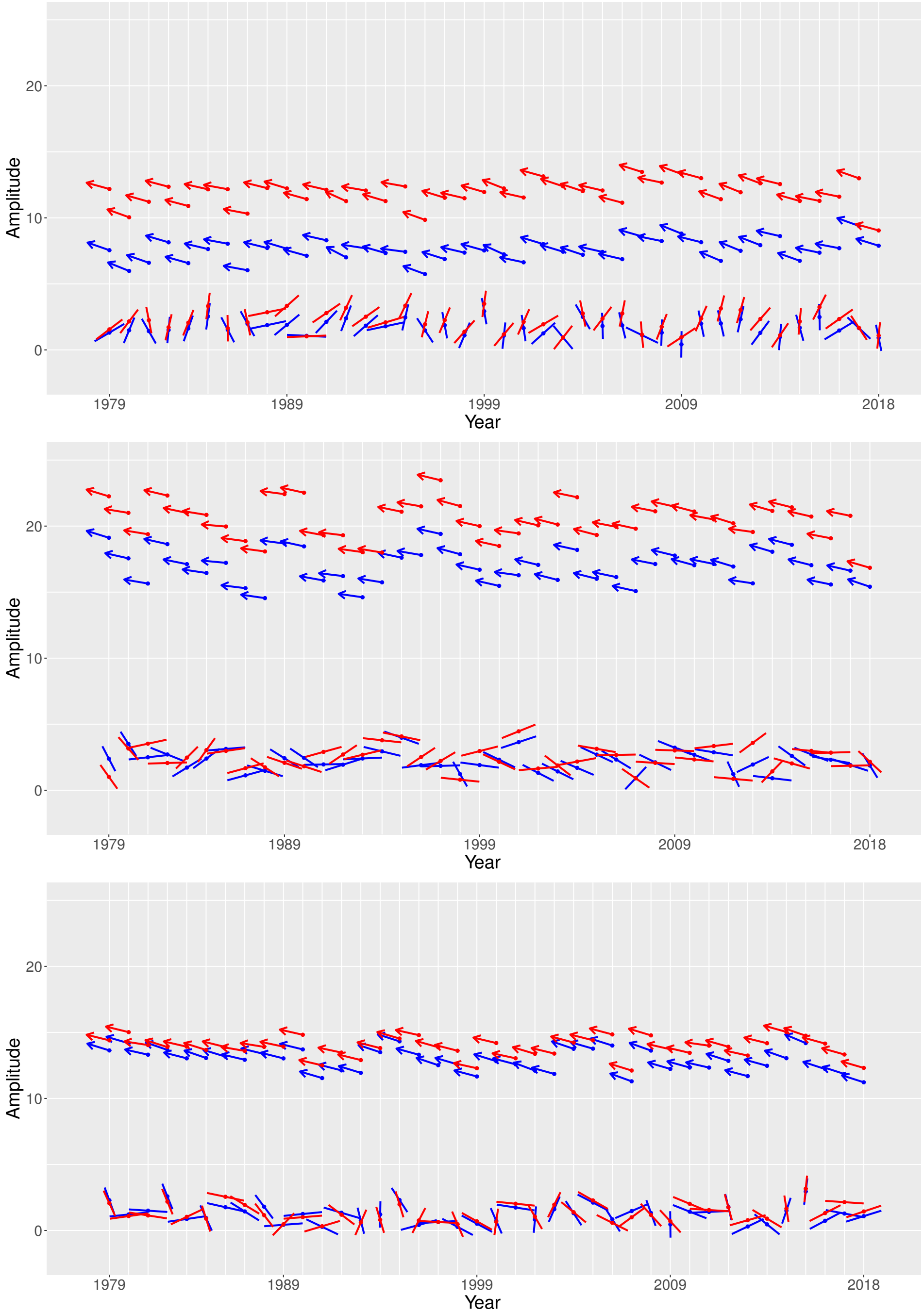


Figure 2.6: The phase in radians (angular direction of the arrow/line), $\varphi \in [0, 2\pi]$, and amplitude (height of the point) in degrees Celsius, $A \in [0, \infty)$, for the minimum (blue) and maximum (red) temperature cycles at three spatial locations across the United States. Arrows correspond to the annual estimates and lines correspond to the semi-annual estimates. See Figure 2.4 corresponding to geographic locations, with the top plot corresponding to location A, middle plot to location B, and bottom plot to location C.

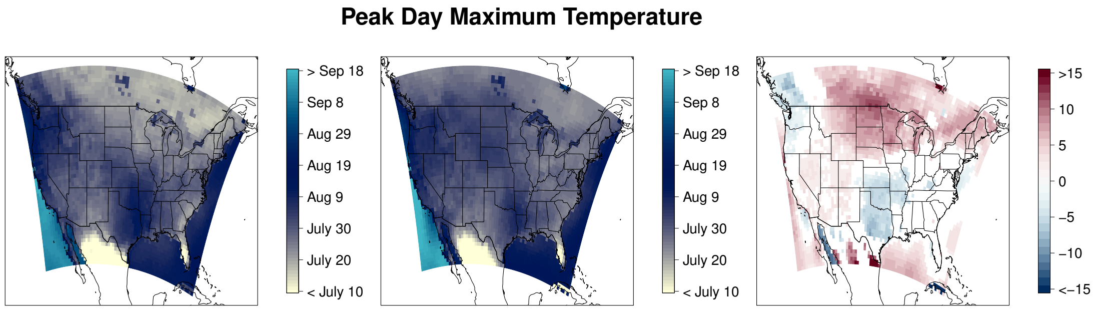


Figure 2.7: Day at which the maximum temperature cycle reaches its peak. The left image is the average over the years 1979-1988, the center image is the average over the years 2009-2018, and the right image is the difference between the two images (2009-2018 minus 1979-1988), showing only locations where there is a significant difference.

little year-to-year change in the semi-annual amplitude. Compared to the annual phase, the semi-annual phase is more volatile with each location experiencing differing degrees of variability. The semi-annual phase appears the most variable for location "C".

The extent to which the temperature cycles have shifted (i.e., how the temperature cycle determined by the estimates of the first two harmonics has changed over time) over the 40 year period can be seen by comparing the day of the year at which the temperature cycles are at their peak and trough. We computed the average peak and trough day for the years 1979-1988 and 2009-2018 as well as the differences between these two time periods (computed as 2009-2018 minus 1979-1988). These posterior distributions can be used to identify spatial regions experiencing significant shifts in the minimum and maximum temperature cycles. We consider a shift to be significant if the 95% credible interval of the difference does not include zero.

Figures 2.7 and 2.8 show the average day of the year in which the maximum and minimum temperature cycles, respectively, obtain their peak, as determined by the first two harmonics. The peak day for maximum temperature can be thought of as the hottest day

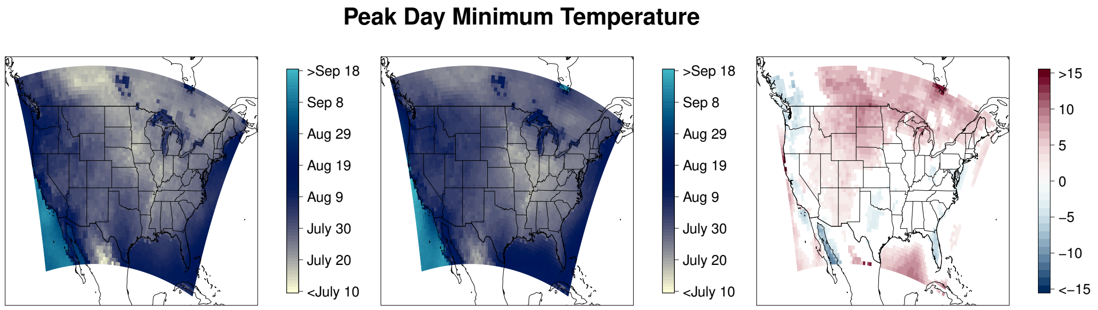
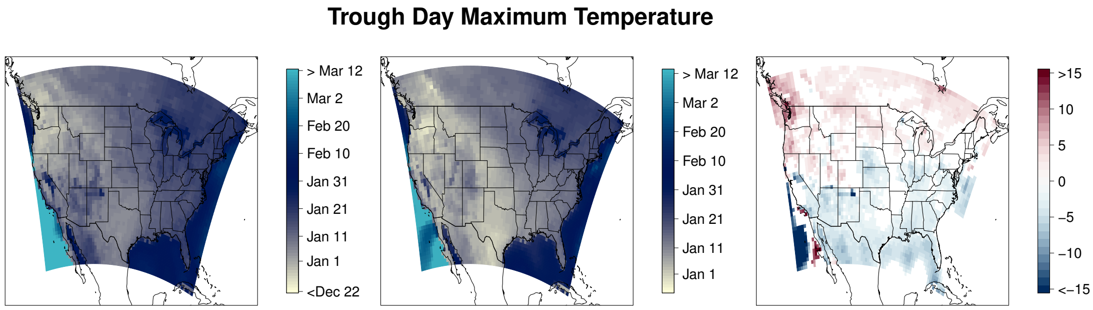


Figure 2.8: Day at which the minimum temperature wave reaches its peak. The left image is the average over the years 1979-1988, the center image is the average over the years 2009-2018, and the right image is the difference between the two images (2009-2018 minus 1979-1988), showing only locations where there is a significant difference.

of the year, and the peak day for minimum temperature is the day at which the warmest low temperature occurs. The differences in peak days between the two decades for both maximum and minimum temperature clearly identify regions experiencing seasonal shifts in temperature. The areas in red indicate locations for which the peak day is occurring later in the year for the 2009-2018 decade, whereas blue regions correspond to locations in which the peak day is occurring earlier in the year for the more recent decade. The spatial patterns in the shifts in maximum and minimum temperature appear similar across the domain. The northern regions (Montana, the Dakotas, Minnesota, and Canada) appear to be experiencing the greatest shift towards later seasonal peaks, with much of the western United States experiencing more moderate shifts. For both minimum and maximum temperature, the only two areas experiencing a shift to earlier seasonal peaks are located in the Midwest United States and along the Northwest coast.

Figures 2.9 and 2.10 show the day for which each maximum and minimum temperature cycle reaches its trough as determined by the first two harmonics. For minimum temperature, the trough corresponds to the coldest day of the year, and for the maximum



Trough Day Maximum Temperature

Figure 2.9: Day at which the maximum temperature cycle reaches its lowest. The left image is the average over the years 1979-1988, the center image is the average over the years 2009-2018, and the right image is the difference between the two images (2009-2018 minus 1979-1988), showing only locations where there is a significant difference.

temperature, the trough captures the day of the coldest high temperature. Again, we see very similar patterns between minimum and maximum temperature cycles and shifts. In contrast to the spatial distribution of the shift for the peak day, the spatial distribution of the shift for the trough day has a strong north/south pattern. The northern half of the United States and Canada are experiencing a shift toward later seasonal troughs, whereas the southern half of the United States and Mexico are experiencing a shift toward earlier seasonal troughs.

To highlight the contribution of the semi-annual component on the temperature shift, we obtained posterior distributions of the peak and trough days as well as the decadal shifts for both the minimum and maximum temperature cycles using only the annual components. We then computed the posterior difference in the decadal shifts between those obtained when both the annual and semi-annual component were included and those when only the annual component was included. The posterior mean of these differences are shown in Figure 2.11. We considered the contribution of the semi-annual component to be significant if the 95% point-wise credible interval of the posterior distribution of differences did not

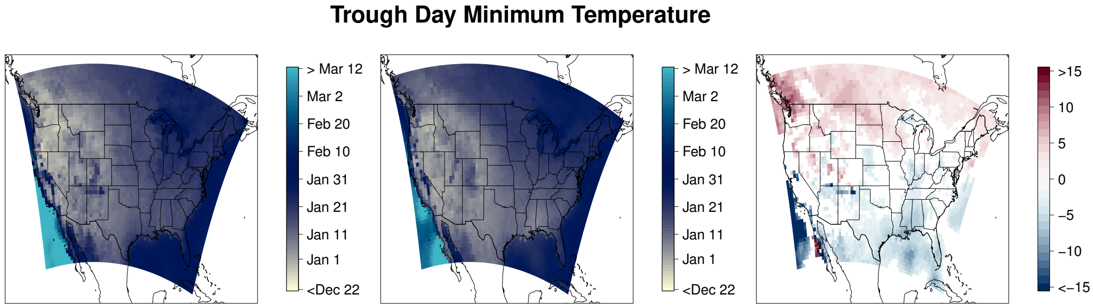


Figure 2.10: Day at which the minimum temperature cycle reaches its lowest. The left image is the average over the years 1979-1988, the center image is the average over the years 2009-2018, and the right image is the difference between the two images (2009-2018 minus 1979-1988), showing only locations where there is a significant difference.

include zero. Based on these posterior credible intervals, white indicates locations where the semi-annual harmonic did not significantly contribute to the shift in the temperature cycle. All other locations indicate that the semi-annual component contributed significantly in capturing shifts in temperature cycles between the two decades. In the left panels of Figure 2.11, red (blue) indicates locations for which the peak day has shifted to later (earlier) in the year when the semi-annual harmonic is considered. Similarly, in the right panels of Figure 2.11, red (blue) indicate locations where the trough day occurs later (earlier) in the year when the semi-annual harmonic is considered. In each of these figures, the shading corresponds to the magnitude of these differences. The spatial distribution of significant semi-annual harmonic contributions are similar for both the minimum and maximum temperature cycles. The magnitude of the shift is higher for maximum temperature than for minimum, which could be attributed to the maximum temperature having more seasonal variation than the minimum. The semi-annual component significantly contributes to the later peak day (positive shift) in both the minimum and maximum temperature cycle in the North (North Dakota, South Dakota, and Minnesota), Southwest (New Mexico and Ari-

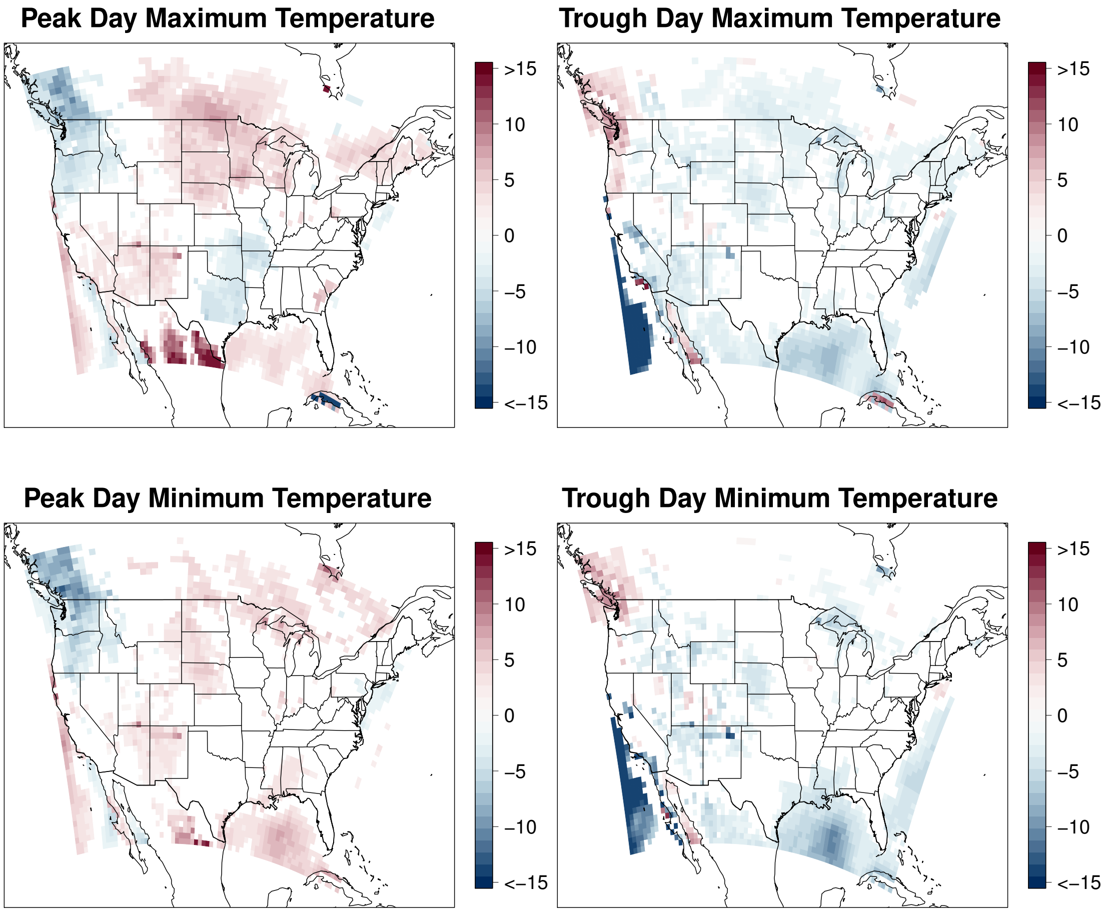

Peak Day Maximum Temperature
Trough Day Maximum Temperature
Peak Day Minimum Temperature
Trough Day Minimum Temperature
>15
10
5
0
−5
−10
<−15

Figure 2.11: Difference of the decadal shifts for the estimates considering the annual and semi-annual component with estimates considering only the annual component. Shading corresponds to the magnitude of the differences, with white areas corresponding to locations where the semi-annual component is not significant. Red indicates the peak/trough day has occurred later in the year when the semi-annual harmonic is considered, and blue indicates the peak/trough has occurred earlier in the year.

zona), and the Gulf of Mexico. The semi-annual component significantly contributes to the earlier trough day (negative shift) in both minimum and maximum temperature cycles in the Gulf of Mexico and western United States.

To visualize the importance of modeling the dependence between the parameters, Fig-

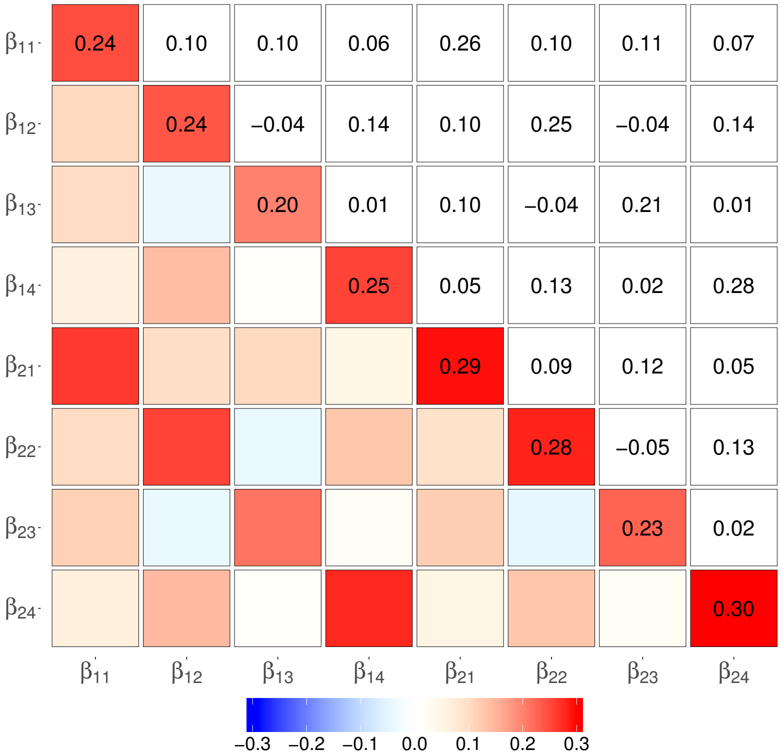


Figure 2.12: Posterior mean estimate of covariance, Σ_β. All numeric values on the figure have been rounded to two digits for readability. The posterior estimates of the symmetric matrix are shown numerically in the upper triangle while the lower triangle shows the magnitude on the color-scale for easy comparison. The two 4×4 sub-matrices on the diagonal correspond the the covariance between the minimum and maximum Fourier coefficients, respectively. The off-diagonal 4×4 block sub-matrix corresponds to the covariance between the minimum and maximum Fourier coefficients.

ure 2.12 shows the posterior mean estimate for the covariance matrix, $\widehat{\Sigma}_\beta$. The two 4×4 sub-matrices on the diagonal of $\widehat{\Sigma}_\beta$ correspond to the estimated covariance of the minimum and maximum Fourier coefficients, respectively. The 4×4 off-diagonal block sub-matrix of $\widehat{\Sigma}_\beta$ corresponds the estimated covariance between the minimum and maximum Fourier coefficients. Within each 4×4 sub-matrix, the top 2×2 and bottom 2×2 sub-matrices on the diagonal correspond to the estimated covariance of the annual and semi-annual Fourier coefficients, respectively, and the two 2×2 off-diagonal block sub-matrices correspond to the estimated covariance between the annual and semi-annual Fourier coefficients. The posterior estimates of the symmetric matrix are shown numerically in the upper triangle while the lower triangle shows the magnitude on the color-scale for easy comparison. All elements of the matrix were significant based on their 95% credible intervals not covering 0. From Figure 2.12, aside from the expected main diagonal component, the off-diagonal block 4×4 sub-matrix has a strong diagonal component. The diagonal elements of the sub-matrix show the positive dependence between the minimum and maximum temperature cycles. The two 4×4 sub-matrices on the diagonal, which capture the dependence between the harmonic coefficients within the minimum and maximum cycles, share a similar structure. All 4 diagonal and off-diagonal block sub-matrices show the same positive and negative correlation, with negative dependence only between the semi-annual cosine term to the annual sine term. Importantly, this negative dependence between the semi-annual cosine term and annual sine term is consistent both within and between minimum and maximum cycles.

2.5 Discussion and Future Work

We proposed modeling minimum and maximum temperature cycles jointly through the components of the annual and semi-annual harmonics using a DSTM to detect temporal changes in the seasonal temperature cycle that may vary across space. Implementing our model in a Bayesian paradigm, we obtain estimates of the annual and semi-annual phase and amplitude through composition sampling. Spatial maps showing the difference in peak/trough days of the minimum and maximum temperature cycles for the years 2009-

2018 relative to 1979-1988 identified regions experiencing seasonal shifts, as well as regions for which the semi-annual component contributed significantly to these shifts. These maps showed that the peak day for both minimum and maximum temperature cycles has shifted to later in the year in northern regions, and the trough day has shifted toward later in the year in the northern half and earlier in the year in the southern half of the United States.

The results of our analysis can be compared to those presented in previous research. For example, a similar east-west structure in the semi-annual cycle was reported by Wikle and Chen (1996) and attributed to the land-sea contrast. Additionally, using only the annual harmonic, Stine et al. (2009) found an asymmetrical spatial pattern in the shift of the temperature cycle. However, since we considered both the annual and semiannual harmonic, our results differed from theirs in terms of the regions identified as experiencing asymmetric shifts in temperature cycles. Lastly, our model detected spatially-varying shifts in the peak/trough of the temperature cycle ranging between 15 days earlier to 15 days later in the year, whereas Dwyer et al. (2012) reported the annual phase in the temperature cycle is shifting to only later in the year.

While the results of our model have scientific merit of their own, they can also be used to detect changes in spatial synchrony between temperature cycles and other important environmental processes. For example, incorporating estimates of shifting temperature cycles in models for bird migration could identify regions for which the spatial synchrony between migration patterns of birds and temperature cycles have been disrupted. Similarly, we can investigate the effects temperature shifts on the occupancy or abundance of native bark beetles, which could lead to improved predictions of beetle spread or risk of invasion as well as aid in conservation efforts. While these are just two brief examples, a better understanding of the direction and magnitude in the shifts in temperature cycles over the last 40 years will motivate future scientific hypotheses with regard to the effects of these changes on important environmental processes.

Chapter 3

A Review of Data-Driven Discovery of Dynamic Systems

3.1 Introduction

Recently there has been a push from within the computer science, physics, and mathematical fields to learn the governing equations in complex dynamic systems parameterized through dynamic equations (DE). There are a variety of reasons researchers may want to know the underlying laws driving a system – to reinforce their assumptions, uncover extra information about the system, or to produce a more realistic mathematical equation for the system. Historically, scientists have relied on their ability represent physical systems using mathematical equations in the form of DEs. Dating back to at least the inference of equations describing the motion of orbital bodies around the sun based on the positions of celestial bodies (Legendre, 1806; Gauss, 1809), DEs have been used to model the evolution of complex processes (e.g., the use of susceptible, infected, recovered models for epidemics), and have become ubiquitous across virtually every area of science and engineering. Here, we review some of the methods used to discover the governing equations driving complex, potentially nonlinear, processes, often referred to as *data-driven discovery*.

Consider the general DE describing the evolution of a continuous process $\{\mathbf{u}(\mathbf{s},t) : \mathbf{s} \in D_s, t \in D_t\}$,

$$\mathbf{u}_{t^{(J)}}(\mathbf{s},t) = M\left(\mathbf{u}(\mathbf{s},t), \mathbf{u}_x(\mathbf{s},t), \mathbf{u}_y(\mathbf{s},t), ..., \mathbf{u}_{t^{(1)}}(\mathbf{s},t), ..., \mathbf{u}_{t^{(J-1)}}(\mathbf{s},t), \boldsymbol{\omega}(\mathbf{s},t)\right) \tag{3.1}$$

where the vector $\mathbf{u}(\mathbf{s},t) \in \mathbb{R}^N$ denotes the state of the system at location $\mathbf{s}$ and time t, $\mathbf{u}_{t^{(j)}}(\mathbf{s},t)$ is the jth order temporal derivative of $\mathbf{u}(\mathbf{s},t)$, J denotes the highest order of the temporal derivative, $M(\cdot)$ represents the (potentially nonlinear) evolution function, and $\boldsymbol{\omega}(\mathbf{s},t)$ represents any covariates that might be included in the system. We will denote partial derivatives by a subscript; that is $\frac{\partial \mathbf{u}}{\partial x} = \mathbf{u}_x$ and $\frac{\partial \mathbf{u}}{\partial t} = \mathbf{u}_t$, for example. Here, N is the number of components in the system (e.g., $\mathbf{u}(\mathbf{s},t) = [u(\mathbf{s},t,1), u(\mathbf{s},t,2), ..., u(\mathbf{s},t,N)]'$, sometimes called the system state), $\mathbf{s} \in \{\mathbf{s}_1, ..., \mathbf{s}_S\} = D_s$ is a discrete location in the domain with $|D_s| = S$, and $t \in \{1, ..., T\} = D_t$ is the realization of the system at discrete times where $|D_t| = T$. Equation (3.1) is composed of partial derivatives of the system with $D_s \in \mathbb{R}^2$ and $\mathbf{s} = (x,y)$ and is often referred to as a partial differential equation (PDE). Removing the spatial component from (3.1) results in a temporal ordinary differential equation (ODE),

$$\mathbf{u}_{t^{(J)}}(t) = M\left(\mathbf{u}(t), \mathbf{u}_{t^{(1)}}(t), ..., \mathbf{u}_{t^{(J-1)}}(t), \boldsymbol{\omega}(t)\right), \tag{3.2}$$

where M is composed solely of derivatives of the components in time (i.e., no partial derivatives) and $D_s \in \mathbb{R}^1$ and $\mathbf{s} = x$. This review will focus on methods to discover the evolution function M for both PDEs (3.1) and ODEs (3.2).

The goal of data-driven discovery is to learn the governing equation(s) in (3.1) and (3.2) – specifically the (non)linear function M – having only observed noisy realizations of $\mathbf{u}$ (i.e., true derivatives are unknown). Broadly, there are two approaches used for data-driven discovery. The first approach uses sparse regression where a library of potential solutions are proposed and the correct solution set is obtained by regularization based techniques, resulting in a sparse solution. The second uses symbolic regression where the solution is learned, or generated, through the estimation procedure. Within both of these approaches, regression based methods or deep models are used to facilitate the discovery process. While delineation between these methods is not always clear, we attempt to separate the ideas into three categories – classical sparse methods, classical symbolic methods, and deep modeling methods using either symbolic or sparse regression techniques. Methods to quantify uncertainty in the discovered equations have been proposed, but they do not account for uncertainty in the observed data, missing a vital piece of the statistical puzzle. We draw parallels between traditional statistical models and data-driven discovery, discussing how statistical models can be formulated for data-driven discovery and highlighting how the proposed statistical data-driven discovery methods can be improved upon.

3.2 Sparse Regression

Sparse regression approaches for dynamic discovery of ODEs and PDEs are fundamentally the same. We formulate the general approach using (3.1), noting that the approach for (3.2) is equivalent but with only one spatial location (i.e., $S = 1$). First, consider rewritting (3.1) as a linear (in parameters) system

$$\mathbf{u}_{t^{(J)}}(\mathbf{s},t) = \mathbf{f}(\mathbf{u}(\mathbf{s},t),...)\,\mathbf{M}, \tag{3.3}$$

where $\mathbf{M}$ is a $D \times N$ *sparse* matrix of coefficients and $\mathbf{f}(\cdot)$ is a vector-valued nonlinear transformation function of length D termed the *feature library*. The input of the arguments for $\mathbf{f}(\cdot)$ are general and contain anything that *potentially* relates to the system (e.g., advection term, polynomial terms, interactions). Sparse identification seeks to identify relevant terms of $\mathbf{M}$, thereby identifying the terms of $\mathbf{f}$ that drive the system and discovering the governing dynamics.

Denote the matrix of all data (all components at all time points) for the jth derivative of the system as

$$\mathbf{U}_{t^{(j)}} = \begin{bmatrix} u_{t^{(j)}}(\mathbf{s}_1,1,1) & u_{t^{(j)}}(\mathbf{s}_1,1,2) & \cdots & u_{t^{(j)}}(\mathbf{s}_1,1,N) \\ u_{t^{(j)}}(\mathbf{s}_1,2,1) & u_{t^{(j)}}(\mathbf{s}_1,2,2) & \cdots & u_{t^{(j)}}(\mathbf{s}_1,2,N) \\ \vdots & \vdots & \vdots & \\ u_{t^{(j)}}(\mathbf{s}_S,T,1) & u_{t^{(j)}}(\mathbf{s}_S,T,2) & \cdots & u_{t^{(j)}}(\mathbf{s}_S,T,N) \end{bmatrix}.$$

The response matrix is $\mathbf{U}_{t^{(J)}}$ of size $(ST) \times N$ and we generically denote the feature library as

$$\mathbf{F} = \left[\mathbf{1}, \mathbf{U}_{t^{(0)}}, ..., \mathbf{U}_{t^{(J)}}, \mathbf{U}_x, \mathbf{U}_y, \mathbf{U}_{xx}, ..., \boldsymbol{\Omega}\right].$$

where $\boldsymbol{\Omega}$ are the associated covariates at each space-time step and $\mathbf{F}$ is a $(ST) \times D$ matrix. The library may also contain interactions of the components, partial derivatives, and covariates. This reduces to a linear system

$$\mathbf{U}_{t^{(J)}} = \mathbf{F}\mathbf{M}, \tag{3.4}$$

whereby identifying the terms of $\mathbf{M}$ that are non-zero, the DE is identified.

However, the derivatives of the system are rarely observed (i.e., only $\mathbf{U}_{t^{(0)}}(t)$ is measured). To obtain derivatives in space and time, numerical techniques are used to approximate the derivatives. There are multiple methods to approximate derivatives numerically, and the choice of approximation has the potential to impact the discovered equation. Originally, a finite difference approach was suggested, but this approach is sensitive to noise. When measurement noise is present, data are either smoothed *a priori* and then derivatives are computed, or derivatives are computed using total variation regularization (Chartrand, 2011) or polynomial interpolation (Knowles and Renka, 2012).

Due to both the numerical approximation of the derivative and the potential for noise in the observed data, (3.4) does not hold exactly. Instead,

$$\mathbf{U}_{t^{(J)}} = \mathbf{F}\mathbf{M} + \boldsymbol{\epsilon}, \tag{3.5}$$

where $\boldsymbol{\epsilon} \overset{i.i.d.}{\sim} N(\mathbf{0}, \sigma^2 \mathbf{I}_N)$ and σ^2 is the variance in the data. It is crucial to note that σ^2 is not the measurement uncertainty from the original data, but rather some form of uncertainty associated with the model approximation and the numerical differentiation. To induce sparsity, solutions to (3.5) of the form

$$\mathbf{M} = \underset{\widehat{\mathbf{M}}}{argmin} \|\mathbf{U}_{t^{(J)}} - \mathbf{F}\widehat{\mathbf{M}}\|_2^2 + Pen_\theta(\widehat{\mathbf{M}}), \tag{3.6}$$

are sought, where $Pen_\theta(\widehat{\mathbf{M}})$ generically denotes some penalty term based on parameters θ (i.e., $Pen_\theta(\widehat{\mathbf{M}}) = \lambda||\widehat{\mathbf{M}}||_1$ where $\theta = \lambda$ for the LASSO penalty).

3.2.1 Deterministic Approaches

The majority of deterministic approaches are composed of three steps – denoising and differentiation, constructing a feature library, and sparse regression. Assuming data have been properly differentiated and a library has been proposed, the deterministic approach seeks solutions of the form (3.6). The original sparse regression approach to data-driven discovery, *Sparse Identification of Nonlinear Dynamics* (SINDy; Brunton et al., 2016), uses sequential threshold least-squares (STLS; Algorithm 2) to discover the governing terms for ODEs. While the original paper does not discuss the algorithm in terms of a penalty term, STLS has been shown to be equivalent to the ℓ_0 penalty, $Pen_\theta(\widehat{\mathbf{M}}) = ||\widehat{\mathbf{M}}||_0$ (Zhang and Schaeffer, 2019), which removes values of $\mathbf{M}$ less than some pre-specified threshold κ. That is, at each iteration of the minimization procedure, values of $\mathbf{M} < \kappa$ are set to zero and the remaining values of $\mathbf{M}$ are re-estimated. In the original implementation, the algorithm was only iterated over 10 times, but a stopping criteria (e.g., change in loss or identified parameters) could be used. In this manner, a sparse solution set is obtained.

SINDy is illustrated on a variety of simulated ODE problems with varying amounts of noise. The examples used generally contain a lot of observations (on the order of hundreds of thousands), and it is unclear the impact of noise if a smaller number of observations were considered. In contrast to the symbolic approaches discussed in Section 3.3, SINDy can be fit rather quickly. However, a drawback of the approach is the sensitivity to the thresholding parameter and the dependence on the method approximating the derivative.

To extend SINDy to PDEs, Rudy et al. (2017) propose Sequential Threshold Ridge

Regression (STRidge, Algorithm 3), a variant to STLS. Due to the correlation present in $\mathbf{F}$ for data pertaining to PDEs, STLS was insufficient at finding a sparse solution set. Instead, STRidge uses the same iterative technique as STLS, where values of $\mathbf{M} < \kappa$ are set to zero at each iteration, but with the addition of the penalty term $Pen_\theta(\widehat{\mathbf{M}}) = \lambda ||\widehat{\mathbf{M}}||_2^2$. Cross-validation is then used to find the optimal values for κ and λ. The effectiveness of STRidge is shown on multiple simulated data sets with varying noise. Again, in comparison to the symbolic counterparts, the algorithm is quick, but still dependent on the method used to approximate the derivative.

STRidge can be adapted to allow for parametric PDEs by grouping terms either spatially or temporally (Rudy et al., 2019a). To incorporate parametric PDEs in 3.4, the coefficients now vary in space or time (i.e., $\mathbf{M}(\mathbf{s})$ or $\mathbf{M}(t)$) and $\mathbf{F}$ is constructed as a block diagonal matrix of the appropriate form (e.g., either in space or time). Similar to the group LASSO (Meier et al., 2008), coefficients are assigned group indices $g \in \mathscr{G}$ by grouping the same location in space over the entire time domain (e.g., $g \equiv \mathbf{s}$ and $\mathscr{G} \equiv D_S$) or the same time point over the whole spatial domain (e.g., $g \equiv t$ and $\mathscr{G} \equiv D_T$). Within the STRidge algorithm all coefficients with the same group index are set to zero if $||\mathbf{M}(g)||_2 < \kappa$. In this manner, the same dynamics are identified across space and time and only the coefficient estimate is allowed to vary in space or time.

Champion et al. (2020) propose a robust unifying algorithm (Algorithm 4) for the SINDy framework based on sparse relaxed regularized regression (SR3; Zhang and Lin, 2018). SR3 introduces an auxiliary variable $\mathbf{W}$ within the penalization term, resulting in $Pen_\theta(\widehat{\mathbf{M}}) = \lambda R(\mathbf{W}) + \frac{1}{2\nu}||\widehat{\mathbf{M}} - \mathbf{W}||$, where $R(\cdot)$ is another penalization term (e.g. ℓ_1). The addition of the auxiliary variable provides a geometrically more favorable surface to optimize (Zhang and Lin, 2018). SR3 is shown to be able to handle outliers (a potential issue when numerically differentiating noisy data), accommodate parametric formulations, and allow for physical constraints in the library.

While not discussed in detail here, there are other areas of DE where discovery has been applied and approaches at discovering dynamics within the sparse identification category. Applying SINDy to stochastic differential equations (Boninsegna et al., 2018) and systems where the dynamics evolve on a different coordinate system (Champion et al., 2019) further build on the SINDy applicability. Instead of using finite differences or total variation regularization, Schaeffer (2017) use spectral methods to compute spatial derivatives and the Douglas-Rachford algorithm (Combettes and Pesquet, 2011) to find a sparse solution. Further consideration of highly corrupt signals (Tran and Ward, 2017), convergence properties of the SINDy algorithm (Zhang and Schaeffer, 2019), and the choice of denoising and differentiation methods (Lagergren et al., 2020) have also received treatment within the literature. For ease of use, SINDy and some related variants have been developed into a python package *PySINDy* (de Silva et al., 2020).

3.2.2 Addressing Uncertainty

Bayesian and bootstrapping approaches have been proposed to quantify uncertainty in the parameters for the sparse regression formulation of data-driven discovery. These approaches seek to quantify the variability in the discovered equation and parameters for (3.5).

Bayesian Approach

A penalized likelihood estimator of the form (3.6) can analogously be cast as the posterior mode in a Bayesian framework under the prior $p(\mathbf{M}|\boldsymbol{\theta})$ where $Pen(\widehat{\mathbf{M}})_{\boldsymbol{\theta}} = \log p(\mathbf{M}|\boldsymbol{\theta})$.

That is, (3.5) can be formulated in the Bayesian framework where priors are put on $\mathbf{M}$ and σ^2. Instead of an optimization procedure, the Bayesian approach aims to sample from the joint posterior distribution

$$p(\mathbf{M}, \boldsymbol{\sigma}^2 | \mathbf{F}, \mathbf{U}_{t(j)}) \propto p(\mathbf{U}_{t(j)} | \mathbf{F}, \mathbf{M}, \boldsymbol{\sigma}^2) p(\mathbf{M}|\theta) p(\boldsymbol{\sigma}^2), \tag{3.7}$$

where $p(\mathbf{U}_{t(j)}|\mathbf{F}, \mathbf{M}, \boldsymbol{\sigma}^2)$ is the data likelihood (3.5), and $p(\mathbf{M}|\theta)$ and $p(\boldsymbol{\sigma}^2)$ are prior distributions for $\mathbf{M}$ and $\boldsymbol{\sigma}^2$, respectively. To enforce a sparse solution set in a Bayesian framework, a regularization prior is placed on the parameter of interest, in this case $\mathbf{M}$. Further discussion comparing the sparse regression approach to a Bayesian formulation of the problem is explored by Niven et al. (2020).

Using the Bayesian framework in an algorithmic setting, Zhang and Lin (2018) propose using the priors $p(m_d|\alpha_d) = \prod_{d=1}^{D} N(0, \alpha_d^{-1})$, $p(\sigma^2) = IG(a_s, b_s)$, and $p(\alpha_d^{-1}) = IG(a_a, b_a)$. They estimate the parameters using a threshold sparse Bayesian regression algorithm, which maximizes the marginal likelihood instead of sampling from the full conditionals. Their algorithm uses a hard thresholding parameter, similar to the deterministic sparse regression approaches, where at each iteration values of the posterior $\mathbf{M} < \kappa$ are set to zero. From their procedure, they are able to assign what they term “error bars” to their parameter estimates based on the ratio of the estimate for the posterior variance to the estimate for the posterior mean squared. Zhang and Lin (2018) consider many of the same simulated ODEs and PDEs used to illustrate the deterministic approaches and provide error bars to the parameter estimates for these systems with varying amounts of measurement noise.

Hirsh et al. (2021) explore the use of two common Bayesian selection priors on system discovery and uncertainty quantification – the continuous spike and slab (i.e., stochastic search variable selection (SSVS); Mitchell and Beauchamp, 1988; George et al., 1993; George and McCulloch, 1997), and the regularized horseshoe (Carvalho et al., 2010; Piironen and Vehtari, 2017) – calling the approach *uncertainty quantification SINDy* (UQ-SINDy). Their choice of priors are distinct in that SSVS is a mixture of two continuous mean zero Gaussian distributions and the horseshoe is part of the global-local shrinkage prior family. For the SSVS prior, variables that are not to be included in the model are sampled from a mean zero Gaussian distribution with a small variance, rendering their effect on inference negligible, and variables that are to be included are sampled from a mean zero Gaussian distribution with a larger variance. The posterior inclusion probability for a variable is the number of times it was sampled from the Gaussian with a large variance over the total number of samples. In contrast, the horseshoe prior has a hyper-prior performing global shrinkage on all variables in conjunction with individual hyper-priors on all the variables performing individual shrinkage. To determine the probability a variable is included under the regularized horseshoe, the ratio of the estimate of $\mathbf{M}$ with no prior and with the horseshoe prior is computed, providing a *pseudo-probability* (i.e., not necessarily bounded by 0 and 1) of inclusion probabilities. Using both of these priors, Hirsh et al. (2021) provide inclusion probabilities for multiple simulated ODE systems with varying amounts of noise and to the classic hare-lynx population data set (Elton and Nicholson, 1942).

However, UQ-SINDy is limited in that the uncertainty being quantified is the uncertainty in the numerical approximation of the system (i.e., the numerical differentiation and de-noising). That is, because the approximated derivative is, in fact, a single realization of the true derivative (which is unknown), the uncertainty estimates recovered by this approach are biased toward this single approximation of the derivative. A more complete treatment of the problem would be to consider the derivative as a random process and account for uncertainty in the random process.

Yang et al. (2020) propose the use of Bayesian differentiable programming as a method by which to discover the dynamics and account for measurement uncertainty when estimating parameters. Generally speaking, Bayesian differentiable programming uses a numerical solver (e.g., Runge-Kutta) to predict the state at a new time, and the loss between the predicted data and observed data is used to estimate parameters. More precisely, let $M_\theta(\mathbf{u}(t))$ be the output of a numerical solver at time t. Bayesian differentiable programming aims to minimize $\sum \|\mathbf{u}(t+\Delta t) - M_\theta(\mathbf{u}(t))\|^2$, where Δt does not need to be uniformly spaced. The parameters are estimated using Hamiltonian Monte Carlo and differentiable programming is used to compute gradients within the Hamiltonian Monte Carlo algorithm. By directly relating the observed data to the dynamics, measurement uncertainty is accounted for in the estimation procedure, providing a more thorough statistical treatment to the data-driven discovery problem. The approach is illustrated on multiple simulated ODE systems with varying amounts of measurement noise.

Bootstrap Approach

Fasel et al. (2021) propose two methods of bootstrapping (3.4) – either sampling rows of the data (i.e., space-time sampling) or sampling library terms (i.e., columns of $\mathbf{F}$). The first approach samples rows of the data with replacement and uses STRigde to estimate the parameters in the model q times. In the second approach, the columns of $\mathbf{F}$ are sampled *without* replacement to create q data sets, and again STRidge is used to estimate parameters. For both methods, the q models are then averaged and coefficients with an inclusion probability below some pre-specified value are set to zero. Uncertainty is quantified by the inclusion probability and the distribution of values obtained from the q different estimates.

However, as with Hirsh et al. (2021), the uncertainty associated with the observed data is not considered and the numerical approximation to the derivative is assumed the true realization of the derivative, limiting true uncertainty quantification. The method is illustrated on multiple simulated ODE and PDE systems with varying noise and applied to the classic hare-lynx population data set.

3.3 Symbolic Regression

Symbolic regression is a type of regression that searches over mathematical expressions (e.g., $+, -, \times$) to find the optimal model for a given data set (Wang et al., 2019). This approach differs from classical regression where the model structure is fixed and a set of parameters are estimated. One of the main challenges underlying symbolic regression is that there are an infinite number of combinations of expressions that can be used to fit any particular data set. An algorithmic procedure called *genetic programming* is used to efficiently search over the possible model structures (Willis, 1997; Koza et al., 1993; Koza, 1994) and regression techniques are used to determine coefficient values given the model structure. Genetic programming follows Darwin's theory of evolution, selecting the "fittest" solution that is the product of generations of evolution (i.e., iterating through an algorithm). Here, we give a brief overview of genetic programming and its roll in symbolic regression and subsequently data-driven discovery of dynamics. For a more detailed overview of genetic programming, see Minnebo and Stijven (Chapter 4, 2011) and Garg and Tai (2012).

Genetic programming relies on a predefined *function set* of mathematical expressions. For symbolic regression, the function set typically consists of basic mathematical expressions such as addition, multiplication, and trigonometric terms (see Nicolau and Agapitos,

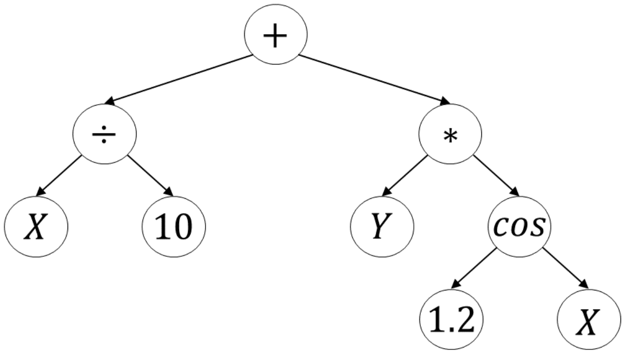


Figure 3.1: Symbolic representation of $f(X,Y) = \frac{X}{10} + Y * 1.2\cos(X)$.

2018, for details on function set choice). Possible model solutions are constructed using a combination of functions from the function set and encoded in a tree structure (Figure 3.1). Within the tree, decision nodes are the mathematical expressions and terminal nodes are the input data passed into the mathematical expression. To make the searchable space smaller, the maximum node size of the tree can be specified (i.e., restrict the depth of the tree). A *population* of potential solutions is composed of *individual* potential solutions. The ability of an individual to properly represent data is determined based on the *fitness function*, which is analogous to an objective or loss function in statistics. Individuals can then *reproduce* to create a copy of themselves, *crossover* with another individual, or *mutate* themselves. Crossover is where two individuals swap sub-trees (i.e., randomly select a decision node from each tree and exchange) to produce two new individuals, which is equivalent to parents producing offspring with shared genetics. Mutation is where an individuals decision node is randomly changed (e.g., plus to multiplication or plus to a variable), which is akin to a genetic mutation.

The general algorithm for genetic programming proposes an initial population, assesses the fitness of each individual, and then generates the next population based on the fittest individuals of the current population (Algorithm 5). Taking the fitness function to be based on regression, where the goal is to minimize mean squared error or the one minus r-squared (Schmidt and Lipson, 2009), results in symbolic regression. This basic genetic programming/symbolic regression method has generated multiple extensions (Icke and Bongard, 2013; Chen et al., 2017; Amir Haeri et al., 2017; Jin et al., 2019) and incurred extensive discussion (Korns, 2014; Nicolau and Agapitos, 2018; Ahvanooey et al., 2019).

Within the context of data-driven discovery, symbolic regression attempts to find the evolution function $M(\cdot)$ in (3.1) or (3.2). The difficulty is relating a proposed choice for $M(\cdot)$ that is generated withing the genetic algorithm to derivatives of the observed data. Specifically, because derivatives of the system are unknown, either the fitness function needs to account for the derivative or the derivatives must be obtained in order to use a traditional fitness function.

Bongard and Lipson (2007) were the first to apply symbolic regression to data-driven discovery of dynamic systems, focusing on the discovery of ODEs. In order to use symbolic regression to discover dynamic models with potentially nonlinear interactions of multiple variables, the authors introduced partitioning, automated probing, and snipping within a symbolic regression algorithm. Partitioning regards each variable in a system separately, even though they may be coupled, drastically reducing the search space of possible equations. With partitioning, a candidate equation for a single variable is integrated with the others assumed fixed. Automated probing is where initial conditions used for temporal integration of the dynamic equation of the system are found. Last, snipping is the process of simplifying and restructuring models by replacing sub-expressions (sub-trees) in the generated population with a constant. Using these three components, each variable in the system is integrated forward in time to produce a “test” based on the initial condition and compared to the observed data. The fitness of each potential solution is computed based on the average absolute difference between the observed data and the test.

Bongard and Lipson (2007) incorporate these components into the symbolic regression methodology by generating an initial population, partitioning the system, probing for initial conditions, snipping the solutions, assessing the fitness, and repeat. The approach is illustrated on simulated data and on two real-world examples - the classic hare-lynx system and data they collect from a pendulum. However, their method is sensitive to noise and has the same demanding computational requirements as other symbolic regression algorithms.

Schmidt and Lipson (2009) adopt a different approach to data-driven discovery with symbolic regression. They search over a function space constrained by a loss function dependent on partial derivatives computed from the symbolic functions and from the data. Specifically, given two variables observed over time, $x(t)$ and $y(t)$ (i.e., $\mathbf{u}(t) = [x(t), y(t)]'$), the numerical estimate of the partial derivatives between the pair is approximated as $\frac{\Delta x}{\Delta y} \approx \frac{dx}{dt} / \frac{dy}{dt}$, where $\frac{dx}{dt}$ and $\frac{dy}{dt}$ are estimated using local polynomial fits (Thompson and Wallace, 1998). From a potential solution function (i.e., generated in the genetic algorithm), the partial derivatives can be computed using symbolic differentiation to get $\frac{\delta x}{\delta y}$ (i.e., from the symbolic function). To determine how well the potential function expresses the data, the mean log error between the approximated and symbolic partial derivatives,

$$-\frac{1}{N}\sum_{i=1}^{N} \log\left(1 + abs\left(\frac{\Delta x}{\Delta y} - \frac{\delta x}{\delta y}\right)\right),$$

is used as the fitness function. While not discussed here, the approach can be extended to systems with more than two variables by looking at pairs of the variables in the system (see supplementary material of Schmidt and Lipson, 2009, for details). In this manner, they assign a fitness to each proposed individual based on how well the derivative of the system relates to the derivative of the data, resulting in data-driven discovery using symbolic regression. However, noise can be impactful because the derivatives from the observed data are approximated numerically. To accommodate measurement uncertainty, Schmidt and Lipson (2009) use Loess smoothing (Cleveland and Devlin, 1988) prior to fitting to remove the high frequency noise. Their approach is illustrated using simulated data and data collected by motion tracked cameras, showing an ability to recover the equations on complex, real-world problems. However, similar to Bongard and Lipson (2007), the method is also computationally cumbersome with some examples reportedly taking days to converge.

Motivated by symbolic and sparse regression, Maslyaev et al. (2019) embed sparse regression within the coefficient estimation step in a symbolic regression algorithm to discover the governing equations of PDEs. In their approach, derivatives of the data are computed *a priori* using finite difference (in the same manner as sparse regression discussed in Section 3.2) and used as the response in symbolic regression. Within the symbolic regression algorithm, after a population has been proposed, sparse regression using an ℓ_1 penalty is employed, the fitness of each individual in the population is assessed, and mutation, crossover, and replication are performed in the usual manner. Because derivatives are computed before the estimation procedure, they are able to be incorporated into the function set. This allows for the discovered equations to contain spatial derivatives. The approach is tested on multiple simulated PDEs with varying amounts of measurement noise. However, the robustness to measurement noise is dependent on the numerical method used to approximate the derivative, and it is unclear how this impacts model results. Additionally, while specifics are not given, the approach is computationally cumbersome, owing in part to the symbolic regression.

3.4 Deep Models

Deep modeling has been considered for data-driven dynamic discovery in two different ways – approximating and learning dynamics. Approximating dynamics using deep models provides a computationally cheap method to generate data from complex systems while still preserving physical aspects of the system (i.e., emulation). While this review is concerned with the discovery of the governing equations and refers to “data-driven discovery” as the discovery of the *functional form* of the governing system, deep models approximating the dynamics are an important part of the literature and we devote a section to them. Deep models coinciding with our definition of data-driven discovery have also been developed. There are multiple approaches by which dynamics can be approximated and subsequently learned, and we provide a discussion of these following the discussion on approximating dynamics.

3.4.1 Approximating Dynamics with Deep Models

One method of approximating dynamics considers a so-called *physics-informed neural network* (PINN; Raissi et al., 2017a,b; Raissi, 2018; Raissi et al., 2019, 2020). PINNs are applicable to both continuous and discrete time models, and we discuss only the continuous version here. Define

$$\mathbf{g}(\mathbf{s},t) = \mathbf{u}_{t^{(J)}}(\mathbf{s},t) + M\left(\mathbf{u}(\mathbf{s},t), \mathbf{u}_x(\mathbf{s},t), \ldots\right), \tag{3.8}$$

and assume the form of M is known. Approximating $\mathbf{u}(\mathbf{s},t)$ with a neural network results in the PINN $\mathbf{g}(\mathbf{s},t)$, where the derivatives associated with the PINN are computed using automatic differentiation. The neural network is trained using the loss function $MSE = MSE_u + MSE_g$ where MSE_u is the mean squared error of neural network approximating $\mathbf{u}(\mathbf{s},t)$ and $MSE_g = \frac{1}{N_g}\sum_{i=1}^{N_g} \|\mathbf{g}(\mathbf{s}_i,t_i)\|^2$ is the mean squared error associated with structure imposed by $\mathbf{g}(\cdot)$. In this manner, the neural network is trained such that it obeys the physical constraints imposed by $\mathbf{g}(\cdot)$.

Neural networks have also been used to approximate the evolution operator M using a residual network (ResNet). Framing the problem similar to the Euler approximation $\mathbf{U}(t+\Delta t) \approx \mathbf{U}(t) + \Delta t M(\mathbf{U}(t))$, the goal is to find a suitable approximation for $M()$, thereby approximating the dynamics. In contrast to PINN, physics are not incorporated into the NN and the structure of the NN is dependent completely on the data. Applying the problem to ODEs, Qin et al. (2019) show how a recurrent ResNet with uniform time steps (i.e., uniform Δt) and a recursive ResNet with adaptive time steps can be used to approximate dynamics. This approach is further extended to PDEs (Wu and Xiu, 2020), where the evolution operator is first approximated by basis functions and coefficients, and a ResNet is fit to the basis coefficients.

While not described in detail here, there are other approaches to approximating DE using deep models. Physics-informed candidate functions can be used with numerical integration in an objective function to restrict the temporal evolution of a NN (Sun et al., 2019). NN have also been used to approximate parametric PDEs (Khoo et al., 2021), represent molecular dynamics (Mardt et al., 2018), and approximate ODEs with time-varying measurement data (Wu and Xiu, 2019).

3.4.2 Discovering Dynamics with Deep Models

Deep modeling using neural networks (NNs) have become increasingly popular in recent years due to NN's ability as a universal approximator (Hornik et al., 1989). Additionally, computing derivatives of NNs is possible through automatic differentiation (e.g., using PyTorch; Paszke et al., 2017). Assuming a surface can be approximated using a NN, derivatives of the surface in space or time or both are obtainable. This approach, where derivatives are computed using NN, is used in many of the deep model approaches to data-driven discovery.

Deep Models with Sparse Regression

A common issue with data-driven discovery in the "classical" sparse regression approach is the sensitivity to noise when approximating derivatives numerically. To address this issue, Both et al. (2021) propose using a NN to approximate the system, and then perform sparse regression within the NN. Let $\widehat{\mathbf{U}}$ be the output of a NN and construct $\mathbf{F}$ in (3.4) using $\widehat{\mathbf{U}}$ and derivatives computed from $\widehat{\mathbf{U}}$ via automatic differentiation. The NN is trained using the loss function

$$\mathscr{L} = \frac{1}{ST}\sum|\mathbf{U} - \widehat{\mathbf{U}}|^2 + \frac{1}{ST}\sum|\mathbf{FM} - \widehat{\mathbf{U}}_{t^{(J)}}|^2 + \lambda\sum|\mathbf{M}|.$$

After training the NN and estimating parameters, most terms of $\mathbf{M}$ are still nonzero (but very close to zero), and a thresholding is performed on $\mathbf{M}$ to obtain the final sparse representation. Through this formulation of the problem, whereby derivatives are obtained from a NN, Both et al. (2021) show their ability to recover highly corrupt signals from traditional PDE systems and apply their approach to a real-world electrophoresis experiment.

Deep Models with Symbolic Regression

Using symbolic regression with a neural network has become an increasingly popular method for data-driven discovery. In a series of papers, Xu et al. (2019, 2020, 2021) construct a deep-learning genetic algorithm for the discovery of parametric PDEs (DLGA-PDE) with sparse and noisy data. DLGA-PDE first trains a NN that is used to compute derivatives and generate meta-data (global and local data), thereby producing a complete de-noised reconstruction of the surface (i.e., noisy sparse data are handled through the NN). Using the *local* metadata produced by the NN, a genetic algorithm learns the general form of the PDE and identifies which parameters vary spatially or temporally. At this step, the coefficients may be incorrect or missrepresent the system because the global structure of the data is not accounted for. To correct the coefficient estimates, a second NN is trained using the discovered structure of the PDE and the *global* metadata. Last, a genetic algorithm is used to discover the general form of the varying coefficients.

One method of implementing symbolic regression within a deep model is to allow the activation functions to be composed of the function set instead of classic activation functions (e.g., sigmoid or ReLU; Martius and Lampert, 2016; Sahoo et al., 2018; Kim et al., 2021). Motivated by this idea, Long et al. (2019) propose a symbolic regression NN, *SymNet*. Similar to a typical NN, the ℓth layer of *SymNet* is

$$\mathbf{f}^{\ell} = \mathbf{W}^{\ell}[\mathbf{f}^{0}, \mathbf{f}^{\ell-1}] + \mathbf{b}^{\ell},$$

where $\mathbf{f}^0$ is the function set that contains partial derivatives (e.g., $\mathbf{f}^0 = [u, u_x, u_y, ...]$). In this manner, each subsequent layer adds a dimension to the activation function based on the previous layer, allowing the construction of complex functions. Similar to Long et al.

(2017), spatial derivatives are computed using finite-difference via convolution operators. To model the time dependence of PDEs, they employ the forward Euler approximation, termed a δt-block, as

$$\mathbf{U}(t+\delta t) \approx \mathbf{U}(t) + \delta t \cdot SymNet_m^k(u, u_x, u_y, ...),$$

where δt is the temporal discritization, and $SymNet_m^k(u, u_x, u_y, ...)$ has k hidden layers (i.e., $\ell = 0, ..., k$) and m variables (i.e., number of arguments $u, u_x, u_y, ...$). In order to facilitate long-term predictions, they train multiple δt-blocks as a group so the system has long-term accuracy.

Distinct from the previous two approaches, Atkinson et al. (2019) incorporate differential operators into the function set of a genetic algorithm. They train a NN on the observed data and supply the NN to a genetic algorithm where the function set contains typical operators (e.g., addition, multiplication) and differential operators. The differential operators are computed from the NN using PyTorch (Paszke et al., 2017), enabling the inclusion of derivatives in the search space of the genetic algorithm.

3.5 Physical Statistical Models

To account for observational uncertainty and missing data when modeling complex nonlinear systems, dynamic equations (DE) parameterized by ordinary and partial differential equations have been incorporated into Bayesian hierarchical models (BHM). While there are various methods by which to model DE in a probabilistic framework, here we focus on physical statistical models (PSM; Berliner, 1996; Royle et al., 1999; Wikle et al., 2001) due to the similarities with data-driven discovery that will become apparent shortly. Broadly, PSM are a class of BHMs where scientific knowledge about some process is known and incorporated into the model structure.

PSMs are generally composed of three modeling stages – data, process, and parameter models – where dynamics are modeled in the process model and the observed data are modeled conditioned on the latent dynamics. That is, the observed data are considered to be a *realization* of the "true" latent dynamic process. This formulation results in the data being described conditionally given the process model, simplifying the dependence structure in the data model and enabling complex structure to be captured in the process stage. The evolution of the latent dynamic process is then parameterized by a DE, incorporating physical dynamics into the modeling framework.

Consider the $R(t) \times 1$ observed data vectors $\mathbf{V}(t) \equiv [v(\mathbf{r}_1,t),...,v(\mathbf{r}_{R(t)},t)]'$ where $\{v(\mathbf{r},t) : \mathbf{r} \in D_s, t \in D_t\}$ where $\mathbf{r} \in \{\mathbf{r}_1,...,\mathbf{r}_{R(t)}\} \subset D_s$ is a discrete location in the spatial domain with D_s, $t \in \{1,...,T\} \subset D_t$ is the realization of the system at discrete times in some temporal window D_t. Assume we are interested in the latent "true" dynamic process $\{u(\mathbf{s},t) : \mathbf{s} \in D_s, t \in D_t\}$ where $\mathbf{U}(t) \equiv [u(\mathbf{s}_1,t),...,u(\mathbf{s}_S,t)]'$ is a length S vector. It is common that the observation locations do not coincide with the process (e.g., due to missing data or different resolution). In the case of missing observations, the observed data are mapped to the latent process using an incidence matrix $\mathbf{H}(t)$, which is a matrix of zeros except for a single one in each row corresponding the the observation associated with a process location (see Chapter 7 of Cressie and Wikle, 2011, for examples of $\mathbf{H}(t)$). The general data model for time t is

$$\mathbf{V}(t) = \mathbf{H}(t)\mathbf{U}(t) + \boldsymbol{\eta}(t), \tag{3.9}$$

where $\mathbf{H}(t) \in \mathbb{R}^{L(t) \times N}$ and uncertainty in the observations of the process are captured by

$\boldsymbol{\eta}(t) \overset{\text{indep.}}{\sim} N_{L(t)}(\mathbf{0}, \boldsymbol{\Sigma}_V(t))$ where $\boldsymbol{\Sigma}_V(t)$ is the variance/covariance matrix.

By specifying how $\mathbf{U}(t)$ evolves over time, the dynamic process is characterized. For example, the process model, which specifies this evolution under a first-order Markov assumption, is given as

$$\mathbf{U}(t) = M(\mathbf{U}(t-1), \boldsymbol{\theta}) + \boldsymbol{\epsilon}(t), \tag{3.10}$$

where $M(\cdot)$ is some (non)linear function relating a previous space-time location (or multiple locations) to the next, $\boldsymbol{\theta}$ are parameters associated with M, and $\boldsymbol{\epsilon}(t) \overset{\text{i.i.d.}}{\sim} N(\mathbf{0}, \boldsymbol{\Sigma}_U)$ is a mean zero Gaussian process with variance/covariance matrix $\boldsymbol{\Sigma}_U$. While not discussed here, the error term $\boldsymbol{\epsilon}(t)$ can be considered multiplicative (see Chapter 7 of Cressie and Wikle, 2011, for more detail).

Physical dynamics are encoded through the parameterization of M. Here, we consider physical dynamic parameterizations (i.e., ODEs and PDEs), but a general autoregressive structure for M (i.e., not parameterized with differential equations) can also be considered. Consider the general PDE

$$\mathbf{U}_t(t) = M(\mathbf{U}(t), \boldsymbol{\theta}),$$

analogous to the motivating PDE (3.1), which can be approximated using finite differences

$$\mathbf{U}(t) = \mathbf{U}(t-1) + \Delta t M(\mathbf{U}(t-1), \boldsymbol{\theta}),$$

where Δt is the difference in time between time t and $t-1$ and $\boldsymbol{\theta}$ are parameters associated with the PDE. Because the finite difference approximation can be written as a linear system, we can write

$$\mathbf{U}(t) = \mathbf{M}\mathbf{U}(t-1), \tag{3.11}$$

where $\mathbf{M}$ is a sparse matrix derived from the finite difference scheme. Replacing the general process model with (3.11), the process model parameterized by a linear finite difference equation is

$$\mathbf{U}(t) = \mathbf{M}\mathbf{U}(t-1) + \boldsymbol{\epsilon}(t), \tag{3.12}$$

where $\boldsymbol{\epsilon}(t)$ may now account for approximation error due to the finite difference approximation.

As a clarifying example, assume a spatio-temporal process $U(x,t)$ in one-dimensional space $0 \leq x \leq L$ and time t. Assume the process is approximated by the diffusion equation $U_t(x,t) = bU_{xx}(x,t)$ where b is a diffusion constant and the boundary conditions $Y(0,t) = U_0$ and $U(L,t) = U_L$ and initial condition $\{U(0,t) : 0 \leq x \leq L\}$ are known. Using numerical analysis, the time derivative can be approximated using the forward difference

$$U_t(x,t) \approx \frac{U(x,t+\Delta t) - U(x,t)}{\Delta t},$$

and the spatial derivative can be approximated by the central difference

$$U_{xx}(x,t) \approx \frac{U(x+\Delta x,t) - 2U(x,t) + U(x-\Delta x,t)}{\Delta x^2}.$$

Using the finite difference approximation, we can reformulate the diffusion equation as

$$U(x,t+\Delta t) \approx U(x,t) + \frac{b\Delta t}{\Delta x^2}\left(U(x+\Delta x,t) - 2U(x,t) + U(x-\Delta x,t)\right).$$

Assuming three internal spatial locations, x_1, x_2, x_3 and boundary locations x_0, x_L, let $\mathbf{U}(t) = [U(x_1,t), U(x_2,t), U(x_3,t)]'$ and $\mathbf{U}^b(t) = [U(x_0,t), U(x_L,t)]'$. Then,

$$\mathbf{U}(t+\Delta t) \approx \begin{bmatrix} 1-\frac{2b\Delta t}{\Delta x^2} & \frac{b\Delta t}{\Delta x^2} & 0 \\ \frac{b\Delta t}{\Delta x^2} & 1-\frac{2b\Delta t}{\Delta x^2} & \frac{b\Delta t}{\Delta x^2} \\ 0 & \frac{b\Delta t}{\Delta x^2} & 1-\frac{2b\Delta t}{\Delta x^2} \end{bmatrix} \mathbf{U}(t) + \begin{bmatrix} \frac{b\Delta t}{\Delta x^2} & 0 \\ 0 & 0 \\ 0 & \frac{b\Delta t}{\Delta x^2} \end{bmatrix} \mathbf{U}^b(t),$$

which can be written more compactly as $\mathbf{U}(t+\Delta t) \approx \mathbf{M}\mathbf{U}(t) + \mathbf{M}^b\mathbf{U}^b(t)$. Thus, the PDE dynamics have been "encoded" into the structure of the transition operator, $\mathbf{M}$. In most PSM implementations, the (banded) structure of $\mathbf{M}$ is retained, but the specific elements are estimated from the data, rather than given by the finite difference representation. This adds flexibility and explicitly assumes that the PDE is not an exact representation of the data. Note that other PDE representations, such as finite element, or spectral, can be used to motivate such models.

This simple example can be made more complex by considering a parametric diffusion equation (i.e., resulting in $\mathbf{M}(\boldsymbol{\theta})$ instead of $\mathbf{M}$) or by placing priors on the boundary conditions and or the initial condition (see Cressie and Wikle, 2011, for details). Additionally, there are certain numerical conditions that need to be satisfied in order to guarantee numerical stability from the approximation, which can vary based on the system and approximation scheme considered (e.g., see CFL condition in Higham et al., 2016). For a more complete overview of PSMs and possible parameterizations, see Berliner (2003); Cressie and Wikle (2011); Kuhnert (2017) and references within.

PSMs have been used to study a variety of real-world systems. PSMs parameterized using shallow-water equations (Wikle, 2003) and the Rayleigh friction equation (Milliff et al., 2011) have been used to study ocean surface winds. Using a parametric diffusion equation (Wikle, 2003) or parametric reaction-diffusion equation (Hooten and Wikle, 2008), PSMs have modeled the spread of invasive avian species. PSMs can be grouped into a larger category of models called general quadratic nonlinear model (GQN; Wikle and Hooten, 2010; Wikle and Holan, 2011; Gladish and Wikle, 2014), which accommodate multiple classes of scientific-based parameterization such as PDEs and integro-difference equations.

3.5.1 General Quadratic Nonlinear Models

General quadratic nonlinear models provide a nice generalization to the PSM framework and, as discussed in the subsequent section, provide an interesting link between data-driven discovery methods and PSMs. The general GQN model is

$$u(\mathbf{s}_i,t) = \sum_{j=1}^{S} a_{ij} u(\mathbf{s}_j, t-1) + \sum_{k=1}^{S}\sum_{l=1}^{S} b_{i,kl} u(\mathbf{s}_k, t-1) g(u(\mathbf{s}_l, t-1); \boldsymbol{\theta}) + \varepsilon(\mathbf{s}_i, t), \tag{3.13}$$

for $i = 1, ..., S$, where a_{ij} are linear evolution parameters, $b_{i,kl}$ are nonlinear evolution parameters, $g()$ is some transformation function of $u(t-1)$ dependent on parameters $\boldsymbol{\theta}$, and $\varepsilon(\mathbf{s}_i,t)$ is an error process. The motivation here is that many real-world mechanistic processes have been described by PDEs that have quadratic (nonlinear) interactions, often where the interaction of system components consists of the multiplication of one component times a transformation of another (see Wikle and Hooten, 2010, for details).

Equation (3.13) can be condensed in matrix form as

$$\mathbf{U}(t) = \mathbf{A}\mathbf{U}(t-1) + (\mathbf{I}_S \otimes g(\mathbf{U}(t-1);\boldsymbol{\theta})')\mathbf{B}\mathbf{U}(t-1) + \boldsymbol{\epsilon}(t), \tag{3.14}$$

where $\mathbf{A}$ and $\mathbf{B}$ are matrices constructed from a_{ij} and $b_{i,kl}$, respectively, and $\mathbf{I}_S$ is a size S identity matrix (see Wikle and Hooten, 2010, for specific details). From (3.14), we see that letting $\mathbf{M}(\mathbf{U}(t-1),\boldsymbol{\theta}) = \mathbf{A}\mathbf{U}(t-1) + (\mathbf{I}_S \otimes g(\mathbf{V}(t-1);\boldsymbol{\theta})')\mathbf{B}\mathbf{U}(t-1)$ recovers the PSM model. The GQN framework is very flexible, due in part to the over-parameterization of the model from all possible quadratic interactions. To constrain the parameter space, either physics-informed priors or strong shrinkage priors are used. For examples on what these constraints may be and the underlying physical motivation, see Wikle and Hooten (2010).

3.5.2 Relation to Data-Driven Discovery

While unexplored in the literature, there is a strong connection between PSMs (particularly, the more general GQNs) and data-driven discovery. Formulating a BHM where the latent process evolves according to the generic PDE (3.1), the two-stage data-process model for location $\mathbf{s}$ and time t is

$$\begin{aligned} \mathbf{v}(\mathbf{s},t) &= \mathbf{H}(\mathbf{s},t)\mathbf{u}(\mathbf{s},t) + \boldsymbol{\epsilon}(\mathbf{s},t) \\ \mathbf{u}_{t^{(J)}}(\mathbf{s},t) &= M(\mathbf{u}(\mathbf{s},t), \mathbf{u}_x(\mathbf{s},t), \ldots) + \boldsymbol{\epsilon}(\mathbf{s},t), \end{aligned} \tag{3.15}$$

where $\boldsymbol{\epsilon}(\mathbf{s},t) \sim N(\mathbf{0}, \boldsymbol{\Sigma}_V(\mathbf{s},t))$ is the measurement error process with $\boldsymbol{\Sigma}_V(\mathbf{s},t)$ a variance/-covariance matrix, $\boldsymbol{\epsilon}(\mathbf{s},t) \sim N(\mathbf{0}, \boldsymbol{\Sigma}_U(\mathbf{s},t))$ the process model error process with $\boldsymbol{\Sigma}_U(\mathbf{s},t)$ a variance/covariance matrix. However, as discussed in Section 3.5, PSMs rely on M to be parameterized according to known dynamics. Instead, borrowing the notion of a feature library from the sparse regression approach to data-driven discovery, linearizing the process model results in a matrix of coefficients $\mathbf{M}$ and a feature library $\mathbf{f}(\cdot)$. The goal is to find the correct values of $\mathbf{M}$ (as in sparse regression), given a library of values to search over $\mathbf{f}(\cdot)$. The connection in the case of GQN is that we rarely need the whole set of quadratic interactions, so the "discovery" connection is selecting which quadratic components are needed to describe the data.

As an example, consider two approaches that can be used to incorporate dynamic discovery into PSMs - employing a finite difference scheme or using (3.5) for the process model – each of which have their own pros and cons. The finite difference approach results in the same model as in Section 3.5 and 3.5.1,

$$\begin{aligned} \mathbf{v}(\mathbf{s},t) &= \mathbf{H}(\mathbf{s},t)\mathbf{u}(\mathbf{s},t) + \boldsymbol{\epsilon}(\mathbf{s},t) \\ \mathbf{u}(\mathbf{s},t) &= \mathbf{Mf}(\mathbf{u}(\mathbf{s},t-1), \mathbf{u}_x(\mathbf{s},t-1), \ldots) + \boldsymbol{\epsilon}(\mathbf{s},t), \end{aligned} \tag{3.16}$$

where $\mathbf{M}$ represents the coefficients associated with the finite difference and the discovered equation. Directly incorporating (3.5) in the process model results in

$$\begin{aligned} \mathbf{v}(\mathbf{s},t) &= \mathbf{H}(\mathbf{s},t)\mathbf{u}(\mathbf{s},t) + \boldsymbol{\epsilon}(\mathbf{s},t) \\ \mathbf{u}_{t^{(J)}}(\mathbf{s},t) &= \mathbf{Mf}(\mathbf{u}(\mathbf{s},t), \mathbf{u}_x(\mathbf{s},t), \ldots) + \boldsymbol{\epsilon}(\mathbf{s},t), \end{aligned} \tag{3.17}$$

where now the temporal derivative is directly related to a library of potential functions and $\mathbf{M}$ represents the coefficients associated only with the discovered equation.

The benefit of formulating the problem using (3.16) is a Kalman filter or ensemble Kalman filter can be used to estimate parameters (see Stroud et al., 2018; Katzfuss et al., 2020, for examples of the Kalman filter with dynamic systems in statistics). Additionally, as mentioned previously, the GQN framework naturally provides a construction of an over-parameterized library of potential dynamical interactions into the library. However, interpreting parameters can be difficult and incorporating spatial derivatives into the library is not as straightforward as with traditional PSMs. In contrast, (3.17) has a very clear interpretation of parameters but requires a method to obtain derivatives. Additionally, model estimation will rely on Metropolis-Hastings Monte-Carlo as the Markov assumption required for Kalman filter and EnKF methods is violated. For both approaches, parameter shrinkage or variable selection or both will need to be employed on $\mathbf{M}$, producing a sparse solution set. The field of Bayesian variable selection is quite large and there are a variety of priors that can be used (see George et al., 1993; Park and Casella, 2008; Carvalho et al., 2010; Li and Lin, 2010, for possible choices)

Assuming model estimation is possible, either formulation provides significant contributions to the data-driven discovery. In contrast to the sparse regression approaches with uncertainty quantification discussed in Section 3.2.2, (3.16) and (3.17) treat the latent process $\mathbf{u}(\mathbf{s},t)$ as a random process and do not disregard the measurement noise when estimating the system. That is, instead of computing derivatives and de-noising prior to model estimation, uncertainty in the derivatives as a product of measurement noise is accounted for. This makes estimation more challenging as the derivatives are no longer assumed known *a prior*. Additionally, missing data can be handled through the incidence matrix $\mathbf{H}$. By formulating the problem within a BHM, known methods accounting for missing data can be used, providing more real-world applicability than the deterministic counterparts.

Reference	Library	System (ODE/PDE)	Type	UQ	Noise	Missing Data	Real Data
Bongard et al. (2007)	Symbolic	ODE	T	No	No	No	Yes
Schmidt et al. (2009)	Symbolic	ODE	T	No	Yes	No	Yes
Maslyaev et al. (2019)	Symbolic	PDE	T	No	Yes	No	No
Brunton et al. (2016)	Sparse	ODE	T	No	Yes	No	No
Rudy et al. (2017)	Sparse	PDE	T	No	Yes	No	No
Rudy et al. (2019)	Sparse	PDE	T	No	Yes	No	No
Schaeffer (2017)	Sparse	PDE	T	No	Yes	No	No
Hirsh et al. (2021)	Sparse	ODE	B	Yes	Yes	No	Yes
Zhang et al. (2018)	Sparse	PDE	B	Yes	Yes	No	No
Yang et al. (2020)	Sparse	ODE	B	Yes	Yes	No	No
Fasel et al. (2021)	Sparse	PDE	BO	Yes	Yes	No	Yes
Both et al. (2021)	Sparse	PDE	NN	No	Yes	No	Yes
Xu et al. (2021)	Symbolic	PDE	NN	No	Yes	Yes	No
Long et al. (2019)	Symbolic	PDE	NN	No	No	No	No
Atkinson et al. (2019)	Symbolic	PDE	NN	No	No	No	Yes
Chapter 4	Sparse	ODE	B	Yes	Yes	Yes	Yes
Chapter 5	Sparse	PDE	B	Yes	Yes	Yes	Yes

Table 3.1: Summary of some discussed papers where the columns are: *Library* - method used to construct the library, *System* - type of system, either ODE or PDE, considered, *Type* - our categorization of the model (combined with library to get the section it is discussed in) where *T* is Traditional, *B* is Bayesian, *BO* is Bootstrap, and *NN* is Neural Network, *UQ* - if uncertainty quantification is considered, *Noise* - if the approach considers or can accommodate measurement noise, *Missing Data* - if the approach considers or can accommodate missing data, *Real Data* - if the approach is illustrated using real data.

3.6 Discussion

While relatively young, the field of data-driven discovery is expanding quickly. The areas that are currently under-studied include properly accounting for uncertainty quantification and missing data and applications of the methods on real-world data sets (see Table 3.1). One method of addressing these issues is the use of statistical methods via Bayesian hierarchical models. BHMs have been extensively used in the statistical literature and are a proven method to account for measurement uncertainty and missing data and have been applied to a variety of real-world problems. However, the extensions discussed using BHMs relies on the same assumptions as the sparse regression approach – the library is pre-specified.

The ability to remove the pre-specified library assumption while retaining the benefits of the statistical approach promises to be a major improvement in the data-driven discovery realm. A recent advance in symbolic regression is the extension to the Bayesian framework (Jin et al., 2019). The incorporation of Bayesian symbolic regression into a BHM could provide the next step to a truly user-free, unbiased, method at data-driven discovery. Additionally, recent advances in deep modeling, where NN have been embedded in the BHM (Zammit-Mangion et al., 2021), could provide a framework where symbolic regression using a NN can be combined with a BHM, providing a alternate method of joining the approaches.

In the following two chapters, we propose a method addressing uncertainty quantification in data-driven discovery of nonlinear dynamic equations (see Table 3.1 for context within the literature). Originally proposed for ODEs, the method is generalized to PDEs, providing the ability to discover a variety of DE within a unified probabilistic framework.

Chapter 4

A Bayesian Approach for Data-Driven Dynamic Equation Discovery of Ordinary Differential Equations

4.1 Introduction

Mathematical modeling using mechanistic dynamic equations (DEs) is a rich and diverse field with many real-world applications. Historically, biology, ecology, epidemiology, economics, and atmospheric and geological sciences, among others, have used DEs to model the evolution of complex processes. Generally, the DE in complex models are derived based on an understanding of the governing dynamics of the system of interest, termed mechanistic modeling, and are usually an approximation of the real-world dynamics. Mechanistic modeling has a long history, dating back to at least Legendre (1806) and Gauss (1809) who infer equations describing the motion of orbital bodies around the sun based on the positions of celestial bodies. More recently, mechanistic modeling has been used in ecology

(Holmes et al., 1994; Hastings, 1996), epidemiology (Zhang et al., 2017; Mangal et al., 2008), pharmacodynamics (Mager et al., 2003; Goutelle et al., 2008), and atmospheric sciences (Zeng et al., 1996; Riley et al., 2002), among many others. Mechanistic modeling typically adopts a deterministic perspective of the system that ignores observational uncertainty, attributing any discrepancy in the estimates to the chaotic nature of the world, and assuming the specified dynamics adequately represent the true system.

Using DEs to motivate statistical models, Berliner (1996), Royle et al. (1999) and Wikle et al. (2001) take a Bayesian hierarchical approach to model the dynamic process in a latent space. Their approach, termed physical-statistical modeling (PSM), motivates dynamic equations through mechanistic relationships and enables researchers to model complex dynamic systems within a statistical framework (Berliner, 2003) (see also Kuhnert, 2017). Modeling the dynamics in a latent space allows the model to separately quantify uncertainty in the observed process and model specification. For example, Wikle et al. (2001) and Milliff et al. (2011) use PSM's to model surface winds over the Equatorial Pacific and Mediterranean Sea, respectively. In both cases, the authors model the surface wind motivated by dynamics supported by the physical understanding of the system. Critically, allowing the parameters in these DE to be random processes (i.e., spatial fields, time series) enables flexible learning and uncertainty quantification. Wikle (2003) and Hooten and Wikle (2008) use this approach with discretized nonlinear reaction-diffusion equations to model the continuous processes for growth and spread of the House Finches across the eastern United states and Eurasian-Collard Doves across the southern United States, respectively. Wikle and Hooten (2010) formalize this approach and define a general framework for linear and nonlinear mechanistically motivated models, termed *General Quadratic Nonlinear* models (GQN). GQN models are able to accommodate many classes of DE systems, and are implemented in a BHM framework that accommodates uncertainty, missing data, and regularization. GQN models are also computationally expensive, can be unstable without careful modeling choices, and often require informative priors. Except in a few specialized cases, the DE used in all of these complex models are approximations of the true underlying dynamics.

Recently, work has been done to discover the governing equation(s) that define dynamic systems in a purely deterministic setting. Seminal work by Bongard and Lipson (2007) and Schmidt and Lipson (2009) presents a new method to dynamic system discovery using symbolic regression. While this approach uncovers nonlinear dynamics that drive data, it is not scalable to large dynamic systems. In order to combat this scaling issue, Brunton et al. (2016) shift the focus of dynamic system discovery to one of sparse identification, proposing the *Sparse Identification of Nonlinear Dynamics* (SINDy) model. SINDy involves three important components: (1) numerical differentiation, (2) determining the candidate functions, termed the "feature library", and (3) sparse regression. Numerical differentiation can also be combined with a de-noising procedure, and is often computed through the finite difference approximation. The feature library is chosen based on the system in question, and represents the maximum feature space to be considered (see Section 4.3.2 below for more detail). Last, sparse regression is performed using an ℓ_1-norm-based algorithm termed "sparse relaxed regularized regression" (Zheng et al., 2019). The basic SINDy model is easily accessible through the Python package *PySINDy* (de Silva et al., 2020).

The basic SINDy approach has been extended in numerous studies. For example, sparse identification was first extended to partial differential equations via the PDE-FIND algorithm (Rudy et al., 2017), and later extended to PDEs with non-constant coefficients (Rudy et al., 2019a). Using a deep feed-forward network, Long et al. (2017) propose a general method to learn PDE dynamics, coined PDE-Net. To account for stochasticity in the evolution process, Boninsegna et al. (2018) consider cases where stochastic differential equations are more appropriate. To improve on the numerical approximation, Schaeffer (2017), Schaeffer et al. (2018), and Lagergren et al. (2020) propose methods to calculate the derivative that are more robust to noise, resulting in better performance.

From a more statistical perspective, Zhang and Lin (2018) and Niven et al. (2020) consider a Bayesian approach to the sparse regression problem and Fasel et al. (2021) take an ensemble approach, allowing for parameter uncertainty quantification. However, to the best of our knowledge, uncertainty is only considered for the parameters, and not the dynamic process.

While SINDy and the related extensions have been shown to perform well, there are two important shortcomings. First, using a multi-step procedure where the de-noising and differentiation are preformed independently of the estimation procedure, uncertainty is not propagated throughout the model. For example, when de-noising and differentiation is done first, the observation uncertainty is not accounted for when modeling the process, and the resulting derivative is assumed to be the truth. Second, in the SINDy class of models, the inherit dependence between the derivative and the dynamic process is not explicitly accounted for when using the multi-step procedure.

To address the issue with the multi-step procedure, Galioto and Gorodetsky (2020) and Yang et al. (2020) take a Bayesian hierarchical modeling approach to nonlinear dynamic discovery. Galioto and Gorodetsky (2020) show how the Kalman filter can be adapted to estimate the state-space of nonlinear systems when the functional form of the system is known, highlighting the advantage of accounting for measurement, system, and parameter uncertainty. Yang et al. (2020) use differentiable programming within a Bayesian context to quantify uncertainty on model parameters, allowing for the dynamic system to be discovered using the library approach similar to SINDy-based methods. However, each of these approaches has limitations. The Kalman approach is unable to identify the functional form of a nonlinear system. For the differentiable programming approach, because the dynamics are not modeled as a latent process, the method cannot handle missing data. Additionally, both numerically estimate the derivative, which can lead to adverse affects when noise is present in the system (see Section 4.2.2 for more discussion).

To address these limitations, we present a Bayesian hierarchical modeling approach for data driven discovery of dynamics explicitly accounting for uncertainty associated with each aspect of the problem. The first significant contribution of this work is that unlike the other data-driven discovery methods presented in the literature, we account for uncertainty in the observed data and represent the dynamic system as a latent process in a multilevel model. The data model accounts for the measurement (or observation) error given the true, but unobserved, latent process. Second, explicitly accounting for the dependence between the derivative and the dynamic process, we model them jointly using a basis expansion with a common set of basis coefficients. Estimating the basis coefficients conditioned jointly on the basis expansion for the derivative and the system builds dependence between the dynamic process and its derivatives. Third, the basis expansion also allows us to compute the derivatives analytically, bypassing the need for a numerical approximation of the derivative. Additionally, because we estimate the dynamic process in a latent space, we model the dynamics as the true underlying process generating the observations. Finally, we explicitly include priors for sparcity in two places in the multi-level model.

Our Bayesian hierarchical approach to data-driven discovery of nonlinear dynamic equations enables uncertainty quantification at all levels of the model (data, process, and parameters). Using basis functions to obtain derivatives, we bypass the need for a multi-step procedure and do not require any pre-filtering of the observed data. However, as in SINDy, we retain the use of the feature library, using variable selection to identify a sparse solution set of the feature library. Modeling the dynamic system as a latent process in the second layer of our hierarchical model allows us to recover the dynamic equations in a statistical framework. By accounting for the dependence structure in the dynamic process, our method does not require as many observations as the machine learning-centric methods. Additionally, our method can handle scenarios with missing data, including sporadic missingness and imperfect data, as well as an entire missing system component.

We illustrate our method's performance on data generated from different well-known dynamic processes with varying levels of measurement noise, missing data, and with a missing component. Through simulation, we find our approach to be robust to measurement noise and able to learn the dynamics of complex dynamical systems. We also apply our method to a real-world application and recover dynamics consistent with the theoretical physics of the systems.

The remainder of this article is organized as follows. In Section 4.2 we give background on the general dynamical system, state how we make inference on the derivative of the system, and present the Bayesian hierarchical model. In Section 4.3, we describe parameter estimation and discuss modeling choices. In Section 4.4, we test our method on multiple simulated data sets and perform inference on three real-world data sets. Section 4.5 concludes the chapter.

4.2 Bayesian Dynamic Equation Discovery

Here, we propose a general hierarchical model for making inference on nonlinear dynamic systems. Analogous to the observation and state model in state-space modeling of time series (e.g., Shumway and Stoffer, 2017), we consider the dynamic process to be latent and the observed data to be a noisy realization of the "true" underlying process. As is customary in hierarchical modeling, we specify the three components of the model, namely the data model, the process model, and the parameter models in the sections below. In Section 4.2.1 we motivate the dynamic systems and describe in detail the components of the latent process. Section 4.2.2 describes how we use basis functions to approximate the latent process and obtain derivatives of the system. We specify the data model in Section 4.2.3, and specify the prior distributions in Section 4.2.4.

4.2.1 Dynamic System

Consider the ordinary differential equation (ODE) dynamic system

$$\frac{d^{(J)}}{dt^{(J)}}\mathbf{u}(t) = \mathbf{u}_{t^{(J)}}(t) = M(\mathbf{u}_{t^{(0)}}(t), \mathbf{u}_{t^{(1)}}(t), ..., \mathbf{u}_{t^{(J-1)}}(t)), \tag{4.1}$$

where the vector $\mathbf{u}(t) \in \mathbb{R}^N$ denotes the realization of the system at time t, the function $M(\cdot)$ represents the (potentially nonlinear) evolution function, and $\mathbf{u}_{t^{(j)}}(t), j = 0, ..., J$ represents the j^{th} derivative of $\mathbf{u}(t)$. Equations of the form of (4.1) are often used to model processes in biology, ecology, climatology, epidemiology, economics, meteorology, pharmacodynamics, and geological sciences, among others.

For illustration, we consider mechanistic systems that only have a few relevant terms that govern the dynamics (e.g., the pendulum equations, Lorenz attractor, Lotka–Volterra model; Higham et al., 2016) so the function space of $M(\cdot)$ will be sparse. We can reparameterize (4.1) to be intrinsically linear (in parameters) as

$$\mathbf{u}_{t^{(J)}}(t) = \mathbf{M}\mathbf{f}(\mathbf{u}_{t^{(0)}}(t), \mathbf{u}_{t^{(1)}}(t), ..., \mathbf{u}_{t^{(J-1)}}(t)), \tag{4.2}$$

where $\mathbf{M}$ is a $N \times D$ *sparse* matrix of coefficients and $\mathbf{f}(\cdot)$ is a vector-valued nonlinear transformation function of length D. The inputs of the function $\mathbf{f}(\cdot)$ contain arguments that potentially relate to the dynamic system (i.e., more than just the lower order terms of the system). That is, the functions $f_i(\cdot), i = 1, ..., D$ are any functions that *potentially* represent (4.1) (e.g., polynomials, sinusoids, interactions). Crucially, these functions are chosen based on an educated understanding of the general properties of system in question (e.g., diffusion, advection, growth), with the *hope* that all the correct terms in the "true" system are included. Thus, D can be quite large and depending on the number of hypothesized functions chosen, (4.2) has the potential to be drastically over-parameterized. As such, a method to induce sparseness in $\mathbf{M}$ will be required.

As an example, consider the Lotka-Volterra system (Lotka, 1920),

$$\begin{aligned}\frac{dx}{dt} &= \alpha x - \beta xy \\ \frac{dy}{dt} &= \delta xy - \gamma y,\end{aligned}$$

where $\mathbf{u}(t) \equiv [x(t), y(t)]$, x is the number of prey, y is the number of predators, α is the prey population growth rate, β is the rate of predation, δ is the predator population growth, and

γ is the death rate of the predator population. In terms of (4.2), this can be represented as

$$\mathbf{u}_t(t) = \begin{pmatrix} \alpha & 0 & -\beta \\ 0 & -\gamma & \delta \end{pmatrix} \begin{pmatrix} x_t \\ y_t \\ x_t y_t \end{pmatrix}.$$

However, because we generally do not know $\mathbf{f}(\mathbf{u}(t)) = [x(t), y(t), x(t)y(t)]'$, we specify $\mathbf{f}(\mathbf{u}(t))$ in terms of possible solutions to the function (e.g., all polynomials up to the third order, sinusoids, etc.). Then, by selecting against coefficients in $\mathbf{M}$ (i.e., identifying the terms that should be zero) we recover the solution to the dynamic equation.

In real-world problems, (4.2) does not hold exactly. Stochastic forcing could perturb the system (e.g., weather systems, demographic stochasticity) or there could be error in the model specification. We accommodate this unknown stochasticity including an additive error term

$$\mathbf{u}_{t^{(J)}}(t) = \mathbf{M}\mathbf{f}(\mathbf{u}_{t^{(0)}}(t), \mathbf{u}_{t^{(1)}}(t), ..., \mathbf{u}_{t^{(J-1)}}(t)) + \boldsymbol{\eta}(t), \tag{4.3}$$

where, for example, $\boldsymbol{\eta}(t) \overset{\text{i.i.d.}}{\sim} N(\mathbf{0}, \boldsymbol{\Sigma}_U)$ is a mean zero Gaussian process with variance/-covariance matrix $\boldsymbol{\Sigma}_U$.

4.2.2 Basis Expansion

Define the expansion of the nth element of $\mathbf{u}(t)$ as

$$u(t,n) = \sum_{k=1}^{\infty} A(n,k)\phi(t,k), \quad n = 1, ..., N,$$

where $\{\phi(t,k) : k = 1,2,...\}$ are basis function that are differential to (at least) order J and defined at any time t, and $\{A(n,k) : k = 1,2,...\}$ are the associated basis coefficients. To reduce the dimension and transition to discretely observed time data, we keep the first $k = 1,...,p_a$ terms and define $\phi(t,k)$ at finite times $t = 1,...,T$. Let $\mathbf{U} \approx \mathbf{A}\boldsymbol{\Phi}$, where $\mathbf{U} = \{\mathbf{u}(t)\}_{t=1,...,T}$ is an $N \times T$ matrix, $\boldsymbol{\Phi}$ is a $p_a \times T$ matrix of differentiable basis functions where each column is given by $\boldsymbol{\phi}(t) \equiv (\phi(t,1),...,\phi(t,p_a))$, and $\mathbf{A}$ is the $N \times p_a$ matrix of basis coefficients with columns given by $\mathbf{A}(k) \equiv (A(1,k),...,A(N,k))$. We can then analytically obtain higher order derivatives of the elements of $\mathbf{U}$ by taking derivatives of the basis functions. Specifically, let $\mathbf{U}_{t^{(j)}} \approx \mathbf{A}\boldsymbol{\Phi}_{t^{(j)}}, j = 0,...,J$, where $\boldsymbol{\Phi}_{t^{(j)}}$ is a $p_a \times T$ matrix of the jth derivative of known basis functions $\{\boldsymbol{\phi}_{t^{(j)}}(t)\}$ (e.g., the tth column of $\boldsymbol{\Phi}_{t^{(j)}}$ is $\boldsymbol{\phi}_{t^{(j)}}(t) \equiv (\phi_{t^{(j)}}(t,1),...,\phi_{t^{(j)}}(t,p_a)))$. For time t, $\mathbf{u}_{t^{(j)}}(t) = \mathbf{A}\boldsymbol{\phi}_{t^{(j)}}(t)$ with $\boldsymbol{\phi}_{t^{(j)}}(t) \in \mathbb{R}^{p_a}$ and (4.3) can be rewritten,

$$\mathbf{A}\boldsymbol{\phi}_{t^{(J)}}(t) = \mathbf{M}\mathbf{f}(\mathbf{A}, \boldsymbol{\phi}_{t^{(0)}}(t),...,\boldsymbol{\phi}_{t^{(J-1)}}(t)) + \boldsymbol{\eta}_t.$$

In summary, decomposing $\mathbf{U}$ using temporal basis function expansions accomplishes two tasks. First, it enables inference on the derivative of the process, $\{\mathbf{u}_{t^{(j)}}(t)\}$, when only the process, $\{\mathbf{u}_{t^{(0)}}(t)\}$, is observed. Because $\mathbf{U}_{t^{(j)}}$ is decomposed in terms of $\mathbf{A}$ for $j = 0,...,J$, the estimate of $\mathbf{A}$ is jointly informed by the system and the derivatives, allowing for information to be shared between the system and the derivatives. Second, by keeping $p_a \ll T$ basis functions, the resulting reconstruction of $\mathbf{A}\boldsymbol{\Phi}_{t^{(0)}}$ is smooth (Wang et al., 2016) (note, this implies $\boldsymbol{\eta}_t$ now also includes truncation error). This is important because numerically estimating the derivative (e.g., via a finite difference) when the dynamic process $\{\mathbf{u}_{t^{(j)}}(t)\}$ is noisy can amplify the noise of the higher order terms in the system (e.g., $\{\mathbf{u}_{t^{(j)}}(t)\}$ for $j = 1,...,J$, Chartrand, 2011). By taking derivatives analytically through basis functions, the system is more robust to noise.

4.2.3 Data Model

We assume $\mathbf{v}(t)$ is an observation of the latent process $\mathbf{u}_{t(0)}(t)$ outlined in Section 4.2.2 with unknown measurement uncertainty. We model $\mathbf{v}(t)$ using a generalization to the traditional linear data error model that links the dynamics to the observed process (e.g., see Cressie and Wikle, 2011, Chapter 7). That is, we model

$$\mathbf{v}(t) = \mathbf{H}(t)\mathbf{u}_{t(0)}(t) + \widetilde{\varepsilon}(t),$$

where $\mathbf{v}(t) \in \mathbb{R}^{L(t)}$ and $\mathbf{H}(t)$ is a $L(t) \times N$ matrix that maps the latent process to the observed process and accounts for possible missing observations at time t. Uncertainty in the observations of the process are captured by $\widetilde{\boldsymbol{\epsilon}}(t) \overset{\text{indep.}}{\sim} N_{L(t)}(\mathbf{0}, \widetilde{\boldsymbol{\Sigma}_V}(t))$, where $\widetilde{\boldsymbol{\Sigma}_V}(t)$ is the variance/covariance matrix.

Missing data are common in applications. A benefit of the hierarchical model is that it can easily accommodate missing data. Since the latent process is fully specified and missing data are handled at the data level, missing data do not impact the process model. We handle scenarios with missing data by allowing the dimension of $\mathbf{H}(t)$ to vary in time. If there are no missing data at time t, then $L(t) = N$ and $\mathbf{H}(t) = \mathbf{I}_N$. When one or more system components are missing data, then the row corresponding to the missing system component is removed. For example, if we have a three-dimensional system, say $\mathbf{u}(t) = [a(t), b(t), c(t)]$

and the observation component for $b(t)$ is missing at time t, then

$$\mathbf{H}(t) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

This representation can also accommodate situations where an entire system component is not observed. Again, $\mathbf{H}(t)$ is chosen such that the latent system, of dimension N, can map to the observation system of dimension $L(t) < N$ (recall, $\mathbf{H}(t)$ is a $L(t) \times N$ matrix). We then allow the process model to learn the missing dynamic process based purely on the dependence that is present within the process model. However, as we will discuss in more depth through the Lorenz attractor example in Section 4.4.1, there are limitations to the extent of missing information that can be accommodated and care needs to be taken when interpreting these cases.

4.2.4 Parameter Model

Combining the process and observation equations results in the first two levels of our proposed Bayesian hierarchical model. As defined in Section 4.2.2 and 4.2.3, for discrete time points $t = 1, ..., T$, the first two layers of our general model are

$$\begin{aligned} \mathbf{v}(t) = \mathbf{H}(t)\mathbf{u}_{t^{(0)}}(t) + \widetilde{\boldsymbol{\epsilon}}(t) = \mathbf{H}(t)\mathbf{A}\boldsymbol{\phi}_{t^{(0)}}(t) + \boldsymbol{\epsilon}(t) \\ \mathbf{A}\boldsymbol{\phi}_{t^{(J)}}(t) = \mathbf{M}\mathbf{f}(\mathbf{A}, \boldsymbol{\phi}_{t^{(0)}}(t), ..., \boldsymbol{\phi}_{t^{(J-1)}}(t)) + \boldsymbol{\eta}(t), \end{aligned} \tag{4.4}$$

where $\boldsymbol{\epsilon}(t) \overset{indep.}{\sim} N_{L(t)}(\mathbf{0}, \boldsymbol{\Sigma}_V(t))$ and $\boldsymbol{\Sigma}_V(t)$ is the $L(t) \times L(t)$ measurement error covariance matrix where $L(t)$ is the dimension at time t (Figure 4.1). For clarity, we present the details the model parameters in Table 4.1. Our goal is to make inference on the unknown param-

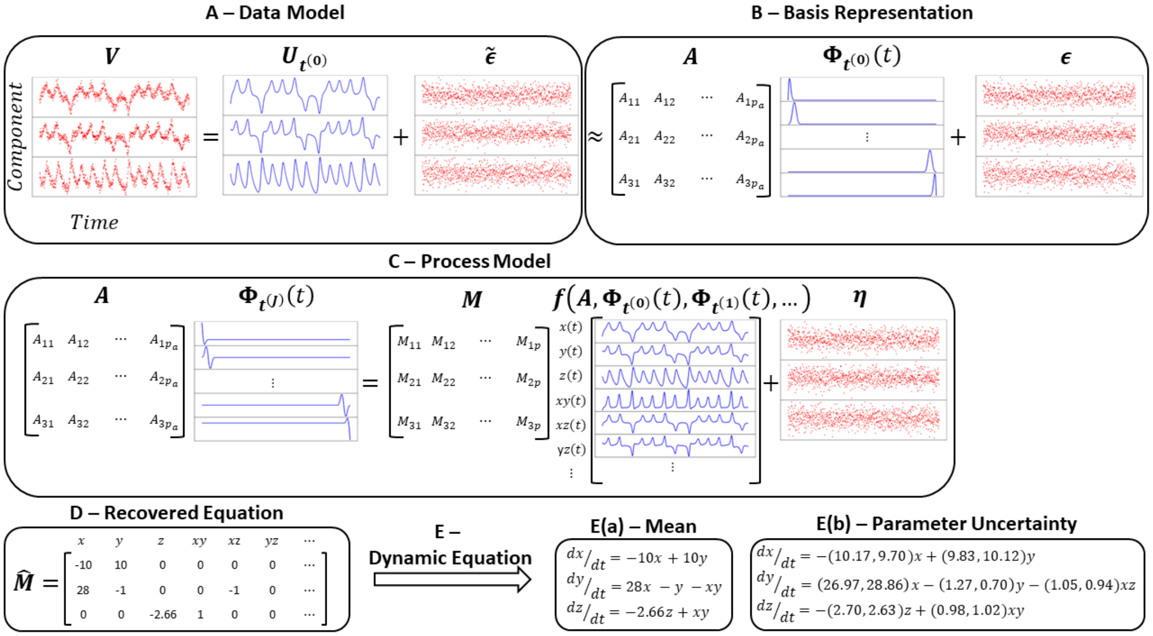


Figure 4.1: (A) Data model relating the observed measurements $\mathbf{V}$ to the latent dynamic process $\mathbf{U}_{t^{(0)}}$ and accounting for measurement error $\widetilde{\boldsymbol{\epsilon}}$. Note, we do not include $\mathbf{H}$ in our pictorial representation of the equation and the error is not to scale. (B) Basis representation of the dynamic process where $\mathbf{U}_{t^{(0)}} \approx \mathbf{A}\mathbf{\Phi}_{t^{(0)}}$ and $\boldsymbol{\epsilon}$ now accounts for the approximation uncertainty. (C) Process model where the derivative of the dynamic process $\mathbf{A}\mathbf{\Phi}_{t^{(0)}}$ is related to the product of the parameter coefficient matrix $\mathbf{M}$ and the library of function $\mathbf{f}(\cdot)$ plus model uncertainty $\boldsymbol{\eta}$. (D) The recovered equation which is computed from $\mathbf{M}$ after the model parameters have been estimated. (E) Resulting dynamic equation mean (a) and equation uncertainty (b) which are computed from $\mathbf{M}$ after the model parameters have been estimated.

eters $\mathbf{M}, \mathbf{A}, \mathbf{\Sigma}_U$, and $\mathbf{\Sigma}_V$, where $\mathbf{M}$ defines the nonlinear dynamic equation, $\mathbf{A}$ defines the smooth latent process, $\mathbf{\Sigma}_U$ captures the error dependencies within the dynamic equation, and $\mathbf{\Sigma}_V$ captures the measurement uncertainty associated with the observed process. To complete our Bayesian hierarchical model, we define the following priors.

As mentioned in Section 4.2.1, $\mathbf{M}$ has the potential to be over-parameterized. To induce sparcity into our estimate of $\mathbf{M}$, we use the stochastic search variable selection (SSVS,

Model	Symbol	Description	Dimension
	Variable		
Data	$\mathbf{v}(t)$	Observed data	$L(t) \times 1$
Data	$\mathbf{H}(t)$	Mapping matrix	$L(t) \times N$
Data	$\boldsymbol{\epsilon}(t)$	Data uncertainty distribution	$L(t) \times 1$
Data	$\boldsymbol{\Sigma}_V(t)$	Measurement error covariance matrix	$L(t) \times L(t)$
Process	$\mathbf{u}_{t(0)}(t)$	Dynamic process	$N \times 1$
Process	$\mathbf{A}$	Basis coefficients	$N \times p_a$
Process	$\boldsymbol{\phi}_{t(j)}(t)$	jth order basis function at time t	$p_a \times 1$
Process	$\mathbf{M}$	Dynamic evolution matrix	$N \times D$
Process	$\mathbf{f}(\cdot)$	Feature library	$N \times 1$
Process	$\boldsymbol{\eta}$	Process uncertainty distribution	$N \times 1$
Process	$\boldsymbol{\Sigma}_U$	Dynamic equation error covariance matrix	$N \times N$
	Dimension		
	T	Number of observed time points	1
	$L(t)$	Dimension of observation vector at time t	1
	N	Dimension of latent process (dynamic system)	1
	D	Number of library functions	1
	p_a	Number of basis functions	1
	J	Highest order derivative in the dynamic system	1
	Indices		
	t	Time interval, $t = 1, ..., T$	1
	j	Order of the derivative, $j = 1, ..., J$	1

Table 4.1: List of symbols used in the Bayesian hierarchical model.

George et al., 1993) prior. Specifically,

$$vec(\mathbf{M}) \sim N_{ND}(\mathbf{0}, \boldsymbol{\Sigma}_M),$$
$$\boldsymbol{\Sigma}_M = diag(\gamma_1^{(c_1)}, ..., \gamma_{ND}^{(c_{ND})}),$$

where $\gamma_l^{(c_l)} = v_1$ if $c_l = 1$ and $\gamma_l^{(c_l)} = v_0$ if $c_l = 0$. The latent variable, c_l, is the inclusion indicator, and the posterior of c_l specifies the probability of inclusion for any parameter in $\mathbf{M}$. The hyperpriors v_0 and v_1 are chosen such that v_0 is small (e.g., $v_0 = 10^{-6}$) and v_1 is large (e.g., $v_1 = 10^4$). Whereas, any method to induce sparsity in $\mathbf{M}$ can be used, we choose

SSVS because it provides the inclusion probabilities and has been shown to work well in nonlinear dynamic models (Wikle and Holan, 2011).

Within the SSVS prior, v_0 and v_1 determine how parsimonious the selected model will be. This is due to the ratio

$$p_l = \frac{\pi[m(l)|c_l = 1, \cdot]}{\pi[m(l)|c_l = 1, \cdot] + (1-\pi)[m(l)|c_l = 0]}$$

for $\mathbf{m} \equiv vec(\mathbf{M})$ and $l = 1, ..., ND$, which determines the inclusion probability for $m(l)$, where $[m(l)|\cdot]$ denotes the distribution of $m(l)$ given all relevant parameters. George et al. (1993); George and McCulloch (1997); George et al. (2008) discuss the specification of v_0, v_1 in detail, and we summarize some of the key points here. One should choose v_0 and v_1 such that if $c_l = 0$, $m(l)$ can safely be replaced with zero, and if $c_l = 1$, $m(l)$ then a non-zero estimate should be included with some probability p_i. However, in practice we do not know which values of $\mathbf{m}$ should or should not be included. As a general rule, we found $v_0 = 10^{-6}$ and $v_1 = 10^4$ work well for most of the simulations we present. However, when models have small parameter values (e.g., see the SIR example in Section 4.4), we find smaller values, such as $v_0 = 10^{-8}$ and $v_1 = 10^2$, are needed.

Both $\mathbf{\Sigma}_V(t)$ and $\mathbf{\Sigma}_U$ have the potential to have small parameter values, and inference using traditional conjugate Inverse Gamma/Wishart priors are overly sensitive to the choice of hyperpriors when estimates are small (Gelman, 2006). Instead, we use the conjugate Half-t prior proposed by Huang and Wand (2013) for covariance estimation, which imposes less prior information and does not have as strong of influence on small estimates.

We restrict the measurement error to be diagonally structured (although this restriction can be removed if warranted) since it is often assumed that measurement noise is independent (Cressie and Wikle, 2011). Let $\mathbf{\Sigma}_V(t) = \mathbf{H}(t) diag(\sigma^2(1), ..., \sigma^2(N)) \mathbf{H}(t)'$, where each diagonal element, $\sigma^2(1),...,\sigma^2(N)$, is assigned a conjugate Half-t$(2,10^{-5})$ prior. In order to account for system dependence within the multivariate latent process error, we model $\mathbf{\Sigma}_U$ as a full rank matrix, which enables us to borrow strength across systems and improve model performance. We assign the matrix Half-t(ν_k, B_k) prior to $\mathbf{\Sigma}_U$ with $\nu_k = 2, B_k = 10^{-5}, k = 1,...,N$.

Last, we specify the Bayesian elastic net prior (Li and Lin, 2010) on $\mathbf{A}$. Specifically, our prior is

$$\pi(\mathbf{A}) \propto \exp\{-\lambda_1 \|\mathbf{A}\|_1 - \lambda_2 \|\mathbf{A}\|_2^2\},$$

where λ_1, λ_2 are penalty parameters. We use the elastic net prior to help regularize the basis coefficients and select against unneeded basis functions. It is possible to specify hyperpriors for the two penalty terms, but we find inference is not overly sensitive to the choice of penalty parameters and fix them to a small value (e.g., 0.1 or 0.01).

4.3 Algorithm and MCMC

There are five full-conditional distributions of interest, $[\mathbf{M}|\cdot]$, $[\mathbf{\Sigma}_V|\cdot]$, $[\mathbf{\Sigma}_U|\cdot]$, $[\mathbf{A}|\cdot]$, and $[\mathbf{c}|\cdot]$ (see Appendix C.1 for the details of the distributions) when performing MCMC inference for this model. The four components $\mathbf{M}, \mathbf{\Sigma}_V, \mathbf{\Sigma}_U$ and $\mathbf{c}$ are updated using classical Bayesian methods and $\mathbf{A}$ is updated using a stochastic gradient approach. We present the general MCMC procedure in Algorithm 1. Within the general MCMC procedure, there are implementation details that warrant a more detailed discussion. Additionally, because the framework we present is general, some modeling choices are problem specific. We discuss how to address these challenges Section 4.3.1.

Algorithm 1: MCMC Sampling Algorithm

```
Require $\mathbf{V}, \mathbf{f}(\cdot), p_\alpha, \Delta t, |\mathscr{X}|, \kappa, \nu_0, \nu_1$
Initialize all parameter values
for $\ell = 1, 2, \ldots$ till convergence do
    1. Sample $\mathbf{M}^{(\ell)}$ from $[\mathbf{M}|\cdot]$
    2. Sample $c_i^{(\ell)}$ from $[c_i|\cdot]$ for $i = 1, \ldots ND$
    3. Sample $\mathbf{\Sigma}_V^{(\ell)}$ from $[\mathbf{\Sigma}_V|\cdot]$
    4. Sample $\mathbf{\Sigma}_U^{(\ell)}$ from $[\mathbf{\Sigma}_U|\cdot]$
    5. Sample $\mathbf{A}^{(\ell)}$ from (4.5)
end
```

4.3.1 Basis Estimation

The basis coefficients pose an estimation challenge because they are embedded in the non-linear function $\mathbf{f}(\cdot)$ and since $\mathbf{f}(\cdot)$ is problem specific, it needs to be specified generally to accommodate different problems. In principle, an Expectation-Maximization (EM) or Metropolis-Hastings (MH) algorithm could be used to estimate $\mathbf{A}$, but they require $\mathbf{f}(\cdot)$ to be known and convergence with either of these methods is slow in our setting. Instead, we use an adapted version of SGD with a constant learning rate (SGDCL; Mandt et al., 2016), which has been shown to scale well.

As with SGD, SGDCL relies on the gradient of a loss function and a learning rate. For SGDCL, the loss function is the negative log posterior for our parameters of interest, $\mathbf{A}$. Here, the loss function for a single time is

$$\mathscr{L}(t) = -\log([\mathbf{v}(t)|\mathbf{H}(t), \mathbf{A}, \phi_{t^{(0)}}(t), \mathbf{\Sigma}_V(t)][\mathbf{A}, \phi_{t^{(J)}}(t)|\mathbf{M}, \mathbf{A}, \phi_{t^{(0)}}(t), \ldots, \phi_{t^{(J-1)}}(t), \mathbf{\Sigma}_U]) + \frac{1}{T}\left(\lambda_1 \|\mathbf{A}\|_1 + \lambda_2 \|\mathbf{A}\|_2^2\right).$$

The gradient of the loss function is dependent on $\frac{\partial}{\partial \mathbf{A}}\mathbf{f}(\mathbf{A},\phi_t^{(0)},...,\phi_t^{(J-1)})$, which we generically denote as $\dot{\mathbf{F}}_t$, and the gradient of $\mathscr{L}(t)$, $\frac{\partial \mathscr{L}(t)}{\partial \mathbf{A}} = \nabla_{\mathbf{A}}\mathscr{L}(t)$, is

$$\begin{aligned}\nabla_{\mathbf{A}}\mathscr{L}(t) = & -\mathbf{H}(t)\mathbf{\Sigma}_V^{-1}(t)\mathbf{v}(t)\phi_{t^{(0)}}(t)' + \mathbf{H}(t)'\mathbf{\Sigma}_V^{-1}(t)\mathbf{H}(t)\mathbf{A}\phi_{t^{(0)}}(t)\phi_{t^{(0)}}(t)' \\ & + \mathbf{\Sigma}_U^{-1}\mathbf{A}\phi_{t^{(J)}}(t)\phi_{t^{(J)}}(t)' - \mathbf{\Sigma}_U^{-1}\mathbf{M}\mathbf{f}(t)\phi_{t^{(J)}}(t)' - \phi_{t^{(J)}}(t)'\mathbf{A}'\mathbf{\Sigma}_U^{-1}\mathbf{M}\dot{\mathbf{F}}(t) \\ & + \mathbf{f}(t)'\mathbf{M}'\mathbf{\Sigma}_U^{-1}\mathbf{M}\dot{\mathbf{F}}(t) + \frac{1}{N}\left(\lambda_1 sign(\mathbf{A}) + 2\lambda_2\mathbf{A}\right).\end{aligned}$$

SGDCL methods replace the true gradient with the stochastic estimate,

$$\widehat{\nabla\mathscr{L}}_{\mathscr{X}}(t) = \frac{1}{\mathscr{X}}\sum_{t\in\mathscr{X}}\nabla_{\mathbf{A}}\mathscr{L}(t),$$

where $\mathscr{X} \subset \{1,...,T\}$ is a random subset of the observations, called a mini-batch, and $|\mathscr{X}|$ is the cardinality of the set. Within the context of a MCMC algorithm, the ℓth update of $\mathbf{A}$ is given by

$$\mathbf{A}^{(\ell)} = \mathbf{A}^{(\ell-1)} - \kappa\widehat{\nabla\mathscr{L}}_{\mathscr{X}^{(\ell)}}(\mathbf{A}^{(\ell-1)}), \tag{4.5}$$

where $\mathscr{X}^{(\ell)}$ denotes a random minibatch specific to the ℓ update and κ is the learning rate. Mandt et al. (2016) show how to select the constant κ, or a preconditioning matrix (i.e, replace κ with a matrix $\mathbf{K}$), to match the stationary distribution to the posterior. In practice, we find κ is problem dependent. If there is a lot of observation noise in the data, an adaptive approach may provide the best results. Specifically, an upper bound is specified for the learning rate. Then, during the burnin process, the learning rate decreases at equal intervals from this initial value to a specified lower bound. After burnin, the learning rate stays fixed at the specified lower bound throughout the sampling algorithm. If there is minimal to no noise, then a fixed small value for κ for the entirety of the sampler works best.

The final challenge to estimating $\mathbf{A}$ is computing $\dot{\mathbf{F}}(t)$. Because $\mathbf{f}(\cdot)$ is problem specific, $\dot{\mathbf{F}}(t)$ is also problem specific and needs to be obtained generally. To overcome this issue, we use automatic differentiation (AD). AD has become increasingly popular, especially with the increasing interest in deep models, and allows one to analytically compute the derivative of $\mathbf{f}(\cdot)$. There are many different libraries and programs that perform AD, and for our implementation we use the *ForwardDiff* (Revels et al., 2016) package in Julia (Bezanson et al., 2017).

Note that we need to estimate the latent process $\mathbf{U}_{t^{(0)}}$, and all subsequent derivatives $\mathbf{U}_{t^{(J)}}$ for $j = 1,...,J$ in the model. Without using a basis expansion approach, estimating each of these processes requires an $O(T)$ calculation. With the basis expansion, this reduces the computational burden to $O(p_a)$ for each process. We further reduce the computation required using the SGDCL to $O(|\mathscr{X}|)$, where $|\mathscr{X}| \ll p_a \ll T$.

4.3.2 Choice of Functional Library

Choosing the potential solutions (the function library $\mathbf{f}(\cdot)$) generally requires some extra thought. Ideally, the functions are chosen based on a general physical understanding of the system (e.g., diffusion, advection, growth). However, this is not always possible. In general, most ordinary differential equations are functions of polynomials and interactions (e.g., Lorenz attractor, van der Pol oscillator, Lotka–Volterra model; Higham et al., 2016). Because of this, we default to a using a library of polynomial functions and interactions when a physical understanding of the system is not applicable. While there are scenarios where more terms need to be included in the library (e.g., sinusoidal terms), using polynomials and interactions as a default library covers a wide range of potential systems.

4.3.3 Choice of Basis Functions

The choice of basis functions have the potential to affect the model fit. Ramsay and Silverman (2005, Chapter 3) provide a discussion on how to choose basis functions based on the "shape" of the data, and we will summarize some key points here. For our method we consider two basis function classes, the B-spline and Fourier basis, with the B-spline being a local basis function and Fourier a global basis function. While we only discuss the B-spline and Fourier basis, other basis functions could be chosen. The Fourier basis is best suited for periodic data with no strong local features and where the curvature of the function is the same order everywhere. In contrast, the B-spline basis works best with non-periodic functions that may or may not have strong local features. With respect to differentiation, the Fourier basis is infinitely differentiable, and the m order B-spline basis is differentiable up to order $m-1$. However, Fourier series suffer from a ringing effect (Hewitt and Hewitt, 1979), and we find the effect is worsened when derivatives greater than the first order are considered or there are local regions with little curvature. This issue makes the Fourier series less useful in practice. B-splines do not suffer from the ringing effect, making them better suited for higher order dynamic systems. The amount of noise in the data also impact the choice of basis. For both local and global basis functions, enough basis functions need to be included so the estimated solution curve is flexible, the dynamics are captured, and the posterior latent space is properly explored, but not so many such that unnecessary noise is captured (see examples below for relation of number of basis functions to number of observations). In general, we found the B-spline basis resulted in the best model performance and use them for all of our simulations and examples.

4.4 Simulations and Examples

Here, we show our proposed model's ability to detect the dynamic equation on four simulated data sets and on three real world data sets. Unless otherwise stated, all reported estimates are rounded to three significant digits for readability and significant terms are shown in bold within their respective tables. As stated in Section 4.3.1, when noise is present in the system, we specify an initial upper bound for the learning rate and decay to a lower bound during the burn-in phase. Where applicable, we denote the upper bound for the learning rate as κ_u and the lower bound as κ_l. Generally, we only specify two learning rates for problems with excessive noise, where the initial large learning rate preforms large scale learning and the smaller learning rate limits excessive noise (from keeping a large learning rate) from being artificially injected into the system. If a lower and upper bound are not specified, then the specified κ is constant for the duration of the sampler. For all simulations and real-world examples, we obtain 10000 posterior samples and discard the first 5000 as burn-in. Convergence of model parameters was assessed visually via trace plots, with no issues detected.

4.4.1 Simulations

We show our ability to recover nonlinear equations on data simulated from four systems: the Susceptible, Infected, Removed (SIR) epidemic model, the Lotka-Volterra (or predator-prey) system, a coupled pendulum, and the Lorenz-63 attractor. To simulate data from the nonlinear systems we use a 4th-order Runge-Kutta (RK4) method. To simulate measurement noise, we add mean zero Gaussian noise to the state vector; specifically $\mathbf{v}(t) + \boldsymbol{\epsilon}(t)$, where $\mathbf{v}(t)$ is the simulated data, $\boldsymbol{\epsilon}(t) \sim N(\mathbf{0}, \xi \mathbf{I}_N)$ is the additional noise, and ξ is the mag-

Model	Parameter Values	Initial Values	Δt	Time Range
SIR	$\beta = 15, \gamma = 0.9$ $n = 100$	$[S, I, R]' = [99, 1, 0]'$	0.01	$[0, 3]$
Lotka-Volterra	$\alpha = 1.1, \beta = 0.4$ $\delta = 0.1, \gamma = 0.4$	$[x, y]' = [10, 10]'$	0.1	$[0, 100]$
Coupled Pendulum	$g = 9.8, l = 1$ $k = 1, m = 1$	$[\theta_1, \theta_2]' = [\pi/2, -\pi/4]'$	0.01	[0,10]
Lorenz-63	$\sigma = 10$, $\rho = 8/3$ $\beta = 28$	$[x, y, z]' = [-8, 7, 27]'$	0.01	[0,10]

Table 4.2: Parameter values, initial values, time step, and time range for each simulated data set.

nitude of the noise variance. Using the Lotka-Volterra system, SIR model, and Lorenz-63 attractor, we will show how the method performs with varying amounts of measurement noise. In addition, for the Lorenz-63 attractor, we will show how the model performs when data are missing sporadically or when an entire system component is missing. Last, we will compare our method to SINDy using data simulated from the Lorenz-63 attractor. For all the simulated data, the parameter values, initial values, time step, and time range are given in Table 4.2.

SIR Model

Susceptible, Infected, and Removed (SIR) models are commonly used to model infectious diseases (Kermack and McKendrick, 1927). At their core, SIR models relate the number of individuals in a population to the number of infected and removed individuals through infection and removal rates. The number of susceptible (S), infectious (I), and removed (R)

individuals are related by the nonlinear ODEs:

$$\begin{aligned} \frac{dS}{dt} &= -\frac{\beta}{n} IS \\ \frac{dI}{dt} &= \frac{\beta}{n} IS - \gamma I \\ \frac{dR}{dt} &= \gamma I \end{aligned} \tag{4.6}$$

where β is the exposure rate, γ is the removal rate, and n is the total population. We simulate from (4.6) with the model and simulation parameters given in Table 4.2 with no measurement noise ($\xi = 0$) and measurement noise ($\xi = 1$). If a simulated value with noise is below zero, which is not biologically possible, we set the value to zero. The data are shown in Figure 4.2, where the x-axis and y-axis correspond to a hypothetical time period and population, respectively. The data with no measurement noise are represented by the solid lines and the data with measurement noise are represented by the solid points.

For both simulated data sets, we fit the proposed model with parameters $p_\alpha = 200, |\mathscr{X}| = 20, \nu_0 = 10^{-8}, \nu_1 = 10^2$. With no measurement noise our learning rate is $\kappa = 10^{-8}$ and with measurement noise we specify $\kappa = 1$. Our library of potential solutions are all polynomials up to the third order with all possible interactions except we do not include the removed term in our library because biologically it is not plausible. After obtaining posterior samples, we keep only terms with greater than a 99% inclusion probability. The recovered equations and 95% credible intervals without and with measurement noise for the included terms are shown in Tables 4.3 and 4.4, respectively. For both situations we correctly identify the components of the dynamic system and the credible intervals of all parameters cover the truth.

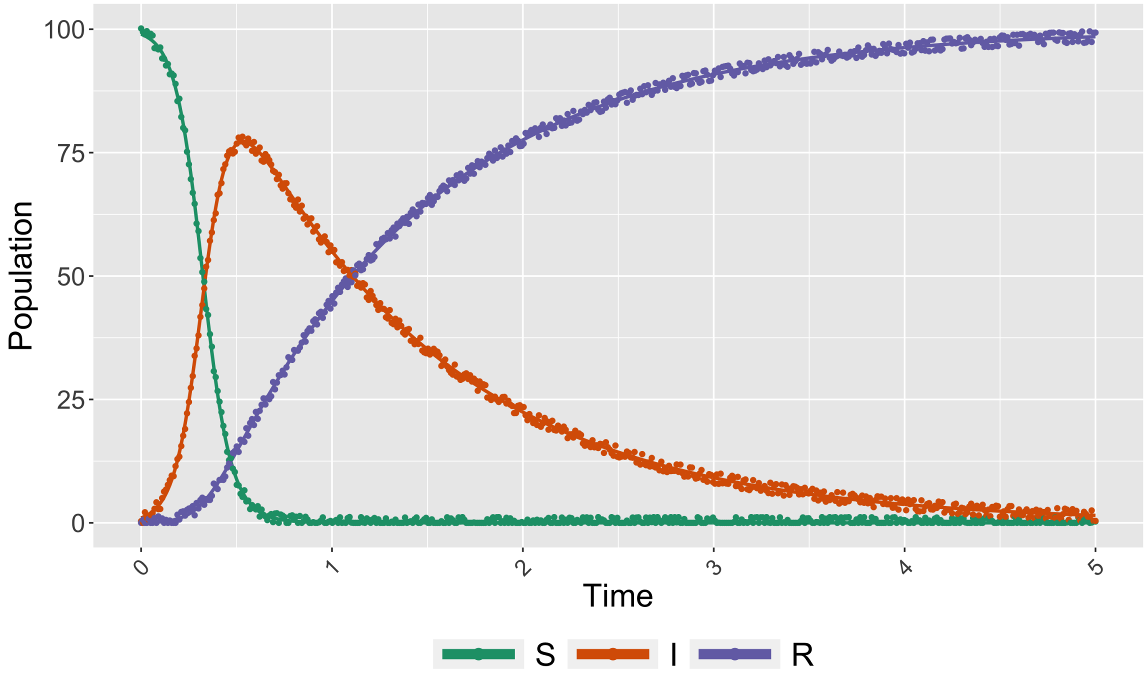


Figure 4.2: Data simulated from the SIR model, (4.6) with parameters $n = 100, \beta = 15, \gamma = 0.9$, and initial condition $[S, I, R]' = [99, 1, 0]'$ for times $t = 0$ to $t = 3$ with a time step of $\Delta t = 0.01$. The solid lines represent the true system with no measurement noise ($\xi = 0$), and the dots are the data with measurement noise ($\xi = 1$).

Lotka-Volterra System

The Lotka-Volterra (LV) equations are often used in biological modeling to describe the dynamics of two (or more) interacting species, commonly a predator-prey interaction. The LV equation consists of the two nonlinear ODEs:

$$\begin{aligned} \frac{dx}{dt} &= \alpha x - \beta xy \\ \frac{dy}{dt} &= \delta xy - \gamma y, \end{aligned} \tag{4.7}$$

where x is the number of prey, y is the number of predators, α is the prey population growth rate, β is the rate of predation, δ is the predator population growth, and γ is the death rate of

	System	Equation
True Equation	dS/dt	$-0.150SI$
	dI/dt	$-0.900I + 0.150SI$
	dR/dt	$0.900I$
Posterior Mean	dS/dt	$\boldsymbol{-0.150SI}$
	dI/dt	$\boldsymbol{-0.900I + 0.150SI}$
	dR/dt	$\boldsymbol{0.900I}$
95% Credible Interval	dS/dt	$(-0.153, -0.147)SI$
	dI/dt	$(-0.900, -0.900)I + (0.147, 0.153)SI$
	dR/dt	$(0.900, 0.900)I$

Table 4.3: Posterior mean estimates and 95% credible intervals (lower bound, upper bound) for the SIR simulation with no measurement noise ($\xi = 0$).

	System	Equation
True Equation	dS/dt	$-0.150SI$
	dI/dt	$-0.900I + 0.150SI$
	dR/dt	$0.900I$
Posterior Mean	dS/dt	$\boldsymbol{-0.155SI}$
	dI/dt	$\boldsymbol{-0.998I + 0.150SI}$
	dR/dt	$\boldsymbol{0.908I}$
95% Credible Interval	dS/dt	$(-0.185, -0.129)SI$
	dI/dt	$(-1.125, -0.869)I + (0.135, 0.165)SI$
	dR/dt	$(0.790, 1.026)I$

Table 4.4: Posterior mean estimates and 95% credible intervals (lower bound, upper bound) for the SIR simulation with measurement noise ($\xi = 1$).

the predator population. We simulate from (4.7) with the model and simulation parameters given in Table 4.2. The simulated data are shown in Figure 4.3, where the blue line is the true system with no noise ($\xi = 0$) and the data with measurement noise $\xi = 1$ are depicted by the red dots. Because the system is only defined for positive x and y, if the noisy simulated data is less than zero, we set it equal to zero.

We fit the proposed model with parameters $p_\alpha = 400, |\mathscr{X}| = 50, \nu_0 = 10^{-6}, \nu_1 = 10^4$. With no measurement noise, we set $\kappa = 10^{-2}$ and with measurement noise we set $\kappa = 1$. Our library of potential solutions are all polynomials up to the third order with all possible

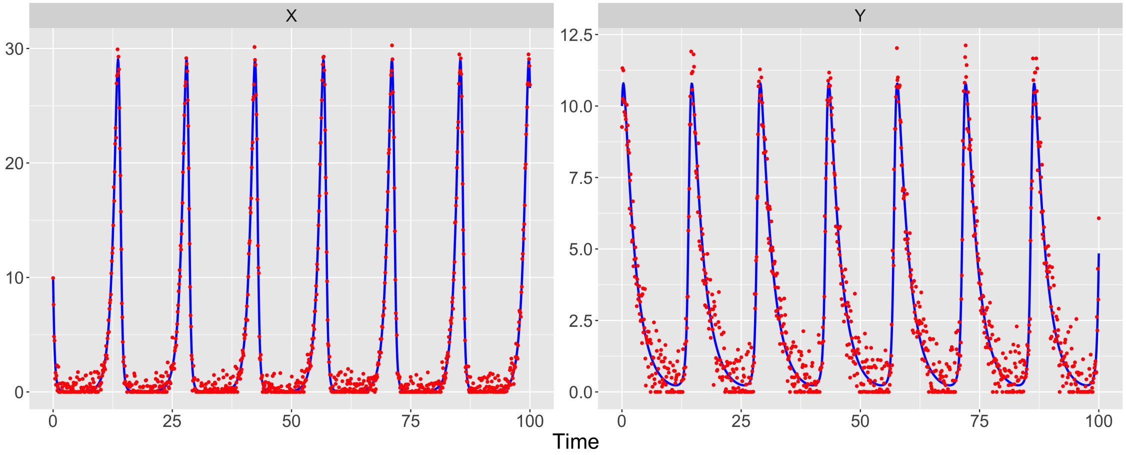


Figure 4.3: Data simulated from the Lotka-Volterra system, (4.7) with parameters $\alpha = 1.1, \beta = 0.4, \delta = 0.1, \gamma = 0.4$, and initial condition $[x, y]' = [10, 10]'$ for times $t = 0$ to $t = 100$ with a time step of $\Delta t = 0.1$. The solid blue line is the true system with no measurement noise ($\xi = 0$), and the red dots are the data with measurement noise ($\xi = 1$).

	System	Equation
True Equation	dx/dt	$1.100x - 0.400xy$
	dy/dt	$-0.400y + 0.100xy$
Posterior Mean	dx/dt	$\boldsymbol{1.099x - 0.409xy}$
	dy/dt	$\boldsymbol{-0.403y + 0.129xy}$
95% Credible Interval	dx/dt	$(1.061, 1.145)x + (-0.590, -0.275)xy$
	dy/dt	$(-0.420, -0.397)y + (0.084, 0.260)xy$

Table 4.5: Posterior mean estimates and 95% credible intervals (lower bound, upper bound) for the Lotka-Volterra simulation without measurement noise.

interactions. After obtaining posterior samples, we keep only terms with greater than a 99% inclusion probability. The recovered equations and 95% credible intervals without and with measurement noise for the selected terms are shown in Tables 4.5 and 4.6, respectively. We see the identified system is correct for both simulations and the credible intervals for all identified parameters are significant and cover the true parameter values.

	System	Equation
True Equation	dx/dt	$1.100x - 0.400xy$
	dy/dt	$-0.400y + 0.100xy$
Posterior Mean	dx/dt	$\boldsymbol{1.045x - 0.482xy}$
	dy/dt	$\boldsymbol{-0.395y + 0.135xy}$
95% Credible Interval	dx/dt	$(0.822, 1.238)x + (-0.673, -0.228)xy$
	dy/dt	$(-0.444, -0.357)y + (0.058, 0.246)xy$

Table 4.6: Posterior mean estimates and 95% credible intervals (lower bound, upper bound) for the Lotka-Volterra simulation with measurement noise ($\xi = 1$).

Coupled Pendulum

A coupled pendulum system consists of two individual pendulums coupled by a spring, resulting in the motion of each pendulum being dependent on the other. Let $\{\theta_i, i = 1, 2\}$ be the angle from vertical for each pendulum, m the mass of each body, L the length of the rod, k is the spring constant, and g the gravitational acceleration. Then, the coupled pendulum system is described by the set of second-order ODEs:

$$\begin{aligned} \frac{d^2\theta_1}{dt^2} &= -\frac{g}{L}\sin\theta_1 - \frac{k}{m}(\theta_1 - \theta_2) \\ \frac{d^2\theta_2}{dt^2} &= -\frac{g}{L}\sin\theta_2 + \frac{k}{m}(\theta_1 - \theta_2). \end{aligned} \tag{4.8}$$

We simulate from (4.8) with the model and simulation parameters given in Table 4.2. The simulated data are shown in Figure 4.4 where the angle from vertical for each pendulum are plotted against time.

We fit the proposed model with parameters $p_\alpha = 200, |\mathscr{X}| = 50, \kappa = 10^{-8}, \nu_0 = 10^{-6}, \nu_1 = 10^4$. Our library of potential solutions contained polynomials up to the second order, an intercept, and sin and cos terms. After obtaining posterior samples, we select only terms that have inclusion probability greater than 99%. Table 4.7 shows the posterior mean for the selected terms the 95% posterior credible intervals. We see the identified solution is

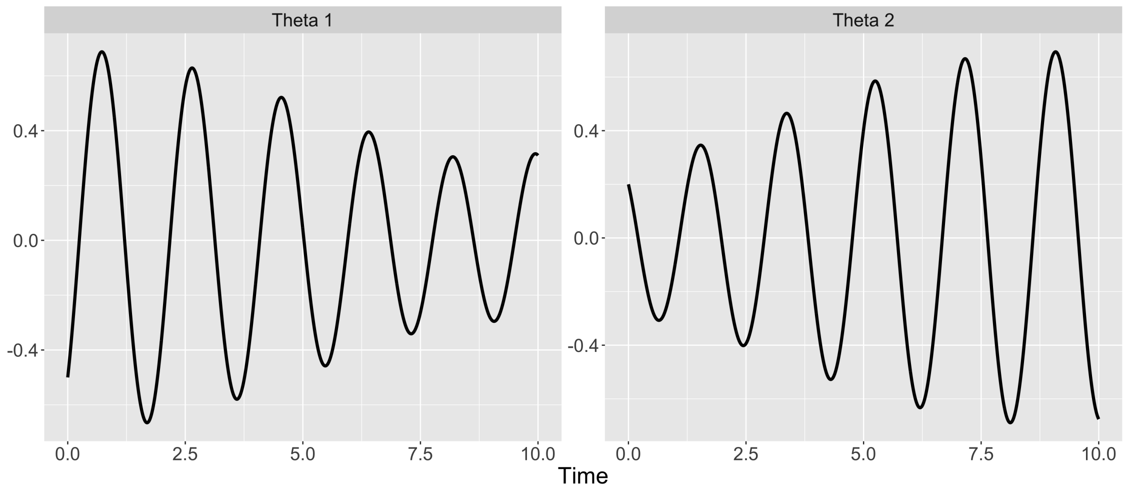


Figure 4.4: Data simulated from the coupled pendulum, (4.8) with parameters $g = 9.8, l = 1, k = 1, m = 1$, and initial condition $[\theta_1, \theta_2]' = [\pi/2, -\pi/4]'$ for times $t = 0$ to $t = 10$ with a time step of $\Delta t = 0.01$.

	System	Equation
True	$d^2\theta_1/dt^2$	$-1\theta_1 + 1\theta_2 - 9.8\sin(\theta_1)$
	$d^2\theta_2/dt^2$	$1\theta_1 - 1\theta_2 - 9.8\sin(\theta_2)$
Mean	$d^2\theta_1/dt^2$	$\mathbf{-1.005}\theta_1 + \mathbf{1.000}\theta_2 - \mathbf{9.795}\sin(\theta_1)$
	$d^2\theta_2/dt^2$	$\mathbf{1.000}\theta_1 - \mathbf{1.002}\theta_2 - \mathbf{9.797}\sin(\theta_2)$
95% CI	$d^2\theta_1/dt^2$	$(-1.102, -0.904)\theta_1 + (0.997, 1.003)\theta_2 + (-9.901, -9.693)\sin(\theta_1)$
	$d^2\theta_2/dt^2$	$(0.997, 1.004)\theta_1 + (-1.089, -0.918)\theta_2 + (-9.886, -9.706)\sin(\theta_2)$

Table 4.7: Posterior mean estimates and 95% credible intervals (lower bound, upper bound) for the coupled pendulum.

correct and the credible intervals for the identified parameters cover the truth.

Lorenz-63

For our last simulation study, we use a classic nonlinear dynamical system - the Lorenz-63 attractor (Lorenz, 1963). The Lorenz-63 attractor, originally proposed to represent a

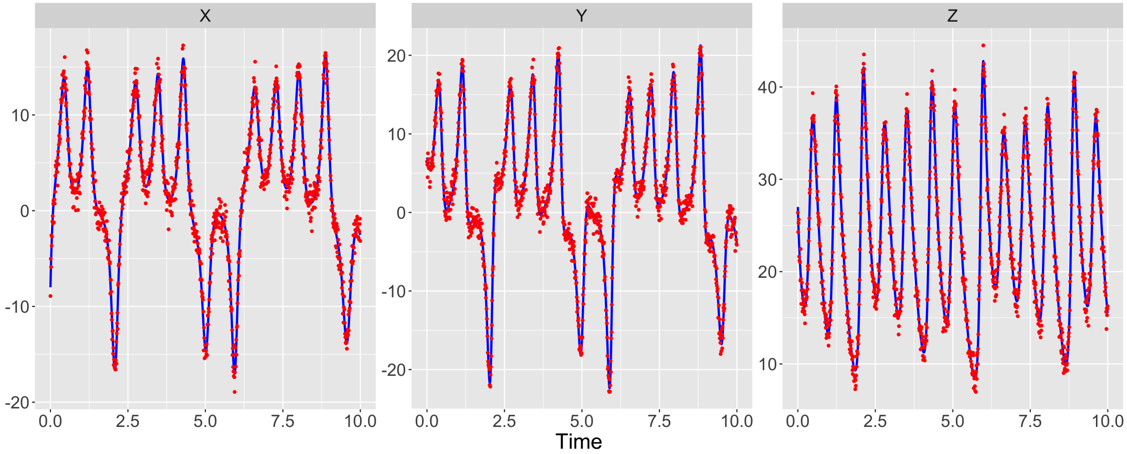


Figure 4.5: Simulated data from the Lorenz attractor under scenarios (1) and (2). The blue lines correspond to scenario (1) with no measurement noise and the red dots correspond to scenario (2) with measurement noise ($c = 1$).

simplified chaotic atmospheric system, consists of the three ODEs:

$$\begin{aligned} \frac{dx}{dt} &= \sigma(y - x) \\ \frac{dy}{dt} &= x(\rho - z) - y \\ \frac{dz}{dt} &= xy - \beta z, \end{aligned} \tag{4.9}$$

where x is proportional to the convection rate, y is proportional to the temperature difference in ascending and descending currents, and z is proportional to the vertical temperature distortion. We test our method on five scenarios: (1) no measurement noise ($\xi = 0$), (2) measurement noise ($\xi = 1$), (3) measurement noise ($\xi = 5$), (4) measurement noise ($\xi = 10$), and (5) measurement noise ($\xi = 1$) and 5% of the data missing at random. For all five scenarios, we simulate from (4.9) with the model and simulation parameters given in Table 4.2. The simulated data for scenario (1) and (2) are shown in Figure 4.5, with scenario (1) represented by the blue lines scenario (2) by the red dots.

	System	Equation
True	dx/dt	$-10x + 10y$
	dy/dt	$28x - 1y - 1xz$
	dz/dt	$-2.667z + 1xy$
Mean	dx/dt	$\boldsymbol{-9.947x + 9.980y}$
	dy/dt	$\boldsymbol{27.932x - 0.980y - 0.997xz}$
	dz/dt	$\boldsymbol{-2.666z + 0.999xy}$
95% CI	dx/dt	$(-10.169, -9.700)x + (9.826, 10.121)y$
	dy/dt	$(26.963, 28.855)x + (-1.269, -0.697)y + (-1.051, -0.939)xz$
	dz/dt	$(-2.700, -2.634)z + (0.977, 1.021)xy$

Table 4.8: Posterior mean estimates and 95% credible intervals (lower bound, upper bound) for the Lorenz-63 simulation with no measurement noise ($\xi = 0$).

For all scenarios, we fit the proposed model with model parameters $p_\alpha = 400, |\mathscr{X}| = 50, \nu_0 = 10^{-6}, \nu_1 = 10^4$. For scenario (1) $\kappa = 0.1$, scenario (2) $\kappa = 10$, scenario (3) $\kappa_u = 100, \kappa_l = 10$, scenario (4) $\kappa_u = 100, \kappa_l = 10$, and scenario (5) $\kappa = 10$. The larger learning rate for scenario (3) and (4) is due to the anticipated magnitude of measurement noise. Our library of potential solutions are all polynomials up to the third order with all possible interactions. After obtaining posterior samples, we select only terms that have inclusion probability greater than 99%. The recovered systems and 95% posterior credible intervals for scenario (1) are shown in Table 4.8, and the 95% posterior credible intervals for scenarios (2) - (5) are shown in Table 4.9. The recovered systems and 95% posterior credible intervals for scenarios (2) - (5) are shown in Tables C.1 - C.4 in Appendix C.2. When there is no noise, we correctly identify the dynamic system and all credible intervals cover the truth. When noise is present, we miss-identify the y component for all scenarios and fewer posterior credible intervals cover the correct parameter values as the noise increases (scenario (2) through scenario (4)). However, the method identifies the correct solution (except for one term) in all scenarios, including when data are missing at random.

Scenario	System	Equation
True Equation	dx/dt	$-10x+10y$
	dy/dt	$28x-1y-1xz$
	dz/dt	$-2.667z+1xy$
Scenario (2)	dx/dt	$\boldsymbol{(-10.325,-7.538)x+(8.393,10.305)y}$
	dy/dt	$(23.697,27.060)x+\boldsymbol{(-1.035,-0.791)xz}$
	dz/dt	$\boldsymbol{(-2.822,-2.518)z+(0.810,1.063)xy}$
Scenario (3)	dx/dt	$(-9.695,-6.533)x+(7.791,9.997)y$
	dy/dt	$(22.432,26.811)x+\boldsymbol{(-1.028,-0.728)xz}$
	dz/dt	$\boldsymbol{(-2.886,-2.516)z+(0.713,1.019)xy}$
Scenario (4)	dx/dt	$(-8.653,-5.609)x+(7.095,9.445)y$
	dy/dt	$(22.339,26.409)x+\boldsymbol{(-1.005,-0.723)xz}$
	dz/dt	$\boldsymbol{(-2.868,-2.472)z}+(0.594,0.942)xy$
Scenario (5)	dx/dt	$\boldsymbol{(-10.173,-6.942)x+(7.931,10.132)y}$
	dy/dt	$\boldsymbol{(24.037,28.186)x+(-1.052,-0.774)xz}$
	dz/dt	$(-2.658,-2.285)z+\boldsymbol{(0.722,1.093)xy}$

Table 4.9: 95% posterior credible intervals (lower bound, upper bound) for scenarios (2) - (5).

Missing System Component - Lorenz 63

As discussed briefly in Section 4.2.3, our model can accommodate certain scenarios where an entire system component is unobserved. To show this, we use the Lorenz-63 data simulated without measurement noise, shown by the blue line in the top panel of Figure 4.5. However, instead of using all components as data we only use y_t and z_t, omitting the x_t component. This results in $L(t)=2$ and $N=3$, so our latent space has a longer dimension than our observed space. To complicate the issue, the Lorenz system is unidentifiable, with $[-x_t,-y_t,z_t]$ being an alternate solution to $[x_t,y_t,z_t]$. We fit the proposed model with model parameters $p_\alpha=200, |\mathscr{X}|=50, \nu_0=10^{-6}, \nu_1=10^4, \kappa_u=10^2, \kappa_l=10^1$. Our library of potential solutions are all polynomials up to the second order with all possible interactions, and an intercept. Different from the previous Lorenz simulations, we use fewer basis functions to force the latent space to be more structured a smaller library of potential solutions

	System	Equation
True Equation	dx/dt	$-10x + 10y$
	dy/dt	$28x - 1y - 1xz$
	dz/dt	$-2.667z + 1xy$
Posterior Mean	dx/dt	$-6.938x + \mathbf{11.58y}$
	dy/dt	$19.623x - 0.713xz$
	dz/dt	$\mathbf{-2.648}z + 0.749xy$
95% Credible Interval	dx/dt	$(-9.128, -3.430)x + (9.715, 13.900)y$
	dy/dt	$(18.060, 21.752)x + (-0.713, -0.661)xz$
	dz/dt	$(-2.804, -2.512)z + (0.699, 0.837)xy$

Table 4.10: Posterior mean estimates and 95% credible intervals (lower bound, upper bound) for the Lorenz-63 simulation with the dx/dt component missing.

to restrict the solution space and make the solution identifiable. After obtaining posterior samples, and keeping only terms with greater than 99% inclusion probability, the recovered systems and 95% posterior credible intervals are shown in Table 4.10. We see the correct terms, except for the y in the dy/dt component, are identified in the solution. Thus, the model is able to infer the missing x_t component based solely on the relationship between the components y_t and z_t.

Comparison to SINDy

Here we compare our proposed method to the current state-of-the-art, SINDy. As with any model, SINDy relies on the specification of model parameters, which have the potential to impact results. We use the default settings except we specify the library of possible coefficients to be polynomials up to the third order with all possible interactions, i.e., the same library specified in Section 4.4.1. We acknowledge the possibility that a different choice of model parameters could lead to better results. We run the SINDy model with data generated from scenarios (1) - (4) of Section 4.4.1, the Lorenz-63 attractor with measurement noise $\xi = 0$, 1, 5, and 10. The SINDy model takes less than one second to run (compared

ξ	System	Equation
0	dx/dt	$-9.980x+9.980y$
	dy/dt	$27.808x-0.964y-0.995xz$
	dz/dt	$-2.659z+0.997xy$
1	dx/dt	$-1.169x+4.535y-0.230z+0.192x^2-0.331xy-0.233xz+0.141y^2+0.148yz$
	dy/dt	$18.643x+3.866y-0.747xz-0.119yz$
	dz/dt	$0.461x-0.287y-3.011z+0.453x^2+0.494y^2$
5	dx/dt	$4.925x+0.782y-0.384z+0.283x^2-0.462xy-0.367xz+0.178y^2+0.223yz$
	dy/dt	$12.425x+5.839y-0.155z+0.248x^2-0.402xy-0.581xz+0.164y^2-0.152yz$
	dz/dt	$-0.443x-2.067z-0.182x^2+1.023xy$
10	dx/dt	$5.222x+0.455y-0.329z+0.244x^2-0.397xy-0.345xz+0.133y^2+0.205yz$
	dy/dt	$9.959x+5.692y+0.192x^2-0.173xy-0.518xz-0.127yz$
	dz/dt	$-0.757x-1.676z-0.233x^2+0.973xy$

Table 4.11: Recovered equations for the Lorenz-63 simulation using the SINDy algorithm with varying amounts of noise.

to approximately 20 minutes for our proposed method), and the recovered equations are shown in Table 4.11. We see the method is impacted by noise, and as the data becomes more corrupted the relevant terms in the Lorenz system become obscured by extraneous terms. Additionally, the uncertainty in the parameter estimates is not provided by SINDy.

In this scenario, we feel our proposed method out performs SINDy, but this will not always be the case. Here, we only have 1000 time points for each system compared to hundreds of thousands of data points in the original SINDy paper. As of the writing of this manuscript, our model is impractical to use when the data is on the order of hundreds of thousands of observations as the MCMC estimation procedure will take too long. In contrast to SINDy, our method is best suited for cases where there is considerable measurement noise, the amount of data available is relatively small, and uncertainty quantification (parameter inference) is of interest.

4.4.2 Real-World Data

We demonstrate our model by discovering the dynamics of three real-world systems: the historic Hare-Lynx predator prey system, motion tracked pendulum, and Sea Surface Temperature.

Hare and Lynx Population Dynamics

We consider the historic Canadian lynx and snowshoe hare data set[1]. The data, black dots shown in Figure 4.6 for the Hare (top) and Lynx (bottom), document the population dynamics between the two species from 1845 to 1939 and were originally recorded by the Hudson's Bay Company (HBC, Elton and Nicholson, 1942). As discussed by Bulmer (1974), the data are reminiscent of a predator-prey relationship with a cycle of approximately 10 years. Based on the work by Bulmer (1974), we use a library of polynomials up to the third order with all possible interactions. We fit the proposed model for with model parameters $p_a = 40, |\mathscr{X}| = 20, \nu_0 = 10^{-6}, \nu_1 = 10^4, \kappa = 10$. After obtaining posterior samples, we keep only terms with greater than 99% inclusion probability. The identified system and 95% posterior credible intervals are shown in Table 4.12. We see the recovered solution has the same components as the Lotka-Volterra system, which is expected. However, there is some debate in the literature as to whether the Lotka-Volterra equations accurately represent the Hare-Lynx system (Zhang et al., 2007) or whether there should be another trophic level included in the system to accurately capture the dynamics (Krebs et al., 2001). Regardless, we do get results corresponding to the Lotka-Volterra system, suggesting predator-prey interactions are present.

[1] https://tuvalabs.com/datasets/lynx_and_snowshoe_hare_in_canada/activities

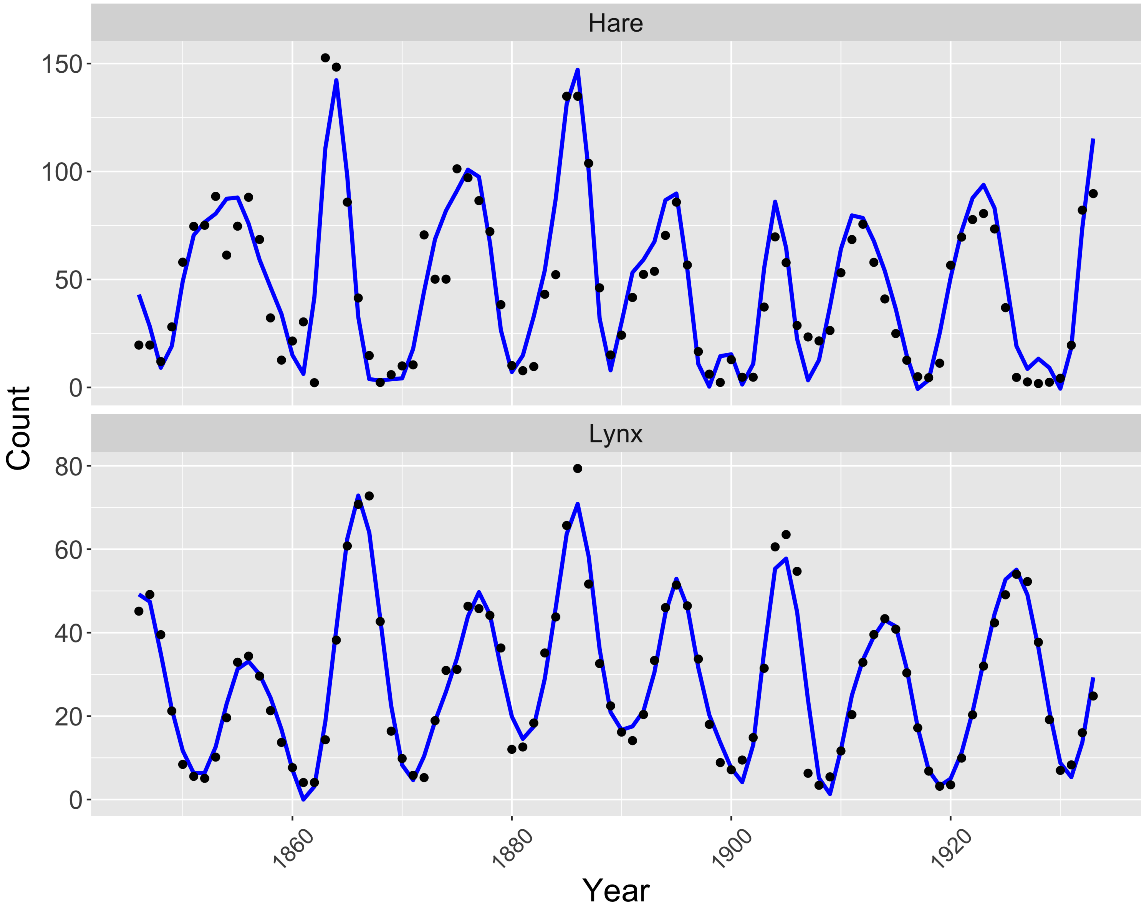


Figure 4.6: Hare-Lynx population relationship shown through the number of pelts the Hudson's Bay company recorded between 1845 and 1935. The black points signify the true data and the blue line is the recovered system for the Hare (top) and Lynx (bottom).

Motion Tracked Pendulum

In their seminal paper on data-driven discovery, Schmidt and Lipson (2009) provide motion tracked data for multiple physical systems. Here, we use their motion tracked data on the single pendulum, shown in Figure 4.7. The data consist of the angle of the pendulum from vertical at time t. It is key to note the data are not sampled at equal intervals. In this scenario, basic principles in physics suggest a solution to the system. Using this information

	System	Equation
Posterior Mean	dH/dt	$\mathbf{6.049}\boldsymbol{H}-\mathbf{0.272}\boldsymbol{HL}$
	dL/dt	$-\mathbf{6.232}\boldsymbol{L}+\mathbf{0.168}\boldsymbol{HL}$
95% Credible Interval	dH/dt	$(3.530, 8.426)H+(-0.400,-0.144)HL$
	dL/dt	$(-7.655,-4.796)L+(0.124, 0.210)HL$

Table 4.12: Posterior mean estimates and 95% credible intervals (lower bound, upper bound) for the Hare-Lynx data.

	System	Equation
Theoretical Equation	$d^2\theta/dt^2$	$-\frac{g}{l}\sin(\theta)-\frac{b}{m}\omega$
Posterior Mean	$d^2\theta/dt^2$	$-\mathbf{1.357}\sin(\theta)-\mathbf{0.012}\boldsymbol{\omega}$
95% Credible Interval	$d^2\theta/dt^2$	$(-1.360,-1.354)\sin(\theta)+(-0.014,-0.009)\omega$

Table 4.13: Posterior mean estimates and 95% credible intervals (lower bound, upper bound) for the motion tracked pendulum data.

to inform our choices, our library of potential solutions includes

$$\theta, \sin(\theta), \cos(\theta), \theta/\sin(\theta), \theta/\cos(\theta), \omega, \omega^2, \omega\sin(\theta), \omega\cos(\theta), \sin(\omega), \cos(\omega),$$

where θ is the angle from vertical and ω refers to the derivative of θ.

We fit our proposed method with $p_a = 250, |\mathscr{X}| = 20, \nu_0 = 10^{-6}, \nu_1 = 10^4, \kappa = 10^{-6}$. After obtaining posterior samples, we select only terms that were included with greater than 99% probability. The posterior mean and 95% credible intervals for the selected terms are shown in Table 4.13 along with the theoretical solution. While we do not know the actual parameter values for the true system, the terms in the identified system agree with the theoretical equation and all selected parameters are significant.

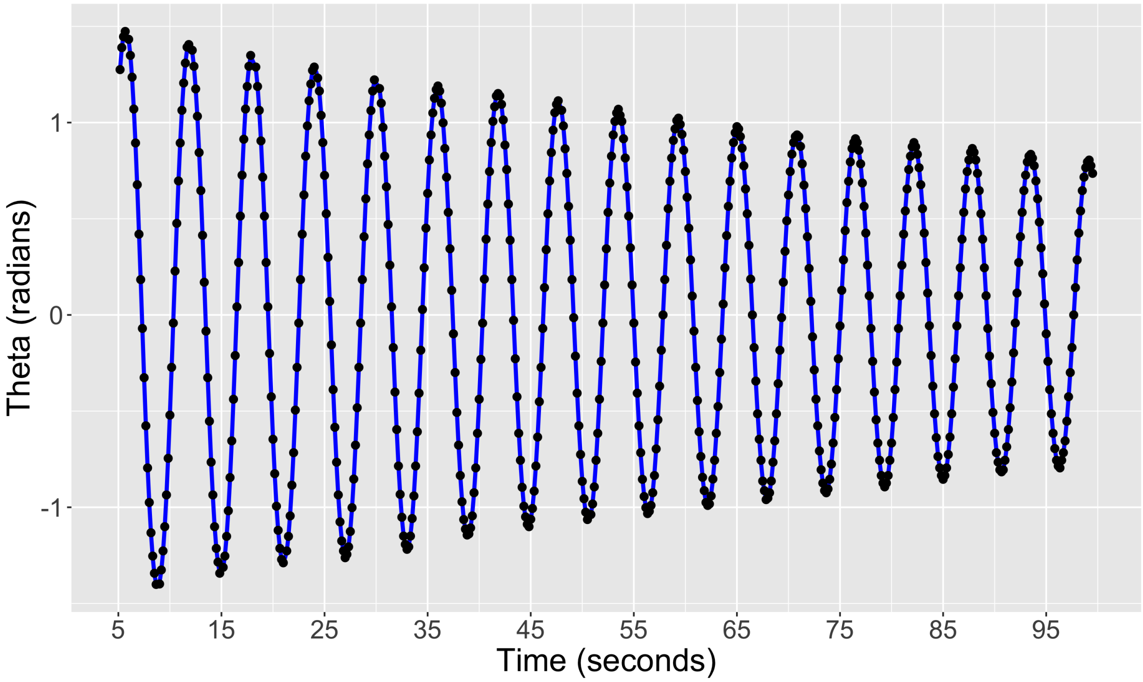


Figure 4.7: Pendulum's angle from vertical over time from the motion tracked data. The black points signify the true data and the blue line is the recovered system.

Sea Surface Temperature

The transitions from El Niño (anomalous warming) to La Niña (anomalous cooling) in the tropical Pacific ocean is known as the El Niño–Southern Oscillation (ENSO) cycle, and occurs quasi-periodically every 3-5 years (Philander, 1990). ENSO influences atmospheric and ecological systems globally and governmental agencies and industries rely on accurate forecasts of the event to make management decisions. Using publicly available sea surface temperature data from the IRI/LDEO Climate Data Library and originally produced by the National Ocean and Atmospheric Administration (NOAA) (Huang et al., 2017)[2], we recover implicit dynamics of the ENSO system. The data consist of monthly sea surface temperature (SST) anomalies from January 1926 to November 2021 and include multiple

[2] http://iridl.ldeo.columbia.edu/SOURCES/.NOAA/.NCDC/.ERSST/.version5/.anom/

ENSO cycles. We focus on two of the more recent events, the 1997-98 and 2015-16 ENSO cycles. We subset the data to include all time points leading up to each of these ENSO cycles – that is, January 1926 to March 1997 and January 1926 to February 2015, which we label ENSO-97 and ENSO-15, respectively. ENSO-97 and ENSO-15 are each decomposed using empirical orthogonal functions (EOFs) (Cressie and Wikle, 2011), where the first ten temporal principal component time series associated with the EOFs are treated as data and used to learn the dynamics.

In this application, we know that we are not considering all possible mechanisms driving SST (e.g., those associated with atmospheric winds, subsurface temperatures). Motivated by the success of statistical models in long-lead forecasting of ENSO (Barnston et al., 1999; Jan van Oldenborgh et al., 2005), our focus is on estimating the system and using it to forecast SST forward in time. As is customary in such applications, we use the average SST in the Niño 3.4 region (5S - 5N, 120W - 70W) to summarize the intensity of an El Niño event. Using the ENSO-97 and ENSO-15 data leading up to the 1997 and 2015 El Niño events, respectively, we learn the dynamics and generate a 12 month forecast of the SST for each event. For both ENSO-97 and ENSO-15, our library of potential functions are all polynomials up to the second order with all possible interactions (see Table C.5 for more detail). We implement our approach with model parameters $p_a = 200, |\mathscr{X}| = 50, \nu_0 = 10^{-6}, \nu_1 = 10^4, \kappa = 1$. We then compute the mean and highest posterior density (HPD) interval of the Niño 3.4 Index for each forecast and compare to the truth (Figure 4.8). For both ENSO events, we capture the parabolic increase and decrease in the Niño 3.4 Index with the point-wise HPD intervals covering the true Niño 3.4 Index for all but one forecast.

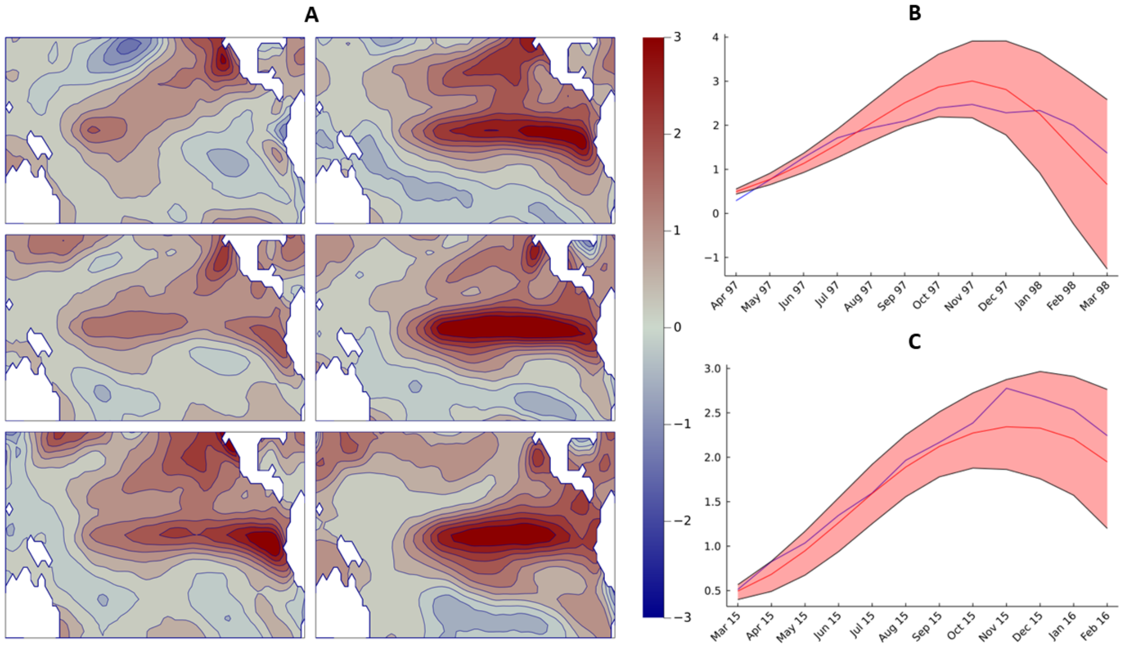


Figure 4.8: (A) Monthly SST anomalies as the ENSO shifts into El Niño (warming phase) shown in two month increments. From top to bottom, the left column is the SST corresponding to March, May, and July 2015, and the right column is the SST corresponding to September, November, and January 2015-16. (B) Predictions of the Niño 3.4 Index for the 1997-98 ENSO event showing the true (blue), posterior predicted mean (red), and 95% highest posterior credible intervals (red bands). (C) Predictions of the Niño 3.4 Index for the 2015-16 ENSO event showing the true (blue), posterior predicted mean (red), and 95% highest posterior credible intervals (red bands). *Note* - (B) and (C) are on different scales.

4.5 Conclusion

We have proposed a Bayesian hierarchical method to learn complex nonlinear dynamic equations using a data-driven approach. Our proposed method is robust to measurement noise and missing data, and can accommodate situations where a component is completely unobserved. The statistical approach to dynamic equation discovery is our most significant contribution, where we provide uncertainty quantification and inclusion probabilities to the terms in the library. This is possible because of the Bayesian hierarchical model that is composed of three components: a data model accounting for the uncertainty in the observed data, a process model learning the nonlinear dynamics in a latent space, and prior models. Additionally, we are able to bypass the need for numerical differentiation by expanding our latent process in terms of basis functions. As a whole, our proposed hierarchical model overcomes the limitations of the multi-step procedure and provides a complete statistical framework to the dynamic equation discovery problem.

We see two clear extensions to our research. The most beneficial extensions relate to the specification of the feature library. A library-free approach, which removes the potential bias associated with the specification of the library, would result in a truly data-driven approach. Additionally, allowing for time-varying parameters will increase the number of real-world applications for which the method can be applied. Most apparent are extensions to the SIR class of models where government intervention, variant strains, and other factors could be accounted for in the model. Another extension would be to impose restricts on different components of the system through the library. For example, when the environment may impact the population but not vise versa. Allowing this unidirectional forcing is beneficial from a physical viewpoint because it restricts the method from considering potential solutions that are not possible. Last, the work can be extended include to partial differential equations (Chapter 5). This would allow for the discovery of nonlinear dynamic spatial processes with uncertainty quantification.

Chapter 5

A Bayesian Approach for Data-Driven Dynamic Equation Discovery for Partial Differential Equations

5.1 Introduction

Dynamic equations (DE) parameterized by partial differential equations (PDE) – an equation relating a partial derivative of a variable to a function of its current state – are used across all fields of science and engineering to describe complex processes. DEs encode physical processes by a set of mathematical equations, enabling complex systems such as the spread of infectious disease (Bolker and Grenfell, 1995; Mangal et al., 2008; Kühnert et al., 2014), evolution of invasive species (Hastings, 1996; Liu et al., 2019), weather and climate (Charney et al., 1950; Holton and Hakim, 2012), and the flow of fluids (White and Majdalani, 2006) to be characterized and modeled (see also Higham et al., 2016, for further discussion). Given that any real world process is only approximately characterized by mathematical relationships, mathematically derived DEs are inherently unable to completely characterize a real world system. This suggests the need to use observations of the real world system to better characterize the underlying DEs.

Recently, there has been a push to use data to discover the governing equations in these complex systems. Originally proposed using symbolic regression (Bongard and Lipson, 2007; Schmidt and Lipson, 2009), the focus has since shifted to either sparse regression or deep modeling. The original sparse regression approach, termed *Sparse Identification of Nonlinear Dynamics* (SINDy; Brunton et al., 2016), used numerical differentiation to construct a response that is regressed against a library of functions that potentially govern the system. Through sparse regression, either with an ℓ_1 penalization term (Tibshirani, 1996) or using a thresholding approach (Zheng et al., 2019; Champion et al., 2020), key terms governing a variety of ordinary differential equations (ODE) are identified. Using the fundamental idea of SINDy, the framework was extended to include PDEs and parametric forms (Schaeffer, 2017; Rudy et al., 2017, 2019a,b), stochastic dynamical systems (Boninsegna et al., 2018), uncertainty quantification of the parameters (Zhang and Lin, 2018; Yang et al., 2019; Niven et al., 2020; Fasel et al., 2021; Hirsh et al., 2021), and has been incorporated into a Python package (PySINDy; de Silva et al., 2020).

Deep models used for data-driven discovery of dynamics can broadly be grouped into two categories – approximating dynamics (Raissi et al., 2017a; Raissi and Karniadakis, 2018; Raissi et al., 2020; Sun et al., 2019; Wu and Xiu, 2020) and discovering dynamics (Both et al., 2021; Xu et al., 2019, 2020, 2021; Long et al., 2017, 2019). Using deep models to approximate the dynamics of complex systems enables a computationally cheap method to obtain measurements of otherwise difficult to simulate systems while still obeying physical principles (see Reichstein et al., 2019, for an in-depth discussion on the topic).

However, our goal is the discovery of the governing equations and refer to “data-driven discovery” as the discovery of the *functional form* of the system in contrast to generating realistic dynamics. Deep models have also been used to discover the governing equations of complex systems. Combining deep modeling and sparse identification, Both et al. (2021) approximate the PDE using a neural network, which is used to compute derivatives and construct a sparse formulation similar to the SINDy approach. Long et al. (2017, 2019) use a symbolic neural network, an extension of symbolic regression, and a numerical approximation of differential operators in a feed-forward network to discover PDEs and Xu et al. (2021) use a fully connected neural network with a genetic algorithm to express and generate terms of a PDE.

Two open problems in data-driven discovery are (i) accounting for measurement uncertainty (i.e., missing data and measurement noise) and (ii) parameter uncertainty. Existing methods that extend the SINDy framework to account for uncertainty quantification employ either a bootstrap approach (Fasel et al., 2021) or a Bayesian approach with variable shrinkage/selection priors placed on the coefficients associated with the library terms (Zhang and Lin, 2018; Niven et al., 2020; Hirsh et al., 2021). These approaches directly follow the first step of the SINDy framework, where derivatives are computed numerically, data is de-noised, and the feature library is constructed. In this manner, the true uncertainty associated with the observed data is ignored; the estimate for the system uncertainty is now dependent on the numerical differentiation method, which subsequently influences the estimate for the parameter uncertainty. Yang et al. (2020) developed a method to jointly account for uncertainty in the observed data and parameters based on differential Bayesian programming, and while this approach now directly accounts for measurement uncertainty, it requires derivatives be computed using a numerical solver (e.g., Runge-Kutta). This can lead to numerical instabilities, and cannot account for missing data.

To account for observational uncertainty and missing data when modeling complex non-linear systems, statisticians have incorporated dynamic equations parameterized by PDEs into Bayesian hierarchical models (BHM; Berliner, 1996; Royle et al., 1999; Wikle et al., 2001). These models, sometimes called physical statistical models (PSM), enable modeling mechanistic relationships within a probabilistic framework (see Berliner, 2003; Cressie and Wikle, 2011; Kuhnert, 2017, for an overview). PSMs are composed of three sub-models – data, process, and parameter models. To account for observational and mechanistic uncertainty, PSMs consider the dynamics to be latent in the process stage, and represent the observed data in the data stage conditioned on these latent dynamics. While PSMs have been used to model and better understand complex systems, such as ocean surface winds (Wikle et al., 2001; Milliff et al., 2011) and the spread of avian species (Wikle, 2003; Hooten and Wikle, 2008), they require the dynamic relationships (although not weights/parameters associated with those relationships) to be specified *a priori*. To increase flexibility for representing complex processes, PSMs consider the parameters that describe the influence of dynamic components to be random, and often allow them to have spatial or temporal dependence, enabling the model to adapt to the data. While PSMs are adaptable to a variety of problems and provide inference on how the process may be evolving, they cannot be used to discover new dynamical relationships.

Chapter 4 of this dissertation proposed a Bayesian data-driven discovery method that accounts for observational and parameter uncertainty using a BHM framework composed of data, process, and parameter models for ODEs. Analogous to PSMs, the dynamics are modeled as a latent process and observational error is accounted for in the data model. Allowing the dynamics to be a latent random process is different than previous data-driven discovery methods that attempt to quantify uncertainty. To link the dynamic system to its derivatives probabilistically, the dynamic process and all the derivatives are modeled using a basis expansion with a common set of basis functions. Derivatives are then obtained analytically using the basis expansion, which incorporates dependence between the dynamic process and its derivatives. A library of potential functions can be constructed based on the basis coefficients and functions, and a variable selection prior is used to identify the key functions governing the nonlinear system.

Here, we propose a spatio-temporal extension to Bayesian data-driven discovery for PDEs. While our general framework is the same as North et al. (2022), the addition of the spatial dimension requires a reformulation of the process model. To account for the extra dimension (i.e., space), we model the dynamic process as a higher-order tensor where the dimensions represent space, time, and the number of components (sometimes called the system states) in the system. The tensor is decomposed using differentiable basis functions in space and time, probabilistically linking the dynamic system with its spatial and temporal derivatives. The basis decomposition is incorporated into the BHM, enabling potential functions to be constructed using the basis functions and coefficients. A variable selection prior on the coefficients produces a sparse solution set and the resulting system. In contrast to the ODE discovery problem, the library of potential functions for PDEs can exhibit strong multicollinearity and accounting for this multicollinearity is another fundamental extension beyond the previous approach.

We demonstrate our method on data generated from Burgers' equation, the heat equation, and a predator-prey reaction-diffusion equation with varying levels of measurement noise. In addition, we demonstrate our model's ability to accommodate missing data using Burgers' equation. The simulations show that our approach is robust to measurement noise and missing data, able to learn the dynamics of complex systems, and provides formal uncertainty quantification on parameter estimates and the confidence of the discovered dynamics. Last, we apply our method to to infer the evolution of atmospheric vorticity over time having only observed the streamfunction and obtain results that coincide with geophysical laws (i.e., the barotropic vorticity equation).

The remainder of this chapter is organized as follows. In Section 5.2 we define the tensor and derivative notation used throughout the manuscript. In Section 5.3 we give background on the general dynamic system, showcase how inference on the derivative of the system is made, and present the Bayesian hierarchical model. In Section 5.4 we describe parameter estimation and discuss modeling choices. In Section 5.5 we demonstrate our method on multiple simulated data sets and in Section 5.6 we perform inference on a real-world system. Section 5.7 concludes the chapter.

5.2 Preliminary Notation

In this section we define tensor and derivative notation. All variables in this section are used only for illustrating notation. Problem specific notation will be introduced in Section 5.3.

5.2.1 Tensor Notation

PDEs are commonly defined over multiple dimensions (e.g., space, time, components), and benefit from the use of higher-order tensor notation when the number of dimensions is three or more. We generally follow the notation of Kolda and Bader (2009) and refer the reader to their work for more details and references of tensor notation and applications.

Let $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_N}$ be a tensor of order N where the $(i_1, i_2, ..., i_N)$ element is denoted by $\mathcal{X}(i_1, i_2, ..., i_N)$. A *slice* of the tensor is a two-dimensional section where all but two indices are held constant. For example, the horizontal, lateral, and frontal slices of the third order tensor $\mathcal{Y} \in \mathbb{R}^{I \times J \times K}$ are denoted by $\mathbf{Y}_{i::}$, $\mathbf{Y}_{:j:}$, and $\mathbf{Y}_{::k}$, respectively. A tensor can be converted to a matrix using n-mode matricization (also known as unfolding or flattening). The n-mode matricization of the tensor $\mathcal{X}$, denoted by $\mathbf{X}_{(n)}$, arranges the mode$-n$ fibers (the higher-order equivalent of matrix rows and columns) to be columns in the resulting matrix. For example, the possible modes of $\mathcal{Y}$ are $\mathbf{Y}_{(1)} \in \mathbb{R}^{I \times (J \times K)}$, $\mathbf{Y}_{(2)} \in \mathbb{R}^{J \times (I \times K)}$, and $\mathbf{Y}_{(3)} \in \mathbb{R}^{K \times (I \times J)}$. In general, we will only be concerned with the mode-3 matricization of a tensor and will denote $\mathbf{Y}$ in place of $\mathbf{Y}_{(3)}$ (all other modes will be properly denoted).

To multiply a tensor by a matrix $\mathbf{B} \in \mathbb{R}^{I_n \times J}$, we use the n-mode product (i.e., multiply a tensor by a matrix or vector in mode n). The n-mode product of the tensor $\mathcal{X}$ and matrix $\mathbf{B}$ is denoted as $\mathcal{X} \times_n \mathbf{B}$ and is of size $I_1 \times I_2 ... \times I_{n-1} \times J \times I_{n+1} \times ... \times I_N$. Equivalently, in terms of unfolded (matricized) tensors, $\mathcal{Z} = \mathcal{X} \times_n \mathbf{B} \Leftrightarrow \mathbf{Z}_{(n)} = \mathbf{B}\mathbf{X}_{(n)}$.

5.2.2 Tensor Basis Representation

Let $\mathcal{Y} \in \mathbb{R}^{I \times J \times K}$ be the order 3 tensor from Section 5.2.1. Define the expansion of the (i, j, k) element of $\mathcal{Y}$ as

$$y(i, j, k) \equiv \sum_{p=1}^{\infty} \sum_{q=1}^{\infty} \sum_{r=1}^{\infty} g(p, q, r) a(i, p) b(j, q) c(k, r),$$

where $\{a(i, p) : p = 1, 2, ...\}$, $\{b(j, q) : q = 1, 2, ...\}$, and $\{c(k, r) : r = 1, 2, ...\}$ are basis functions and $\{g(p, q, r) : p, q, r = 1, 2, ...\}$ is the tensor of associated basis coefficients. To reduce the dimension, we keep the first P, Q, and R terms from a, b, and c, and define each basis function at discrete values $p = 1, ..., P$, $q = 1, ..., Q$, and $r = 1, ..., R$, respectively. That is, let

$$\mathcal{Y} \approx \sum_{p=1}^{P}\sum_{q=1}^{Q}\sum_{r=1}^{R} g(p,q,r)\mathbf{a}(p)\circ\mathbf{b}(q)\circ\mathbf{c}(r) = [\![\mathcal{G};\mathbf{A},\mathbf{B},\mathbf{C}]\!] = \mathcal{G}\times_1\mathbf{A}\times_2\mathbf{B}\times_3\mathbf{C},$$

where $\circ$ is the vector outer product, $\mathbf{A}$ is a $I\times P$, $\mathbf{B}$ is a $J\times Q$, and $\mathbf{C}$ is a $K\times R$ matrix of basis coefficients where each column is given by $\mathbf{a}(p)\equiv(a(1,p),...,a(I,p))$, $\mathbf{b}(q)\equiv(b(1,q),...,a(J,q))$, and $\mathbf{c}(r)\equiv(a(1,r),...,a(K,r))$, and $[\![\mathcal{G};\mathbf{A},\mathbf{B},\mathbf{C}]\!]$ is shorthand notation introduced in Kolda (2006). Our basis decomposition is similar to the Tucker decomposition (Tucker, 1966), except we assume $\mathbf{A},\mathbf{B}$, and $\mathbf{C}$ are known and our goal is to estimate $\mathcal{G}$. Note, we provide the expansion only for an order 3 tensor (sufficient for this manuscript), but the concept can be extended to higher order tensors.

5.2.3 Derivative Notation

As discussed in Section 5.1, we propose a method to discover the governing equations in PDEs. As the name suggests, a PDE is composed of partial derivatives of some variable $u=u(x,y,t)$ that is indexed in space or time or both. We denote partial derivatives using a subscript, for example $\frac{\partial u}{\partial t}=u_t$, $\frac{\partial u}{\partial x}=u_x$, $\frac{\partial^2 u}{\partial t^2}=u_{tt}$, and so forth. We denote the ith order of a derivative generally as $\frac{\partial^{(i)}t}{\partial t^{(i)}}=u_{t^{(i)}}$. In order to disambiguate notation, we denote the index of a vector/matrix/tensor using parentheses (e.g., $\mathbf{a}(i),\mathbf{A}(i,j),\mathcal{A}(i,j,k)$), reserving the subscript to denote derivatives.

Within the PDE literature there are different choices of notation to denote the same operation. For example, the Laplacian operator can be denoted as $\Delta u=\nabla^2 u=\nabla\cdot\nabla u=\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=u_{xx}+u_{yy}$. Wherever an operator such as the Laplacian is used for the first time, we will define it. This may result in our notation differing from other texts, but we aim to be consistent within the paper.

5.3 Bayesian Dynamic Equation Discovery

Here we propose a general hierarchical model for making inference on nonlinear spatio-temporal dynamic systems. We begin by motivating the general class of PDEs and manipulate them to fit within a statistical framework.

5.3.1 Dynamic Equations

Consider the general PDE dynamic system describing the evolution of a continuous field $\{\mathbf{u}(\mathbf{s},t) : \mathbf{s} \in D_s, t \in D_t\}$,

$$\mathbf{u}_{t^{(J)}}(\mathbf{s},t) = M\left(\mathbf{u}(\mathbf{s},t), \mathbf{u}_x(\mathbf{s},t), \mathbf{u}_y(\mathbf{s},t), \mathbf{u}_{xy}(\mathbf{s},t), ..., \mathbf{u}_{t^{(1)}}(\mathbf{s},t), ..., \mathbf{u}_{t^{(J-1)}}(\mathbf{s},t), \boldsymbol{\omega}(\mathbf{s},t)\right) \quad (5.1)$$

where the vector $\mathbf{u}(\mathbf{s},t) \in \mathbb{R}^N$ denotes the realization of the N-dimensional system at location $\mathbf{s}$ and time t (e.g., $\mathbf{u}(\mathbf{s},t) = [u(\mathbf{s},t,1), u(\mathbf{s},t,2), ..., u(\mathbf{s},t,N)]'$), $M(\cdot)$ represents the (potentially nonlinear) evolution function, and $\boldsymbol{\omega}(\mathbf{s},t)$ represents any covariates that might be included in the system. Here, $\mathbf{s} \in \{\mathbf{s}_1, ..., \mathbf{s}_S\} = D_s$ is a spatial location in the domain with $|D_s| = S$, and $t \in \{1, ..., T\} = D_t$ is the temporal realization of the system where $|D_t| = T$. Whereas we define (5.1) in two dimensions with $D_s \in \mathbb{R}^2$ and $\mathbf{s} = (x,y)$, the problem can be simplified to one dimension (i.e., $D_s \in \mathbb{R}^1$ and $\mathbf{s} = x$) or generalized to higher spatial dimensions. Finally, as is common in the dynamic systems literature, we refer to the N-dimensional multivariate vector $\mathbf{u}(\mathbf{s},t)$ as the state or system state, and use the term *component* to refer to each of the N elements of $\mathbf{u}(\mathbf{s},t)$.

We reparameterize 5.1 to be intrinsically linear (in parameters) as

$$\mathbf{u}_{t^{(J)}}(\mathbf{s},t) = \mathbf{M}\mathbf{f}\left(\mathbf{u}(\mathbf{s},t), \mathbf{u}_x(\mathbf{s},t), \mathbf{u}_y(\mathbf{s},t), \mathbf{u}_{xy}(\mathbf{s},t), ..., \mathbf{u}_{t^{(1)}}(\mathbf{s},t), ..., \mathbf{u}_{t^{(J-1)}}(\mathbf{s},t), \boldsymbol{\omega}(\mathbf{s},t)\right), \tag{5.2}$$

where $\mathbf{M}$ is a $N \times D$ *sparse* matrix of coefficients and $\mathbf{f}(\cdot)$ is a vector-valued nonlinear transformation function of length D. The input of the arguments for $\mathbf{f}(\cdot)$ are general and contain anything that *potentially* relates to the system. For example, this could include terms describing advection, diffusion, dispersion and growth, polynomial functions and interactions, or sinusoidal functions, and are chosen based on a general mechanistic understanding of the system. This results in D being quite large and (5.2) has the potential to be highly over-parameterized. Thus, we will employ regularization to induce sparsity in the matrix $\mathbf{M}$.

As an example of a classic PDE within our framework, consider the reaction diffusion equation

$$\mathbf{u}_t(\mathbf{s},t) = \mathbf{D}\nabla^2\mathbf{u}(\mathbf{s},t) + \mathbf{g}(\mathbf{u}(\mathbf{s},t)),$$

where $\mathbf{u}(\mathbf{s},t) = [b(\mathbf{s},t), d(\mathbf{s},t)]'$ represents the densities of two processes, $\mathbf{D}$ is a diagonal matrix where $diag(\mathbf{D}) = [D_b, D_d]$ are the diffusion constants, and $\mathbf{g}(\mathbf{u}(\mathbf{s},t)) = [g_b(b(\mathbf{s},t), d(\mathbf{s},t)), g_d(b(\mathbf{s},t), d(\mathbf{s},t))]'$ are (non)linear reaction functions. The reaction-diffusion equation can be used to model the densities of prey (b) and predator (d) populations (Hastings, 1996). For a predator-prey model, a possible choice for the reaction function is $\mathbf{g}(\mathbf{u}(\mathbf{s},t)) = [\gamma b - \delta bd, -\mu d + \eta bd]'$, a simplistic representation of the Lotka-Volterra system where γ and δ represent the prey's birth and predation rates, respectively, and μ and η represent the predator death and kill success rates. Following (5.2), and suppressing the spatial and temporal indices, we have

$$\begin{bmatrix} b_t \\ d_t \end{bmatrix} = \begin{bmatrix} \gamma & 0 & -\delta & D_b & D_b & 0 & 0 \\ 0 & -\mu & \eta & 0 & 0 & D_d & D_d \end{bmatrix} \begin{bmatrix} b & d & bd & b_{xx} & b_{yy} & d_{xx} & d_{yy} \end{bmatrix}'.$$

Typically we do not know $\mathbf{f}(\cdot) = [b, d, bd, b_{xx}, b_{yy}, d_{xx}, d_{yy}]'$ and instead highly over-parameterize $\mathbf{f}(\cdot)$ by including a library of potential terms and select against the coefficients in $\mathbf{M}$ to identify relevant terms.

In real-world problems, (5.2) does not hold exactly. Stochastic forcing could perturb the system (e.g., weather systems, demographic stochasticity) or there could be error in the model specification. We accommodate this unknown stochasticity by including an additive error term

$$\mathbf{u}_{t^{(J)}}(\mathbf{s},t) = \mathbf{M}\mathbf{f}\left(\mathbf{u}(\mathbf{s},t), \mathbf{u}_x(\mathbf{s},t), \mathbf{u}_y(\mathbf{s},t), \mathbf{u}_{xy}(\mathbf{s},t), ..., \mathbf{u}_{t^{(1)}}(\mathbf{s},t), ..., \mathbf{u}_{t^{(J-1)}}(\mathbf{s},t), \boldsymbol{\omega}(\mathbf{s},t)\right) + \boldsymbol{\eta}(\mathbf{s},t), \tag{5.3}$$

where, for example, $\boldsymbol{\eta}(\mathbf{s},t) \overset{\text{i.i.d.}}{\sim} N(\mathbf{0}, \boldsymbol{\Sigma}_U)$ is a mean zero Gaussian process with variance/-covariance matrix $\boldsymbol{\Sigma}_U$. In general, spatial or temporal dependencies could be considered in this error term.

To represent (5.3) using tensor notation, let $\mathcal{U} = \{u(\mathbf{s},t,n) : \mathbf{s} \in D_s, t = 1, ..., T, n = 1, ..., N\}$ where $\mathcal{U} \in \mathbb{R}^{S \times T \times N}$ is the tensor of the dynamic process. Similarly, let $\mathcal{F} \in \mathbb{R}^{S \times T \times D}$ be the function $\mathbf{f}(\cdot)$ evaluated at each location in space-time and $\widetilde{\boldsymbol{\eta}} \in \mathbb{R}^{S \times T \times N}$

is the space-time-component uncertainty tensor. The tensor formulation of (5.3) in then

$$\mathcal{U}_{t^{(J)}} = \mathcal{F} \times_3 \mathbf{M} + \widetilde{\boldsymbol{\eta}}. \tag{5.4}$$

This forms the core of our process model, where we relate the temporal derivative of some space-time-component process to a nonlinear function of its current state. While not explicitly stated in (5.4), $\mathcal{F}$ is still a function of the state process $\mathcal{U}$.

5.3.2 Basis Representation

As described in Section 5.2.2, we can represent the $\mathcal{U}$ tensor using basis functions. Decomposing $\mathcal{U}$ in terms of a finite collection of spatial, temporal, and component basis functions, we write

$$\mathcal{U} \approx \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{r=1}^{R} a(p,q,r) \psi(p) \circ \phi(q) \circ \boldsymbol{\theta}(r) = \mathcal{A} \times_1 \boldsymbol{\Psi} \times_2 \boldsymbol{\Phi} \times_3 \boldsymbol{\Theta} := [\![\mathcal{A}; \boldsymbol{\Psi}, \boldsymbol{\Phi}, \boldsymbol{\Theta}]\!],$$

where $\mathcal{A} \in \mathbb{R}^{P \times Q \times R}$, $\boldsymbol{\Psi} \in \mathbb{R}^{S \times P}$, $\boldsymbol{\Phi} \in \mathbb{R}^{T \times Q}$, and $\boldsymbol{\Theta} \in \mathbb{R}^{N \times R}$. Here, $\boldsymbol{\Psi}, \boldsymbol{\Phi}$, and $\boldsymbol{\Theta}$ are matrices of spatial, temporal, and component basis functions, respectively, and $\mathcal{A}$ is a tensor of basis coefficients (traditionally called the *core tensor*).

We can obtain derivatives of the elements of $\mathcal{U}$ analytically by taking derivatives of the basis functions. Specifically, let $\boldsymbol{\Psi}$ and $\boldsymbol{\Phi}$ be matrices of basis functions differentiable up to at least the highest order considered in (5.1). We then compute spatial and temporal derivatives of $\mathcal{U}$ by computing the derivatives of $\boldsymbol{\Psi}$ and $\boldsymbol{\Phi}$. That is, denote $\frac{\partial}{\partial x} \boldsymbol{\Psi} = \boldsymbol{\Psi}_x$,

$\frac{\partial}{\partial y}\boldsymbol{\Psi} = \boldsymbol{\Psi}_y$, $\frac{\partial}{\partial t}\boldsymbol{\Phi} = \boldsymbol{\Phi}_t$, and so forth. Derivatives of $\mathcal{U}$ are then computed as

$$\begin{aligned}
\frac{\partial}{\partial t}\mathcal{U} &= \mathcal{A} \times_1 \boldsymbol{\Psi} \times_2 \boldsymbol{\Phi}_t \times_3 \boldsymbol{\Theta} = [\![\mathcal{A}; \boldsymbol{\Psi}, \boldsymbol{\Phi}_t, \boldsymbol{\Theta}]\!] \\
\frac{\partial}{\partial x}\mathcal{U} &= \mathcal{A} \times_1 \boldsymbol{\Psi}_x \times_2 \boldsymbol{\Phi} \times_3 \boldsymbol{\Theta} = [\![\mathcal{A}; \boldsymbol{\Psi}_x, \boldsymbol{\Phi}, \boldsymbol{\Theta}]\!] \\
\frac{\partial^2}{\partial x \partial y}\mathcal{U} &= \mathcal{A} \times_1 \boldsymbol{\Psi}_{xy} \times_2 \boldsymbol{\Phi} \times_3 \boldsymbol{\Theta} = [\![\mathcal{A}; \boldsymbol{\Psi}_{xy}, \boldsymbol{\Phi}, \boldsymbol{\Theta}]\!],
\end{aligned} \tag{5.5}$$

and so forth. Representing (5.4) using the basis decomposition, we have

$$[\![\mathcal{A}; \boldsymbol{\Psi}, \boldsymbol{\Phi}_{t^{(J)}}, \boldsymbol{\Theta}]\!] = \mathcal{F} \times_3 \mathbf{M} + \widetilde{\boldsymbol{\eta}}, \tag{5.6}$$

where $\widetilde{\boldsymbol{\eta}}$ may include truncation error. While not explicitly stated, $\mathcal{F}$ now depends on $\boldsymbol{\Psi}, \boldsymbol{\Phi}, \boldsymbol{\Theta}$, and $\mathcal{A}$.

Proposition 1. *The mode-3 decomposition of* $[\![\mathcal{A}; \boldsymbol{\Psi}, \boldsymbol{\Phi}_{t^{(J)}}, \boldsymbol{\Theta}]\!] = \mathcal{F} \times_3 \mathbf{M} + \widetilde{\boldsymbol{\eta}}$ *where* $\boldsymbol{\eta}(\mathbf{s}, t) \overset{i.i.d.}{\sim} N_N(\mathbf{0}, \Sigma_U)$ *in space and time at location* $\mathbf{s}$ *and time* t *is*

$$\begin{aligned}
&\boldsymbol{\Theta}\mathbf{A}(\boldsymbol{\phi}_{t^{(J)}}(t) \otimes \boldsymbol{\psi}(\mathbf{s}))' = \\
&\mathbf{M}\mathbf{f}(\mathbf{A}, \boldsymbol{\psi}(\mathbf{s}), \boldsymbol{\psi}_x(\mathbf{s}), \boldsymbol{\psi}_y(\mathbf{s}), \boldsymbol{\psi}_{xy}(\mathbf{s}), \ldots, \boldsymbol{\phi}_{t^{(0)}}(t), \ldots, \boldsymbol{\phi}_{t^{(J)}}(t), \boldsymbol{\omega}(\mathbf{s}, t)) + \boldsymbol{\eta}(\mathbf{s}, t),
\end{aligned} \tag{5.7}$$

where $\mathbf{A}$ *is a* $R \times PQ$ *matrix of basis coefficients,* $\boldsymbol{\psi}(\mathbf{s})$ *is a length-*P *vector of spatial basis functions,* $\boldsymbol{\phi}(t)$ *is a length-*Q *vector of temporal basis functions, and* $\boldsymbol{\Theta}$ *is a* $N \times R$ *matrix of component basis functions.*

Proof. See Appendix D.3.

Decomposing (5.4) in terms of basis functions and taking the mode-3 matricization accomplishes two tasks. First, this enables inference on derivatives of the process $\mathbf{u}(\mathbf{s}, t)$ when only the process is known (e.g., see (5.5)). Second, keeping fewer basis functions than observations (e.g., $P < S$, $Q < T$) allows the reconstruction of $\mathcal{U}$ to be smooth (Wang et al., 2016).

Note, we include $\boldsymbol{\Theta}$ for generality in the construction of our method. While one could specify $\boldsymbol{\Theta}$ in terms of basis functions, our goal is not to reduce the dimension of the system state variables. In our analyses, we choose $\boldsymbol{\Theta}$ to be the identity matrix.

5.3.3 Transformation of Derivative

Up to this point, we have considered PDEs that relate the temporal derivative (of some order J) of the continuous surface $\mathbf{u}$ (left hand side (LHS) of (5.1)) to a function of its current state on (right hand side (RHS) of (5.1)). However, equations with a spatio-temporal derivative of $\mathbf{u}$ on the LHS are common (e.g., vorticity equation, Higham et al., 2016). For example, the LHS of (5.1) could depend on the Laplacian operator, where $\nabla^2 \mathbf{u}_{t^{(J)}}(\mathbf{s},t) = \mathbf{u}_{xxt^{(J)}}(\mathbf{s},t) + \mathbf{u}_{yyt^{(J)}}(\mathbf{s},t)$.

To be more general, we now allow the LHS of (5.1) to be a function of spatio-temporal derivatives of $\mathbf{u}$ and consider the more general PDE

$$g(\mathbf{u}_{t^{(J)}}(\mathbf{s},t)) = M\left(\mathbf{u}(\mathbf{s},t), \mathbf{u}_x(\mathbf{s},t), \mathbf{u}_y(\mathbf{s},t), \psi_{xy}(\mathbf{s}), ..., \mathbf{u}_{t^{(1)}}(\mathbf{s},t), ..., \mathbf{u}_{t^{(J-1)}}(\mathbf{s},t), \boldsymbol{\omega}(\mathbf{s},t)\right), \tag{5.8}$$

where $g(\cdot)$ is some linear differential operator. The original PDE (5.1) is a special case of (5.8) where $g(\cdot)$ is the identity function.

Proposition 2. *Let $g(\cdot)$ be a linear differential operator. The basis formulation of a PDE* *with a space-time function $g(\mathbf{u}_{t^{(J)}}(\mathbf{s},t))$ on the LHS is*

$$\mathbf{\Theta A}(\boldsymbol{\phi}_{t^{(J)}}(t) \otimes g(\boldsymbol{\psi}(\mathbf{s})))'.$$

Proof. See Appendix D.3.

From Proposition 2, the basis representation of a PDE with a spatio-temporal function on the LHS is

$$\begin{aligned} &\mathbf{\Theta A}(\boldsymbol{\phi}_{t^{(J)}}(t) \otimes g(\boldsymbol{\psi}(\mathbf{s})))' = \\ &\mathbf{Mf}(\mathbf{A}, \boldsymbol{\psi}(\mathbf{s}), \boldsymbol{\psi}_x(\mathbf{s}), \boldsymbol{\psi}_y(\mathbf{s}), \boldsymbol{\psi}_{xy}(\mathbf{s}), ..., \boldsymbol{\phi}_{t^{(0)}}(t), ..., \boldsymbol{\phi}_{t^{(J)}}(t), \boldsymbol{\omega}(\mathbf{s},t)) + \boldsymbol{\eta}(\mathbf{s},t), \end{aligned} \tag{5.9}$$

where $\boldsymbol{\eta}(\mathbf{s},t) \overset{i.i.d.}{\sim} N_N(\mathbf{0}, \Sigma_U)$ in space and time. Completing the example from before using the $g = \nabla^2$ Laplacian operator, the LHS for (5.9) is $\mathbf{\Theta A}(\boldsymbol{\phi}_{t^{(J)}}(t) \otimes (\boldsymbol{\psi}_{xx}(\mathbf{s}) + \boldsymbol{\psi}_{yy}(\mathbf{s})))'$.

5.3.4 Data Model

We assume $\mathbf{v}(\mathbf{s},t)$ is an observation of the N-dimensional latent process outlined in Section 5.3.2 with some unknown measurement uncertainty. We model $\mathbf{v}(\mathbf{s},t)$ using a generalization to the traditional linear data error model that links the dynamics to the observed process (e.g., see Cressie and Wikle, 2011, Chapter 7). That is, we model

$$\mathbf{v}(\mathbf{s},t) = \mathbf{H}(\mathbf{s},t)\mathbf{u}(\mathbf{s},t) + \widetilde{\boldsymbol{\epsilon}}(\mathbf{s},t), \tag{5.10}$$

where $\mathbf{v}(\mathbf{s},t) \in \mathbb{R}^{L(\mathbf{s},t)}$, $\mathbf{H}(\mathbf{s},t) \in \mathbb{R}^{L(\mathbf{s},t)\times N}$ is the incidence matrix that maps from $\mathbf{u}(\mathbf{s},t)$ to $\mathbf{v}(\mathbf{s},t)$, and uncertainty in the observations of the process are captured by $\widetilde{\boldsymbol{\epsilon}}(\mathbf{s},t) \overset{\text{indep.}}{\sim} N_{L(\mathbf{s},t)}(\mathbf{0}, \widetilde{\boldsymbol{\Sigma}}_V(\mathbf{s},t))$. The dimension of the data, $L(\mathbf{s},t)$, is allowed to vary based on the space-time location due to potentially missing data and we assume the errors are independent in space and time.

Within the hierarchical model, missing data are accommodated by allowing the dimension of the incidence matrix, $\mathbf{H}(\mathbf{s},t)$, to vary in time. Since missing data are handled in the data model and the latent process is fully specified, missing data do not impact the process model specification. If there are no missing data at time t and location $\mathbf{s}$, then $L(\mathbf{s},t) = N$ and $\mathbf{H}(\mathbf{s},t) = \mathbf{I}_N$. When one or more system components are missing data, the row corresponding to the missing system component is removed. For example, if we have a three-dimensional system, say $\mathbf{u}(\mathbf{s},t) = [a(\mathbf{s},t), b(\mathbf{s},t), c(\mathbf{s},t)]$ and the observation component for $b(\mathbf{s},t)$ is missing at location $\mathbf{s}$ and time t, then

$$\mathbf{H}(\mathbf{s},t) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

See Chapter 7 of Cressie and Wikle (2011) for more discussion of this approach for accommodating missing observations in hierarchical spatio-temporal models.

Incorporating the basis expansion of the process in (5.10), at location $\mathbf{s}$ and time t,

$$\mathbf{v}(\mathbf{s},t) = \mathbf{H}(\mathbf{s},t)\Theta\mathbf{A}(\boldsymbol{\phi}_{t(0)}(t) \otimes \boldsymbol{\psi}(\mathbf{s}))' + \boldsymbol{\epsilon}, \tag{5.11}$$

where $\boldsymbol{\epsilon}(\mathbf{s},t) \overset{indep.}{\sim} N_L(\mathbf{0}, \Sigma_V(\mathbf{s},t))$ and $\boldsymbol{\epsilon}(\mathbf{s},t)$ now accounts for the discrepancy between the “true” underlying process and our approximation using the basis formulation.

5.3.5 Parameter Model

The data and process equations correspond to the first and second level of our hierarchical model, respectively. For convince, we restate (5.11) and (5.9) for location $\mathbf{s}$ and time t

$$\mathbf{v}(\mathbf{s},t) = \mathbf{H}(\mathbf{s},t)\boldsymbol{\Theta}\mathbf{A}(\boldsymbol{\phi}_{t^{(0)}}(t) \otimes \boldsymbol{\psi}(\mathbf{s}))' + \boldsymbol{\epsilon}(\mathbf{s},t)$$
$$\boldsymbol{\Theta}\mathbf{A}(\boldsymbol{\phi}_{t^{(J)}}(t) \otimes g(\boldsymbol{\psi}(\mathbf{s})))' = \mathbf{Mf}(\mathbf{A}, \boldsymbol{\psi}(\mathbf{s}), \boldsymbol{\psi}_x(\mathbf{s}), \boldsymbol{\psi}_y(\mathbf{s}), ..., \boldsymbol{\phi}_{t^{(0)}}(t), ..., \boldsymbol{\omega}(\mathbf{s},t)) + \boldsymbol{\eta}(\mathbf{s},t),$$

where $\boldsymbol{\epsilon}(\mathbf{s},t) \overset{indep.}{\sim} N_{L(\mathbf{s},t)}(\mathbf{0}, \boldsymbol{\Sigma}_V(\mathbf{s},t))$ and $\boldsymbol{\eta}(\mathbf{s},t) \overset{i.i.d.}{\sim} N_N(\mathbf{0}, \boldsymbol{\Sigma}_U)$. For clarity, we present the details on the model parameters in Table 5.1. Our goal is to make inference on the unknown parameters $\mathbf{M}, \boldsymbol{\Sigma}_U, \boldsymbol{\Sigma}_V(\mathbf{s},t)$, and $\mathbf{A}$. The sparse matrix $\mathbf{M}$ identifies the nonlinear dynamic equation, $\boldsymbol{\Sigma}_U$ captures the error dependencies within the dynamic equation, $\boldsymbol{\Sigma}_V(\mathbf{s},t)$ captures the measurement uncertainty associated with the observed process, and $\mathbf{A}$ defines the smooth latent process.

To complete our Bayesian hierarchical model, we define the following priors on these parameters. We use the spike-and-slab prior (Mitchell and Beauchamp, 1988; George et al., 1993) to induce sparsity into $\mathbf{M}$. We write

$$\mathbf{M}(n)|\boldsymbol{\gamma}(n), \boldsymbol{\sigma}_U^2(n) = \prod_{d=1}^{D}[(1-\gamma(n,d))\delta_0 + \gamma(n,d)p(M(n,d)|\boldsymbol{\sigma}_U^2(n), \cdot)],$$

where $M(n,d)$ denotes coefficient d of component n, $\boldsymbol{\gamma}$ is a matrix of inclusion indicators of the same dimension as $\mathbf{M}$, δ_0 denotes the Dirac function at 0, $\boldsymbol{\sigma}_U^2(n)$ is the nth diagonal component of $\boldsymbol{\Sigma}_U$, $p(\gamma(n,d) = 1|\pi_n) = \pi(n)$, and $\pi(n) \sim Beta(a,b)$. That is, if a variable is not included (i.e., $\gamma(n,d) = 0$), then the corresponding element $M(n,d)$ is zero. If a variable is included (i.e., $\gamma(n,d) = 1$), then the corresponding element $M(n,d)$ is non-zero. There are multiple choices for the prior $p(M(n,d)|\boldsymbol{\sigma}_U^2(n), \cdot)$. We specify the g-slab prior

Model	Symbol	Description	Dimension
	Variable		
Data	$\mathbf{v}(\mathbf{s},t)$	Observed data	$L(\mathbf{s},t) \times 1$
Data	$\mathbf{H}(\mathbf{s},t)$	Mapping matrix	$L(\mathbf{s},t) \times N$
Data	$\boldsymbol{\epsilon}(\mathbf{s},t)$	Data uncertainty distribution	$L(\mathbf{s},t) \times 1$
Data	$\boldsymbol{\Sigma}_V(\mathbf{s},t)$	Measurement error covariance matrix	$L(\mathbf{s},t) \times L(\mathbf{s},t)$
Process	$\mathbf{u}(\mathbf{s},t)$	Dynamic process	$N \times 1$
Process	$\mathcal{A}$	Basis coefficient tensor	$P \times Q \times N$
Process	$\mathbf{A}$	Basis coefficient matrix (mode-3)	$N \times (P \times Q)$
Process	$\boldsymbol{\psi}(\mathbf{s})$	spatial basis function for location $\mathbf{s}$	$P \times 1$
Process	$\boldsymbol{\phi}_{t(j)}(t)$	jth order temporal basis function for time t	$Q \times 1$
Process	$\boldsymbol{\Theta}$	component basis function matrix	$N \times R$
Process	$\mathbf{M}$	Dynamic evolution matrix	$N \times D$
Process	$\mathbf{f}(\cdot)$	Feature library	$D \times 1$
Process	$\boldsymbol{\eta}(\mathbf{s},t)$	Process uncertainty distribution	$N \times 1$
Process	$\boldsymbol{\Sigma}_U$	Dynamic equation error covariance matrix	$N \times N$
	Dimension		
	T	Number of observed time points	1
	S	Number of observed spatial locations	1
	$L(\mathbf{s},t)$	Dimension of observation vector at time t and location $\mathbf{s}$	1
	N	Dimension of latent process (dynamic system)	1
	D	Number of library functions	1
	P	Number of spatial basis functions	1
	Q	Number of temporal basis functions	1
	R	Number of component basis functions	1
	J	Highest order derivative in the dynamic system	1
	Indices		
	t	Time interval, $t \in \{1,...,T\} = D_t$, $\lvert D_t \rvert = T$	1
	$\mathbf{s}$	Spatial location, $\mathbf{s} \in \{\mathbf{s}_1,...,\mathbf{s}_S\} = D_s$, $\lvert D_s \rvert = S$	1
	j	Order of the derivative, $j = 1,...,J$	1

Table 5.1: List of symbols used in the Bayesian hierarchical model.

corresponding to Zellner's g-prior (Zellner, 1986) where g is taken to be the size of the data. See Malsiner-Walli and Wagner (2016) for other potential choices and further discussion.

While other shrinkage/selection priors could be used, such as Stochastic Search Variable Selection (SSVS; George et al., 1993), LASSO (Park and Casella, 2008), or Horseshoe (Carvalho et al., 2010), we found the spike-and-slab to be preferable since it performs well with correlated predictors (Ročková and George, 2014), which is generally present in the feature library (see Section 5.4). Additionally, the posterior summary of the latent variable $\gamma(n,d)$ gives the inclusion probability for each component of $\mathbf{M}$, providing further insight into the certainty of the recovered system. For all examples presented below, we determine the identified system as composed of terms that are included with at least 50% posterior probability. However, this threshold is subjective and one could choose a different value depending on their specific application.

To estimate γ and avoid reducibility of the Markov chain, we compute the marginal posterior distribution $[\gamma|\mathbf{A},\boldsymbol{\Theta},\boldsymbol{\Phi}_{t^{(J)}},\boldsymbol{\Psi},\boldsymbol{\Sigma}_U] \propto [\mathbf{A},\boldsymbol{\Theta},\boldsymbol{\Phi}_{t^{(J)}},\boldsymbol{\Psi}|\gamma,\boldsymbol{\Sigma}_U][\gamma]$, which is obtained by integrating over the parameters subject to selection. That is,

$$[\mathbf{A},\boldsymbol{\Theta},\boldsymbol{\Phi}_{t^{(J)}},\boldsymbol{\Psi}|\gamma,\boldsymbol{\Sigma}_U] = \int\int [\mathbf{A},\boldsymbol{\Theta},\boldsymbol{\Phi}_{t^{(J)}},\boldsymbol{\Psi}|\mathbf{M},\boldsymbol{\Sigma}_U,\gamma][\mathbf{M}][\boldsymbol{\Sigma}_U]d\mathbf{M}d\boldsymbol{\Sigma}_U.$$

To make the integration analytically tractable and keep conjugacy in the BHM, we restrict $\boldsymbol{\Sigma}_U$ to be diagonally structured where $\boldsymbol{\Sigma}_U = diag(\sigma_U^2(1),...,\sigma_U^2(N))$. Each diagonal element is assigned the non-informative prior $\sigma_U^2(n) \propto 1/\sigma_U^2(n), n = 1,...,N$. Then, the probability any element is included is given as

$$p(\gamma(n,d) = 1|\cdot) = \frac{1}{1 + \frac{1-\pi(n)}{\pi(n)} R_\gamma(n,d)} \tag{5.12}$$

where

$$R_\gamma(n,d) = \frac{[\mathbf{A},\boldsymbol{\Theta},\boldsymbol{\Phi}_{t^{(J)}},\boldsymbol{\Psi}|\gamma(n,d) = 0]}{[\mathbf{A},\boldsymbol{\Theta},\boldsymbol{\Phi}_{t^{(J)}},\boldsymbol{\Psi}|\gamma(n,d) = 1]}.$$

In situations where dependence between the components is required, a different prior could be used.

There is potential for elements of the variance-covariance matrix $\Sigma_V(\mathbf{s},t)$ to have small values. Inference using traditional conjugate Inverse Gamma/Wishart priors are overly sensitive to the choice of hyperpriors when estimates are small (Gelman, 2006). Instead, we use the conjugate Half-t prior proposed by Huang and Wand (2013) for covariance estimation, which imposes less prior information and does not have as strong of influence on small estimates. We restrict the measurement error to be diagonally structured since it is often a reasonable assumption that measurement noise is independent (Cressie and Wikle, 2011) (although this restriction can be removed if warranted). Let $\Sigma_V(\mathbf{s},t) = \mathbf{H}(\mathbf{s},t) diag(\sigma_V^2(1),...,\sigma_V^2(N))\mathbf{H}(\mathbf{s},t)'$, where each diagonal element, $\sigma_V^2(1),...,\sigma_V^2(N)$, is assigned a conjugate Half-t$(2,10^{-5})$ prior.

Finally, in order to induce sparcity in the basis coefficients, we assign a Bayesian elastic net prior (Li and Lin, 2010) to $\mathbf{A}$. Specifically, our prior is

$$\pi(\mathbf{A}) \propto \exp\{-\lambda_1\|\mathbf{A}\|_1 - \lambda_2\|\mathbf{A}\|_2^2\},$$

where λ_1, λ_2 are penalty parameters. The elastic net prior helps regularize the basis coefficients against basis functions. While it is possible to specify hyperpriors for the two penalty terms, we find inference is not overly sensitive to the choice of penalty parameters and fix them each to a small value (e.g., 0.01 or 0.001).

5.4 Model Estimation

Our goal is to obtain samples from the joint posterior distribution $[\mathbf{M}, \Sigma_U, \Sigma_V, \gamma, \mathbf{A}|\cdot]$. We achieve this by sampling from the five full-conditional distributions $[\mathbf{M}|\cdot]$, $[\Sigma_U|\cdot]$, $[\Sigma_V|\cdot]$, $[\gamma|\cdot]$, and $[\mathbf{A}|\cdot]$ (see Appendix D.2 for the details of the distributions and sampling algorithm) using a Markov chain Monte Carlo (MCMC) sampling scheme. The four components $\mathbf{M}, \Sigma_U, \Sigma_V$ and γ are updated using classical Bayesian methods and $\mathbf{A}$ is updated using a stochastic gradient approach. Due to the variety of problems for which our method is applicable, some modeling choices are case specific. Additionally, some aspects of the implementation of the MCMC framework warrant a more detailed discussion. The following sections provide additional information pertaining to these model specifications and procedures.

5.4.1 Basis Coefficient Estimation

The basis coefficients, $\mathbf{A}$, completely define the latent process and all derivatives in both space and time, meaning proper estimation is crucial to the discovery process. Since $\mathbf{A}$ is embedded within the nonlinear function $\mathbf{f}(\cdot)$ (see Proposition 1) and $f(\cdot)$ is problem specific, it is difficult to estimate. To accommodate a generically specified $\mathbf{f}(\cdot)$, we use an adapted version of stochastic gradient descent (SGD) with a constant learning rate (SGDCL; Mandt et al., 2016). Whereas other approaches to estimate $\mathbf{A}$ (e.g., Expectation-Maximization or Metropolis-Hastings) could be used, SGDCL provides important advantages – a conjugate updating scheme and a reduced computational cost for any specification of $\mathbf{f}(\cdot)$.

As with SGD, SGDCL relies on the gradient of a loss function and a learning rate. For

SGDCL, the loss function is the negative log posterior for our parameters of interest, $\mathbf{A}$. The loss function at location $\mathbf{s}$ and time t is

$$
\begin{aligned}
\mathscr{L}(\mathbf{A};\mathbf{s},t) = & -\log([\mathbf{v}(\mathbf{s},t)|\mathbf{A},\mathbf{H}(\mathbf{s},t),\boldsymbol{\Theta},\boldsymbol{\phi}_{t(0)}(t),\boldsymbol{\psi}(\mathbf{s}),\boldsymbol{\Sigma}_V] \\
& [\mathbf{A},\boldsymbol{\Theta},\boldsymbol{\phi}_{t(j)}(t),g(\boldsymbol{\psi}(s))|\mathbf{M},\boldsymbol{\Sigma}_U,\mathbf{A},\boldsymbol{\Theta},\boldsymbol{\phi}_{t(0)}(t),...,\boldsymbol{\phi}_{t(J-1)}(t),\boldsymbol{\psi}(\mathbf{s}),\boldsymbol{\psi}_x(\mathbf{s}),...]) \\
& -\log([\mathbf{A}]).
\end{aligned}
$$

To simplify notation, denote $\mathbf{B}_0(\mathbf{s},t) = \boldsymbol{\phi}_{t(0)}(t)\otimes\boldsymbol{\psi}(\mathbf{s})$ and $\mathbf{B}_J(\mathbf{s},t) = \boldsymbol{\phi}_{t(J)}(t)\otimes g(\boldsymbol{\psi}(\mathbf{s}))$. Then, the gradient of the loss function $\mathscr{L}(\mathbf{A};\mathbf{s},t)$, $\frac{\partial\mathscr{L}(\mathbf{A};\mathbf{s},t)}{\partial\mathbf{A}} = \nabla_{\mathbf{A}}\mathscr{L}(\mathbf{A};\mathbf{s},t)$ for location $\mathbf{s}$ and time t is

$$
\begin{aligned}
\nabla\mathscr{L}(\mathbf{A};\mathbf{s},t) = & -\boldsymbol{\Theta}'\mathbf{H}'(\mathbf{s},t)\boldsymbol{\Sigma}_V^{-1}\mathbf{v}(\mathbf{s},t)B_0(\mathbf{s},t)+\boldsymbol{\Theta}'\mathbf{H}'(\mathbf{s},t)\boldsymbol{\Sigma}_V^{-1}\mathbf{H}(\mathbf{s},t)\boldsymbol{\Theta}\mathbf{A}\mathbf{B}_0'(\mathbf{s},t)\mathbf{B}_0(\mathbf{s},t) \\
& +\boldsymbol{\Theta}'\boldsymbol{\Sigma}_U^{-1}\boldsymbol{\Theta}\mathbf{A}\mathbf{B}_J'(\mathbf{s},t)\mathbf{B}_J(\mathbf{s},t)-\boldsymbol{\Theta}'\boldsymbol{\Sigma}_U^{-1}\mathbf{M}\mathbf{f}(\mathbf{s},t)\mathbf{B}_J(\mathbf{s},t) \\
& -\mathbf{B}_J(\mathbf{s},t)\mathbf{A}'\boldsymbol{\Theta}'\boldsymbol{\Sigma}_U^{-1}\mathbf{M}\dot{\mathbf{F}}'(\mathbf{s},t)+\mathbf{f}'(\mathbf{s},t)\mathbf{M}'\boldsymbol{\Sigma}_U^{-1}\mathbf{M}\dot{\mathbf{F}}'(\mathbf{s},t) \\
& +\frac{1}{ST}\left(\lambda_1 sign(\mathbf{A})+2\lambda_2\mathbf{A}\right),
\end{aligned}
$$

where $\dot{\mathbf{F}}(\mathbf{s},t)$ generically denotes $\frac{\partial}{\partial\mathbf{A}}\mathbf{f}(\mathbf{A},\cdot)$.

SGDCL (Mandt et al., 2016) replaces the true gradient with the stochastic estimate,

$$
\widehat{\nabla\mathscr{L}}_{\mathscr{Z}}(\mathbf{A}) = \frac{1}{|\mathscr{Z}|}\sum_{z\in\mathscr{Z}}\nabla_{\mathbf{A}}\mathscr{L}(\mathbf{A};z),
$$

where $\mathscr{Z}\subset D_s\times D_t$ is a random subset of the observations, called a mini-batch, and $|\mathscr{Z}|$ is the cardinality of the set. Within the context of a MCMC algorithm, the ℓth update of $\mathbf{A}$

is given by

$$\mathbf{A}^{(\ell)} = \mathbf{A}^{(\ell-1)} - \kappa \widehat{\nabla \mathscr{L}}_{\mathscr{X}^{(\ell)}}(\mathbf{A}^{(\ell-1)}), \tag{5.13}$$

where $\mathscr{X}^{(\ell)}$ denotes a random minibatch specific to the ℓ update and κ is the learning rate. To accommodate different scales for each component, we allow κ to be a vector of length N where each component can have a specific learning rate.

The final challenge to estimating $\mathbf{A}$ is computing $\dot{\mathbf{F}}(\mathbf{s},t)$. Because $\mathbf{f}(\cdot)$ is problem specific, $\dot{\mathbf{F}}(\mathbf{s},t)$ is also problem specific. One option is to use automatic differentiation (AD) to analytically compute the derivative of $\mathbf{f}(\cdot)$. There are many different libraries and programs that perform AD, and we explored the use of the *ForwardDiff* (Revels et al., 2016) package in Julia (Bezanson et al., 2017) with success. However, there is computational overhead to AD. For all the examples presented here we computed $\dot{\mathbf{F}}(\mathbf{s},t)$ without AD for each problem to mitigate this computation bottleneck.

5.4.2 Choice of Basis Functions

The choice of basis functions are subjective and have the potential to affect the model fit (Chapter 4 of this dissertation). Furthermore, the choice of spatial and temporal basis functions do not need to be the same (e.g., radial basis functions in space and Fourier basis functions in time). There are other choices regarding basis functions that need to be taken into consideration (see Ramsay and Silverman, 2005, Chapter 3 for a discussion on how to choose basis functions based on the "shape" of the data). The most important requirement is that the spatial and temporal basis functions need to be differentiable up to at least the highest order spatial and temporal derivative considered, respectively. We found local basis functions (e.g., B-splines) perform better than global basis functions (e.g., Fourier basis functions) (see Chapter 4 of this dissertation), especially when there are local regions with minimal curvature. For these reasons, we use B-splines of an order greater than our highest derivative (in both space and time). In choosing how many basis functions to use, enough need to be included such that the estimated solution curve is flexible, the dynamics are captured, and the posterior latent space is properly explored, but not so many such that unnecessary noise is introduced into the system. Empirically, we found a ratio of approximately 1 basis function to every 3 to 5 observations to work well.

5.4.3 Choice of Feature Library

The choice of functions for the feature library is crucial to the identification of the system. Our method is restricted to search over a predefined set of functions, meaning that our method is unable to identify an important function if it is not included in the library. For this reason, it is best to over-parameterize the feature library (and hence $\mathbf{M}$) instead of specifying a restrictive set of functions. Additionally, some knowledge of the problem is beneficial (i.e., this is not a black-box approach). Having an understanding of the potential dynamics *a priori* can assist in the recovery of important dynamics. For example, if the system appears to diffuse over time, then a diffusion term should be included. A good default choice is polynomial terms that interact with varying orders of spatial and/or temporal derivatives of the process (e.g., see the library for Burgers' example) as this will cover a wide collection of systems.

With regard to the choice of $g(\cdot)$ in (5.8) and (5.9), scientific knowledge of the problem is required. The choice of $g(\cdot)$ is not searched over as with the library terms; rather it is pre-specified. As we show in our real-world example, we knew *a prior* our goal was to make inference on the Laplacian of our system of interest. Specifically, our goal is to make inference on the vorticity of the streamfunction. Vorticity is obtained by taking the Laplacian of the streamfunction. As such, this transformation function is not learned from the data, rather, it is a modeling choice that is user specified (the default is the identity as in Proposition 1).

5.4.4 Multicollinearity in Library

A major issue facing the identification of spatio-temporal dynamic equations is multicollinearity in the feature library. Figure 5.1 shows the correlation between different components of a library using data generated from Burgers' equation. In this example, the polynomial terms, u, u^2, and u^3, are very positively linearly correlated ($\rho > 0.8$), posing a challenge to parameter inference. As with classical regression, multicollinearity has the potential to introduce bias into the coefficient estimates, including altering their sign. While the spike-and-slab has been shown to perform well with correlated variables, as discussed in Section 5.3.5, the problem still persists and can pose an estimation issue in problems similar to the example using Burgers' equation.

This issue originates from the over-inflation of minor reductions in the residual sum of squares (RSS) when a highly correlated, but incorrect term, is included. Specifically, the probability of including any term in the library (5.12) is dependent on the ratio of the residual sum of squares for the model with the $M(n,d)$ term included (RSS_γ) to the model without the $M(n,d)$ term included ($RSS_{\setminus\gamma}$) through the value of $R_\gamma(n,d)$. That is, under the g-prior

$$R_\gamma(n,d) = (g+1)^{1/2}(RSS_\gamma/RSS_{\setminus\gamma})^{ST/2-1}.$$

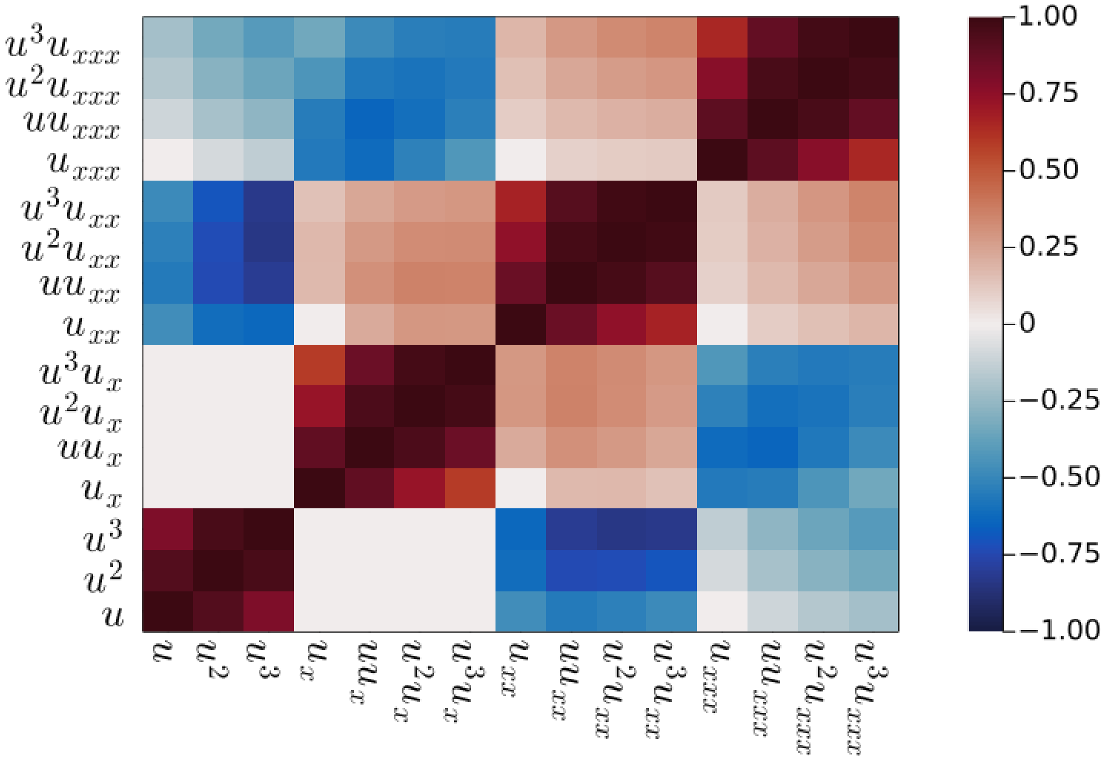


Figure 5.1: Correlation between terms of a potential feature library using data generated from Burgers' equation.

Because $RSS_\gamma < RSS_{\backslash\gamma}$, the ratio is bound between 0 and 1, where correct terms in the library result in the ratio being closer to 0 and incorrect terms result in the ratio being close to 1. However, as the ratio is raised to a power of $ST/2-1$ (proportional to the number of observations), the value of R goes to 0 as the number of observations goes to infinity, resulting in all variables being found significant. This issue is exacerbated by correlated variables, especially if there are multiple confounding variables where the system can be approximated by some linear combination of the feature library *without* the true terms being included.

To combat this issue we propose a method to reduce the impact of correlated variables (i.e., variables where in the ratio of RSSs being close to 1 are found significant). For this, we subsample the process when estimating the inclusion latent variable γ. That is, to compute (5.12) within each iteration of the Gibbs sampler, we randomly sample the process. This results in

$$R^*_\gamma(n,d) = (g+1)^{1/2}(RSS_{\setminus\gamma}/RSS_\gamma)^{S^*T^*/2-1},$$

where S^* and T^* are the size of the subsampled dimensions. We provide details on how to choose the subsample size and our choices for the examples in Appendix D.1. Note that this subsampling is only done for the γ update step of the algorithm.

5.5 Simulations

We show our proposed model is able to discover dynamic equations using data simulated from three well known systems – Burgers' equation, the heat equation, and a reaction-diffusion system. For all three examples, we investigate the impact of measurement noise on inference. We simulate measurement noise by adding mean zero Gaussian errors to the state vector. Specifically, we let $\mathbf{v}(\mathbf{s},t) = \mathbf{u}(\mathbf{s},t) + \epsilon(\mathbf{s},t)$, where $\mathbf{u}(\mathbf{s},t)$ is the simulated data, $\epsilon(\mathbf{s},t) \sim \zeta\sigma N(\mathbf{0},\mathbf{I}_N)$ is the additive noise, σ is the standard deviation of the simulated process $\mathbf{u}(\mathbf{s},t)$, and ζ is the percent of noise ranging from 0 to 1. In addition, we show how the model performs when data are missing sporadically for Burger's equation. Unless otherwise stated, all reported estimates are rounded to three significant digits for readability. For all simulations and real-world examples, we obtain 5000 posterior samples and discard the first 2500 as burn-in.

5.5.1 Burgers' Equation

Burgers' equation is a simplification of the Navier-Stokes equations, describing the speed of a fluid at a location in space and time (Bateman, 1915; Burgers, 1948). We consider Burgers' equation in one spatial dimension defined by the nonlinear PDE

$$u_t(x,t) = -u(x,t)u_x(x,t) + \nu u_{xx}(x,t), \tag{5.14}$$

where $u(x,t)$ is the speed of the fluid at location x and time t and ν is the viscosity of the fluid. Data are generated using spectral differentiation and the *Tsit5* (Tsitouras, 2011) numerical solver from the Julia package *DifferentialEquations.jl* (Rackauckas and Nie, 2017) with initial condition $u(x,0) = exp\{-(x+2)^2\}$. The simulated data consist of 256 spatial locations across 101 time points where $D_s = [-8,8]$ and $D_t = [0,10]$ (Figure 5.2). We consider four cases – no measurement noise, 2% measurement noise, 5% measurement noise, and 2% measurement noise with 5% of data missing at random.

For all four cases we specify the model with $P = 50, Q = 20, |\mathscr{X}| = 100, \kappa = 10^{-4}$ and define the feature library as

$$[u, u^2, u^3, u_x, uu_x, u^2u_x, u^3u_x, u_{xx}, uu_{xx}, u^2u_{xx}, u^3u_{xx}, u_{xxx}, uu_{xxx}, u^2u_{xxx}, u^3u_{xxx}].$$

After obtaining posterior samples, we keep only terms with greater than a 50% inclusion probability to be included in the identified equation. The recovered equations and 95% highest posterior density (HPD) interval without and with measurement noise for the included terms are shown in Table 5.2. In addition, Table 5.3 shows the next two terms that would be included in the model based on their inclusion probabilities. In all four scenarios we correctly identify the true components of the dynamic system.

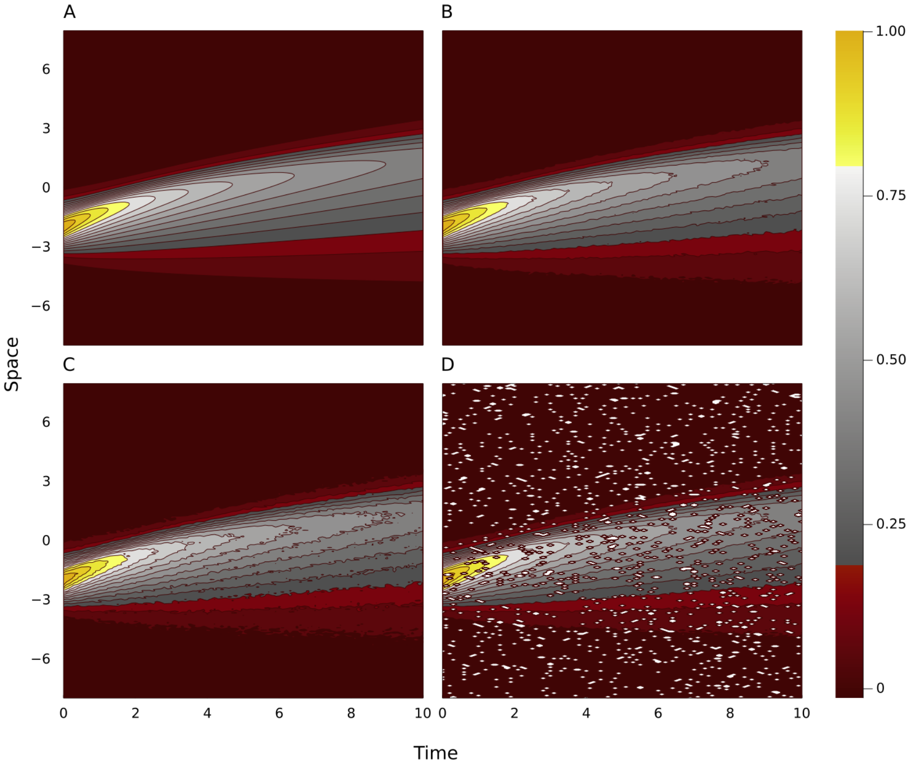


Figure 5.2: Data simulated from Burgers' equation with (A) no added measurement noise, (B) 2% added measurement noise, (C) 5% added measurement noise, and (D) 5% of data missing at random and 2% measurement noise.

The credible intervals of all parameters cover the true value with the exception of u_{xx} in the cases with 5% noise and 2% noise with 5% missing data. In addition, no extraneous terms are identified in any scenario. The probability of including another term (u_{xxx} for all four cases) is low, giving us relative certainty that the identified equation is indeed correct. Clearly, the methodology eventually will fail when measurement error is too large or there is too much missing data. For example, we found when the measurement noise is greater than 8% or more than 10% of data are missing, we no longer recover the true model.

Noise	Missing Data	Statistic	Discovered Equation
0%	0%	Mean	$u_t = -0.994uu_x + 0.098u_{xx}$
		Lower HPD	$u_t = -1.022uu_x + 0.092u_{xx}$
		Upper HPD	$u_t = -0.964uu_x + 0.103u_{xx}$
2%	0%	Mean	$u_t = -0.990uu_x + 0.096u_{xx}$
		Lower HPD	$u_t = -1.033uu_x + 0.086u_{xx}$
		Upper HPD	$u_t = -0.954uu_x + 0.103u_{xx}$
5%	0%	Mean	$u_t = -0.981uu_x + 0.094u_{xx}$
		Lower HPD	$u_t = -1.022uu_x + 0.087u_{xx}$
		Upper HPD	$u_t = -0.951uu_x + 0.099u_{xx}$
2%	5%	Mean	$u_t = -0.957uu_x + 0.087u_{xx}$
		Lower HPD	$u_t = -1.003uu_x + 0.078u_{xx}$
		Upper HPD	$u_t = -0.931uu_x + 0.095u_{xx}$

Table 5.2: Discovered Burgers' equation (mean) and lower and upper HPD intervals with varying amounts of noise and missingness. The true Burgers' equation is $u_t = -uu_x + u_{xx}$.

Noise	Missing Data	First Term	Probability	Second Term	Probability
0%	0%	u_{xxx}	0.193	uu_{xxx}	0.139
2%	0%	u_{xxx}	0.222	uu_{xxx}	0.145
5%	0%	u_{xxx}	0.235	uu_{xxx}	0.164
2%	5%	u_{xxx}	0.229	uu_{xxx}	0.150

Table 5.3: Feature library term with the highest (first term) and next highest (second term) probability of inclusion that was not included in the discovered equation for data generated from Burgers' equation.

5.5.2 Heat Equation

The two-dimensional (2D) heat equation can be used to model the dissipation of heat over time. We consider the 2D heat equation described by the PDE

$$u_t(\mathbf{s},t) = \alpha\nabla^2 u(\mathbf{s},t) = \alpha u_{xx}(\mathbf{s},t) + \alpha u_{yy}(\mathbf{s},t), \tag{5.15}$$

where $u(\mathbf{s},t)$ is the temperature of the surface at location $\mathbf{s} = (x,y)$ and time t, α is the thermal diffusivity, and $\nabla^2 = u_{xx} + u_{yy}$ denotes the Laplacian operator. Data are simulated

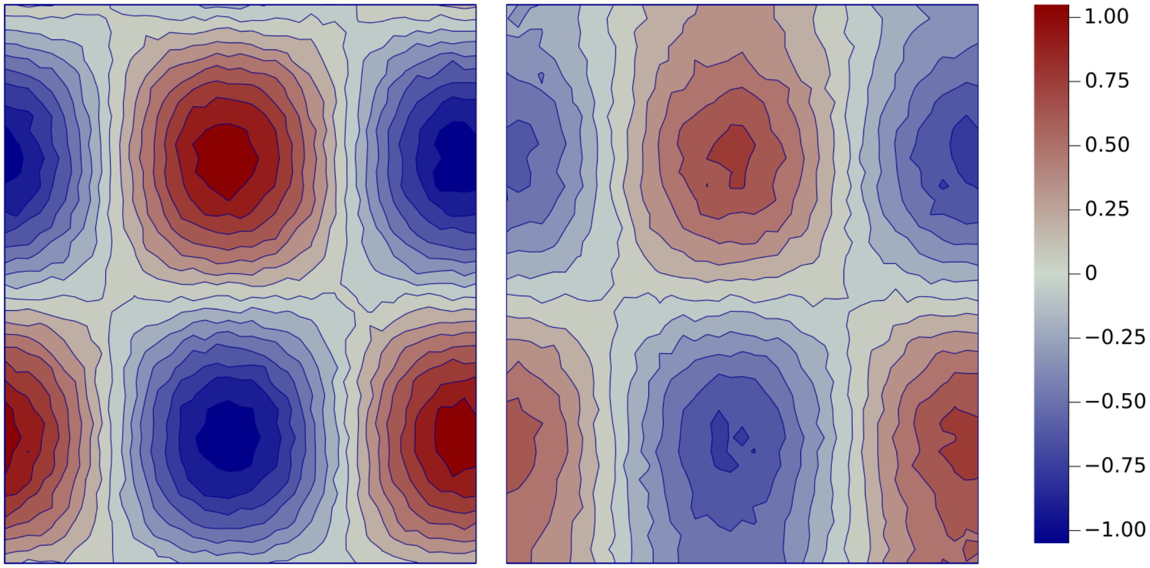


Figure 5.3: Data generated from the heat equation with 5% noise at time step 0 (left, corresponding to $t = 0$) and time step 21 (right, corresponding to $t = 0.2$).

using a central finite difference scheme over the spatial domain $D_s = [0,20] \times [0,20]$ with a spatial resolution of 0.5 for both the x and y directions, and over the time domain $D_t = [0,2]$ with a temporal resolution of 0.01. The thermal diffusivity, α, is set to 1. The surface is initialized as

$$u(\mathbf{s},0) = sin(2\pi x/40) * \cos(2\pi y/40). \tag{5.16}$$

We consider three cases – no measurement noise, 2% measurement noise, and 5% measurement noise (Figure 5.3).

For all cases we specify our model with $P = 100, Q = 80, |\mathscr{X}| = 100, \kappa = 10^{-4}$ and

Noise	Statistic	Recovered Equation
0%	Mean	$u_t = 1.001u_{xx} + 1.000u_{yy}$
	Lower HPD	$u_t = 0.999u_{xx} + 0.998u_{yy}$
	Upper HPD	$u_t = 1.002u_{xx} + 1.002u_{yy}$
2%	Mean	$u_t = 0.997u_{xx} + 1.002u_{yy}$
	Lower HPD	$u_t = 0.991u_{xx} + 0.995u_{yy}$
	Upper HPD	$u_t = 1.002u_{xx} + 1.009u_{yy}$
5%	Mean	$u_t = 0.986u_{xx} + 0.994u_{yy}$
	Lower HPD	$u_t = 0.967u_{xx} + 0.977u_{yy}$
	Upper HPD	$u_t = 1.000u_{xx} + 1.016u_{yy}$

Table 5.4: Discovered heat equation (mean) and lower and upper HPD intervals with varying amounts of noise where the true Heat equation is $u_t = u_{xx} + u_{yy}$.

define the feature library as

$$[u, u^2, u^3, u_x, uu_x, u^2u_x, u^3u_x, u_{xx}, uu_{xx}, u^2u_{xx}, u^3u_{xx}, u_y, uu_y,$$
$$u^2u_y, u^3u_y, u_{xy}, uu_{xy}, u^2u_{xy}, u^3u_{xy}, u_{yy}, uu_{yy}, u^2u_{yy}, u^3u_{yy}].$$

Again, keeping terms with greater than 50% inclusion probability, the recovered equation and 95% HPD interval are shown in Table 5.4. For all scenarios, we are able to correctly identify the true terms. All HPD intervals cover the truth, and no extraneous terms are identified. Table 5.5 shows the next two most likely terms to be included for each scenario. As with before, the probability of including a extraneous term is low, giving us reasonable confidence in to our discovered equation.

Noise	First Term	Probability	Second Term	Probability
0%	u^3u_{xy}	0.050	u^2u_{xy}	0.037
2%	u	0.015	u^2u_{xx}	0.013
5%	u	0.030	u^2u_{xx}	0.010

Table 5.5: Feature library term with the highest (first term) and next highest (second term) probability of inclusion that was not included in the discovered equation for data generated from the heat equation.

5.5.3 Reaction-Diffusion Equation

The reaction-diffusion equation can be used to model the change in concentration or density of substances over time. We consider the 2D reaction-diffusion parameterized by the PDE

$$\mathbf{u}_t(\mathbf{s},t) = \mathbf{D}\nabla^2\mathbf{u}(\mathbf{s},t) + \mathbf{g}(\mathbf{u}(\mathbf{s},t)),$$

where $\mathbf{u}(\mathbf{s},t) = [u(\mathbf{s},t), v(\mathbf{s},t)]'$ may represent the concentration or density of two processes, $\mathbf{D}$ is a diagonal matrix of the diffusion coefficient for each process, and $\mathbf{g}(\cdot)$ is the (non)linear reaction function. The reaction-diffusion equation can be used to model the interaction between a predator and prey population (Hastings, 1996; Liu et al., 2019). To represent the interaction between prey and predator populations, we let $\mathbf{u} = [u, v]'$ where u and v are the densities of the prey and predator populations, respectively. We define $\mathbf{g}(\cdot)$ to be the classic Lotka-Volterra model with a carrying capacity for the prey. Specifically,

$$\mathbf{g}(\cdot) = \begin{bmatrix} g_u(u(\mathbf{s},t), v(\mathbf{s},t)) \\ g_v(u(\mathbf{s},t), v(\mathbf{s},t)) \end{bmatrix} = \begin{bmatrix} \gamma_0 u(\mathbf{s},t) - \frac{\gamma_0}{\gamma_1}u^2(\mathbf{s},t) - \beta u(\mathbf{s},t)v(\mathbf{s},t) \\ \mu u(\mathbf{s},t)v(\mathbf{s},t) - \eta v(\mathbf{s},t) \end{bmatrix},$$

where γ_0 is the prey growth rate, γ_1 is the prey carrying capacity, β predation rate, μ is the predator growth rate, and η is the predator death rate.

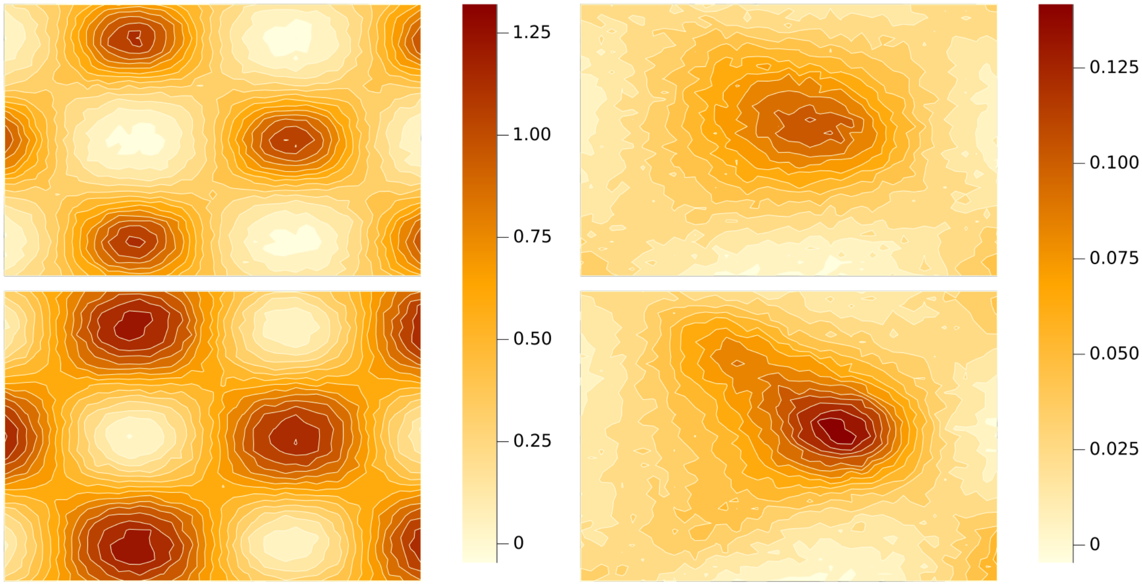


Figure 5.4: Data generated from the prey (left) and predator (right) reaction-diffusion system with 5% measurement noise. Data are shown at time steps 11 (top, corresponding to $t = 1$) and 31 (bottom, corresponding to $t = 3$).

Suppressing the spatial and temporal indices, the predator-prey reaction-diffusion equation is

$$\begin{aligned} u_t &= D_u u_{xx} + D_u u_{yy} + \gamma_0 u - \frac{\gamma_0}{\gamma_1} u^2 - \beta u v \\ v_t &= D_v v_{xx} + D_v v_{yy} + \mu u v - \eta v. \end{aligned} \tag{5.17}$$

We simulate from (5.17) with $\gamma_0 = 0.4, \gamma_1 = 1.5, \beta = 0.5, \mu = 0.3, \eta = 0.1$ using a central finite difference scheme over the spatial domain $D_s = [-10, 10] \times [-10, 10]$ and the temporal domain $D_t = [0, 10]$ with a spatial and temporal resolution of $(0.5, 0.5)$ and 0.1,

Noise	Component	Statistic	Recovered Equation
0%	Prey	Mean	$u_t = 0.400u - 0.266u^2 - 0.500uv + 0.099u_{xx} + 0.100u_{yy}$
		Lower	$u_t = 0.399u - 0.267u^2 - 0.503uv + 0.098u_{xx} + 0.099u_{yy}$
		Upper	$u_t = 0.400u - 0.266u^2 - 0.497uv + 0.100u_{xx} + 0.101u_{yy}$
0%	Predator	Mean	$v_t = -0.100v + 0.300uv + 0.099v_{xx} + 0.099v_{yy}$
		Lower	$v_t = -0.100v + 0.299uv + 0.099v_{xx} + 0.099v_{yy}$
		Upper	$v_t = -0.100v + 0.300uv + 0.100v_{xx} + 0.100v_{yy}$

Table 5.6: Discovered predator-prey reaction-diffusion equation (mean) and lower and upper HPD intervals with no noise. The true equations are $u_t = 0.4u - 0.26u^2 - 0.5uv + 0.1u_{xx} + 0.1u_{yy}$ and $v_t = 0.3uv - 0.1v + 0.1v_{xx} + 0.1v_{yy}$.

Noise	Component	Statistic	Recovered Equation
2%	Prey	Mean	$u_t = 0.403u - 0.270u^2 - 0.492uv + 0.144u_{xx} + 0.145u_{yy}$
		Lower	$u_t = 0.400u - 0.274u^2 - 0.501uv + 0.095u_{xx} + 0.097u_{yy}$
		Upper	$u_t = 0.405u - 0.267u^2 - 0.487uv + 0.163u_{xx} + 0.164u_{yy}$
2%	Predator	Mean	$v_t = -0.098v + 0.297uv + 0.136v_{xx} + 0.138v_{yy}$
		Lower	$v_t = -0.100v + 0.296uv + 0.095v_{xx} + 0.099v_{yy}$
		Upper	$v_t = -0.098v + 0.300uv + 0.161v_{xx} + 0.155v_{yy}$

Table 5.7: Discovered predator-prey reaction-diffusion equation (mean) and lower and upper HPD intervals with 2% noise. The true equations are $u_t = 0.4u - 0.26u^2 - 0.5uv + 0.1u_{xx} + 0.1u_{yy}$ and $v_t = 0.3uv - 0.1v + 0.1v_{xx} + 0.1v_{yy}$.

respectively. The prey and predator densities are initialized as

$$u(\mathbf{s},0) = \exp\{\cos(2\pi x/15)\sin(2\pi y/15)\}$$
$$v(\mathbf{s},0) = 0.1\exp\{\cos(2\pi y/30)\sin(2\pi x/30 - 5)\},$$

respectively. We again consider three scenarios – no measurement noise, 2% measurement noise, and 5% measurement noise (Figure 5.4).

For all cases we specify our model with $P = 225, Q = 40, |\mathscr{X}| = 100, \kappa = [10^{-4}, 10^{-6}]$

Noise	Component	Statistic	Recovered Equation
5%	Prey	Mean	$u_t = 0.401u - 0.269u^2 - 0.493uv + 0.090u_{xx} + 0.099u_{yy}$
		Lower	$u_t = 0.400u - 0.270u^2 - 0.504uv + 0.085u_{xx} + 0.091u_{yy}$
		Upper	$u_t = 0.403u - 0.267u^2 - 0.487uv + 0.093u_{xx} + 0.102u_{yy}$
5%	Predator	Mean	$v_t = -0.101v + 0.300uv + 0.092v_{xx} + 0.106v_{yy}$
		Lower	$v_t = -0.103v + 0.297uv + 0.086v_{xx} + 0.097v_{yy}$
		Upper	$v_t = -0.099v + 0.302uv + 0.095v_{xx} + 0.151v_{yy}$

Table 5.8: Discovered predator-prey reaction-diffusion equation (mean) and lower and upper HPD intervals with 5% noise. The true equations are $u_t = 0.4u - 0.26u^2 - 0.5uv + 0.1u_{xx} + 0.1u_{yy}$ and $v_t = 0.3uv - 0.1v + 0.1v_{xx} + 0.1v_{yy}$.

Noise	Component	First Term	Probability	Second Term	Probability
0%	Prey	u^3	0.040	u_{xy}	0.039
0%	Predator	u^2v	0.057	u_{xx}	0.036
2%	Prey	vv_y	0.022	v_{yy}	0.020
2%	Predator	v^3	0.020	vv_x	0.019
5%	Prey	v_{xx}	0.029	v_x	0.027
5%	Predator	u^2	0.024	u	0.023

Table 5.9: Feature library term with the highest (first term) and next highest (second term) probability of inclusion that was not included in the discovered equation for data generated from the reaction-diffusion equation.

and define the feature library as

$$[u, u^2, u^3, v, v^2, v^3, uv, u^2v, uv^2, uu_x, uu_y, vv_x, vv_y, u_x, u_y, u_{xx}, u_{yy}, u_{xy}, v_x, v_y, v_{xx}, v_{yy}, v_{xy}].$$

Posterior estimates of terms with greater than 50% inclusion probability are shown in Tables 5.6, 5.7, and 5.8 for the case with no noise, 2% measurement noise, and 5% measurement noise, respectively. The next two most likely terms and the probability of including each term for all cases are shown in Table 5.9. With no measurement noise, we see all terms are correctly identified and the 95% HPD intervals all cover the truth. For the scenario with 2% noise, all terms are correctly identified and all coefficients except for u^2 in the prey equation cover the truth. The scenario with 5% measurement noise correctly identifies all terms and only u^2 and u_{xx} for the prey equation and v_{xx} for the predator equation have 95% credible intervals that do not cover the truth. We again see the probability of including an extraneous term is low for all scenarios, providing further confidence to our discovered equation.

5.6 Barotropic Vorticity Equation

Here we show the ability of our model to discover the governing dynamic equations using real world data. The 500-hPa level of the atmosphere is often known as the "level of non-divergence" because in the absence of strong cyclogenesis, the flow is essentially horizontal and non-divergent. Such flows can often be modeled quite effectively with barotropic dynamics (in a barotropic fluid, the density is constant along a constant pressure surface). Indeed, the first successful numerical weather forecasts were based on the advection of relative vorticity (rotation of the fluid in the horizontal dimnension) at the 500-hPa level using the so-called barotropic vorticity equation (BVE; Charney et al., 1950). The BVE is given as

$$\xi_t(\mathbf{s},t) = -\mathbf{v}(\mathbf{s},t) \cdot \nabla(\xi(\mathbf{s},t) + f(\phi(\mathbf{s}))), \tag{5.18}$$

where $\xi = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} = \frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2}$ is the relative vorticity, $\mathbf{v} = (u, v)$ is the non-divergent horizontal wind, $u = -\frac{\partial \psi}{\partial y}$ is the Zonal wind, $v = \frac{\partial \psi}{\partial x}$ is the Meridional wind, ψ is the streamfunction, $f = 2\Omega \sin(\phi)$ is the Coriolis parameter with $\Omega = 7.292 \times 10^{-5}$ rad s^{-1} the angular speed of rotation of the Earth (note – not to be confused with the dynamic discover library as defined previously), and $\phi(\mathbf{s})$ the latitude in radians at location $\mathbf{s}$. Note that the relative vorticity can be written as the Laplacian of the streamfunction, and the non-divergent velocities of the flow can be written in terms of the gradient of the stream function (e.g., Holton and Hakim, 2012). Then, expanding the BVE in terms of the streamfunction and suppressing the spatial and temporal indices, we get

$$\nabla^2 \psi_t = \psi_y \psi_{xxx} + \psi_y \psi_{xyy} - \psi_x \psi_{xxy} - \psi_x \psi_{yyy} - \psi_x f_y, \tag{5.19}$$

where $\nabla^2 \psi_t = \psi_{xxt} + \psi_{yyt}$. Note, the streamfunction is a computed quantity based on the observed geopotential height, Φ, where $\psi(\mathbf{s},t) = \Phi^*(\mathbf{s},t)/f(\phi(\mathbf{s}))$, $\Phi^*(\mathbf{s},t) = \Phi(\mathbf{s},t) - \int_D \Phi(\mathbf{s},t)$, and D is the observed domain.

Here, we use hourly data generated using Copernicus Atmosphere Monitoring Service information (2022)[1] of relative geopotential height at 500-hPA. Data are collected hourly from December 1st to December 31, 2018 over the spatial domain $D_s = [-150°W, -50°W] \times [20°N, 60°N]$ at a resolution of 1.5 degrees (Figure 5.5), resulting in $(67 \times 27) \times 744$ space-time locations. We compute the streamfunction from the geopotential height and use this as the observed data for the discovery. We also compute the derivative of the Coriolis parameter (e.g., Rossby parameter) in the latitudinal direction, $f_y = 2\Omega \cos(\phi(\mathbf{s}))/a$, where $a = 6.371 \times 10^6$ is the radius of Earth in meters, and use it as a covariate in the model (i.e., $\omega(\mathbf{s},t) = f_y(\mathbf{s})$ from (5.9)).

We specify our model with $P = 200, Q = 300, |\mathscr{X}| = 200, \kappa = 10^{-4}$ and define the

[1] https://cds.climate.copernicus.eu/cdsapp#!/dataset/10.24381/cds.bd0915c6?tab=overview

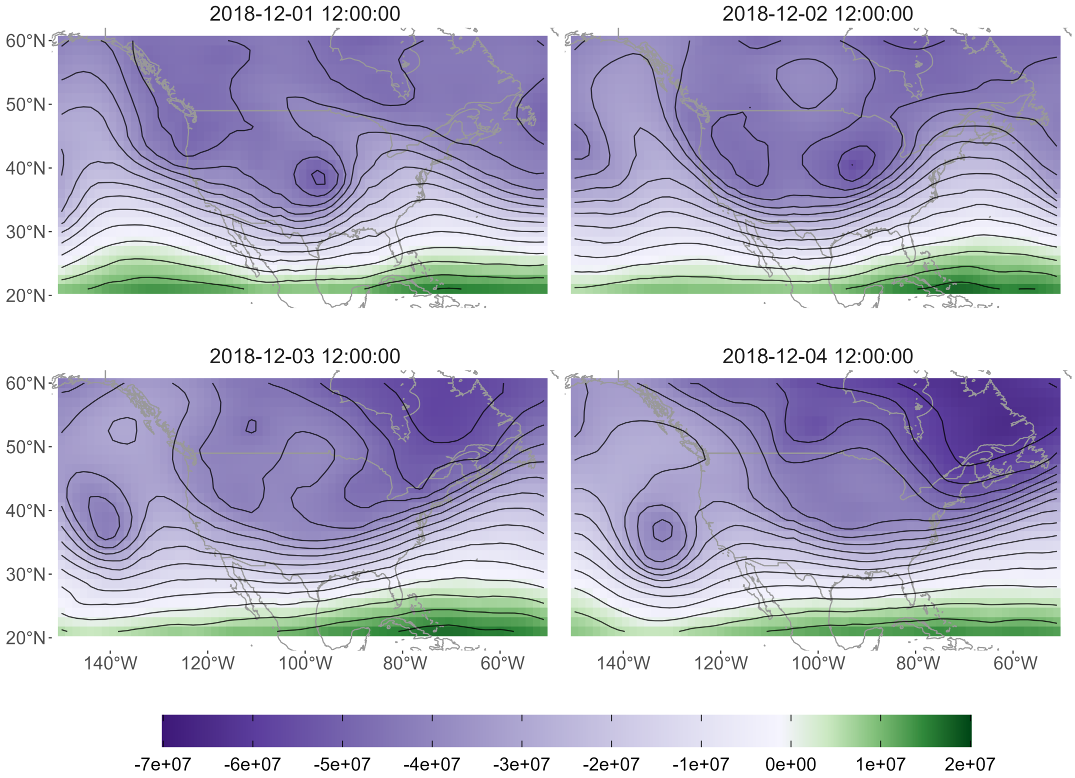


Figure 5.5: Streamfunction ($meters^2/second$) data at 12pm on December 1, 2, 3, and 4, 2018. Purple and green correspond to lower value and upper values, respectively, with contour lines included for visual aid.

feature library as

$$[\psi, \psi_x, \psi_{xx}, \psi_y, \psi_{yy}, \psi_{xy}, \psi_x\psi_{xxx}, \psi_y\psi_{xxx}, \psi_x\psi_{yyy}, \psi_y\psi_{yyy},$$

$$\psi_x\psi_{xxy}, \psi_y\psi_{xxy}, \psi_x\psi_{xyy}, \psi_y\psi_{xyy}, \psi_x f_y, \psi_y f_y]$$

Posterior estimates for the recovered equation are shown in Table 5.10, where only terms with a posterior inclusion probability greater than 50% were kept. While we do not know the true equation in this case (because the barotropic vorticity equation is only an approximation of the dynamics in the atmosphere), we see the discovered equation closely resembles the hypothesized BVE. The sign for each discovered term aligns with the sign in the BVE, however the coefficient values are different and the $\psi_x f_y$ term is not significant. This could be due to the system not being perfectly barotropic (e.g., baroclinic), which would require library terms our current framework cannot accommodate (e.g., flow in the vertical direction, density, temperature). The reason the $\psi_x f_y$ term shows up as non-significant could be due to the system experiencing predominately zonal (east-west) flow (i.e., relatively weak meridional (north-south) flow). Visual examination of the data suggests that the flow is dominated by zonal flow and this is further confirmed when we note that the magnitude of the planetary vorticity term in the BVE ($\psi_x f_y$) is an order of magnitude smaller compared to the relative vorticity terms (recall, ψ_x corresponds to the meridional component of the non-divergent velocity). Yet, the planetary vorticity term still is important to the advection of relative vorticity because of the Coriolis effect on the flow, resulting in the term being identified as important (if not significant). In our results, the coefficient for $\psi_x f_y$ is estimated to be an order of magnitude larger to make up for the scaling difference in the data, but because it is not overly influential on the system, its HPD covers zero. If the meridional flow was stronger, the scale of $\psi_x f_y$ would be larger and this would be reflected by the scale of the coefficient. Regardless, we are able to infer properties of the system of interest (i.e., the vorticity of the streamfunction) and obtain results that generally align with the theoretical equation. To the best of our knowledge, this is the first time that data-driven discovery methods have been applied to real-world atmospheric data and identified physically plausible features.

Statistic	Discovered Equation
Mean	$\nabla^2 \psi_t = 0.289 \psi_y \psi_{xxx} + 0.277 \psi_y \psi_{xyy} - 0.280 \psi_x \psi_{xxy} - 0.185 \psi_x \psi_{yyy} - 6.354 \psi_x f_y$
Lower HPD	$\nabla^2 \psi_t = 0.235 \psi_y \psi_{xxx} + 0.267 \psi_y \psi_{xyy} - 0.343 \psi_x \psi_{xxy} - 0.215 \psi_x \psi_{yyy} - 7.570 \psi_x f_y$
Upper HPD	$\nabla^2 \psi_t = 0.317 \psi_y \psi_{xxx} + 0.286 \psi_y \psi_{xyy} - 0.223 \psi_x \psi_{xxy} - 0.160 \psi_x \psi_{yyy} + 1.491 \psi_x f_y$

Table 5.10: Discovered equation for the BVE (mean) and lower and upper HPD intervals where the theoretical BVE is $\nabla^2 \psi_t = \psi_y \psi_{xxx} + \psi_y \psi_{xyy} - \psi_x \psi_{xxy} - \psi_x \psi_{yyy} - \psi_x f_y$.

5.7 Conclusion

We have proposed a data-driven approach for learning complex non-linear spatio-temporal dynamic equations that is robust to measurement noise and missing data. Our approach uses a Bayesian hierarchical model where the dynamic equation is embedded in the latent process enabling the discovery of dynamic equations within the statistical framework. Additionally, the model provides probabilistic estimates of inclusion for each component of the feature library and estimates of uncertainty for the recovered parameters, giving a deeper understanding to the dynamic system. This all stems from the expansion of the dynamic process in terms of basis functions, bypassing the need for numerical differentiation and enabling the estimate of the derivatives within a probabilistic framework.

While our proposed method is able to correctly discover the underlying dynamics as illustrated by our simulation examples, there are situations in which it will not perform well. For example, take the vorticity transport equation (a product of the Navier-Stokes equation)

$$\xi_t = \nu \nabla^2 w - \frac{1}{Re} \cdot \nabla \xi,$$

where ξ is the vorticity, ν is he viscosity of the fluid, and Re is the Reynolds number. A classic example assumes $\nu = 1$ and $Re = 100$, resulting in the coefficients of ξ_x and ξ_y

to be 0.01. While undoubtedly important for the dynamics, the effects of ξ_x and ξ_y are weak compared to the effects of ξ_{xx} and ξ_{yy}, and the proposed method will have difficulty identifying ξ_x and ξ_y.

The proposed method makes two crucial model assumptions – the data is Gaussian and the coefficients do not vary spatially. Relaxing these assumptions would make the method applicable to a wider range of real-world problems. That is, a non-Gaussian response enables the discovery of systems where the response may be binary or count data and spatially-varying coefficients accounts for the effects of environmental factors, for example, to be included. This would enable systems such as the spread of invasive species where the diffusion coefficient is non-homogeneous in space, infection diseases, or presence-absence of a species to be studied under the guise of dynamic discovery. This significant model extension is the focus of on-going work.

Improvements to the current framework should focus on the specification of the feature library and the selection prior. A feature library that is uninhibited by the users choice (i.e., the model could generate library terms) would remove user bias. This is akin to what has been proposed with symbolic regression. Also, a different choice in selection prior for the coefficients, perhaps one that also penalizes model complexity, has the potential to improve selection performance. While there are a variety of selection priors in the literature, a prior directed towards this problem will provide noticeable improvement on the identification of the system.

Chapter 6

Summary and Concluding Remarks

The field of data-driven discovery is rapidly expanding due to the availability of data, the need to understand complex systems, and increasing computational advances. However, the applicability of these methods is limited due to their treatment of observation and process uncertainty. Chapter 2 presented a system where the governing dynamics are unknown and the system is modeled using a functional form to describe its evolution. We provided a substantive literature review to current methods of data-driven discovery in Chapter 3, highlighting the need for statistical approaches to the current work. In Chapter 4 we proposed a statistical framework for the discovery of dynamic equations parameterized by ordinary differential equations that evolve over time. This framework was further expanded in Chapter 5, enabling the discovery of space-time processes parameterized by partial differential equations.

There are multiple directions the framework presented here could be extended. The most pressing extension concerns the choice of selection prior for the feature library terms. As discussed in Chapter 5, the terms in the feature library have the potential to be highly correlated, which can result in confounding and cause incorrect terms to have an overinflated inclusion probability. While the spike and slab prior is one possible solution to this issue, the choice of prior should be investigated further in order to account for complex library specifications. The development of a prior tailored to this specific problem is alluring. The ideal prior should have a penalty based on the number of selected terms, a penalty based on the correlation between parameter values, and be robust to the size of the data. In addition, the ability to assign an inclusion probability, such as with the spike and slab, is desirable so that a measure of confidence can be given to each term in the library. The inclusion probability could be replaced with a shrinkage component so long as incorrect terms are shrunk close to zero, thereby negating the effect of incorrect terms.

Another major improvement to the current framework would be to learn, rather than specify, the feature library. Similar to the symbolic regression methods, where potential functions are learned based on a function set, a method to learn the library would remove user bias. This approach is related to the selection prior, where the need for a more robust selection prior is nullified if the library is able to be learned based on the data. However, Bayesian approaches for symbolic regression are rare (Jin et al., 2019). Additionally, the adaption of symbolic regression into the proposed framework promises to be difficult. The difficulty is due to the computational requirements of both symbolic regression and the proposed framework and the ability to embed the basis representation of the dynamics into a symbolic regression framework.

To make the method applicable to more real-world systems, the framework can be extended to include spatially indexed functions. For example, the dispersal rate of avian species may not be spatially constant (Wikle, 2003; Hooten and Wikle, 2008), and allowing for this flexibility would aide in recovering the “true” dynamics. This can be achieved by allowing the coefficient matrix $\mathbf{M}$ to be spatially indexed (i.e., $\mathbf{M}(\mathbf{s})$). Then, a spatial process, such as a Gaussian process, would need to be incorporated for each term of $\mathbf{M}(\mathbf{s})$ so the dynamics "evolve" smoothly over space. However, in order to keep the discovered dynamics consistent, the identified terms at each spatial location would need to remain the same and only the coefficient values should be allowed to vary. If this were not the case, different dynamics may be identified at different neighboring locations, resulting in non-contiguous dynamics. Spatially varying dynamics could still be identified through the values of $\mathbf{M}(\mathbf{s})$, where values of $\mathbf{M}(\mathbf{s})$ may be zero over some locations and non-zero over others, producing different discovered dynamics at different locations.

Last, the data-driven discovery models in Chapters 4 and 5 can be extended to accommodate a non-Gaussian response. Specifically, the data model can be written generally as

$$\mathbf{v}(\mathbf{s},t) \sim f(\mathbf{v}(\mathbf{s},t)|\mathbf{H}(\mathbf{s},t),\mathbf{A},\boldsymbol{\Theta},\phi_{t^{(0)}}(t),\psi(\mathbf{s})),$$

where f is a distribution function. This changes the likelihood, and subsequently the update step for $\mathbf{A}$, where the estimation of $\mathbf{A}$ now depends on the choice of the distribution function f. For example, assume f is part of the exponential family,

$$f(\mathbf{v}(\mathbf{s},t)|\mathbf{A},\cdot) = h((\mathbf{s},t))\exp\left\{\xi(\mathbf{A},\cdot)\Gamma(\mathbf{v}(\mathbf{s},t)) - \alpha(\mathbf{A},\cdot)\right\},$$

where $\cdot$ represents $\mathbf{H}(\mathbf{s},t),\boldsymbol{\Theta},\phi_{t^{(0)}}(t),\psi(\mathbf{s})$. The loss function at location $\mathbf{s}$ and time t is

$$\mathscr{L}(\mathbf{A};\mathbf{s},t) = -\log(f(\mathbf{v}(\mathbf{s},t)|\mathbf{A},\cdot) - \log([\mathbf{A},\boldsymbol{\Theta},\phi_{t^{(j)}}(t),g(\psi(s))|\mathbf{M},...]) - \log([\mathbf{A}]).$$

Using the same notation as Chapter 5, denote $\mathbf{B}_0(\mathbf{s},t) = \boldsymbol{\phi}_{t^{(0)}}(t) \otimes \boldsymbol{\psi}(\mathbf{s})$ and $\mathbf{B}_J(\mathbf{s},t) = \boldsymbol{\phi}_{t^{(J)}}(t) \otimes g(\boldsymbol{\psi}(\mathbf{s}))$. Then, the gradient of the loss function for location $\mathbf{s}$ and time t is

$$\begin{aligned}
\nabla\mathscr{L}(\mathbf{A};\mathbf{s},t) =& \frac{\partial}{\partial \mathbf{A}}\xi(\mathbf{A},\cdot)\Gamma(\mathbf{v}(\mathbf{s},t))) - \frac{\partial}{\partial \mathbf{A}}\alpha(\mathbf{A},\cdot) \\
& + \boldsymbol{\Theta}'\boldsymbol{\Sigma}_U^{-1}\boldsymbol{\Theta}\mathbf{A}\mathbf{B}_J'(\mathbf{s},t)\mathbf{B}_J(\mathbf{s},t) - \boldsymbol{\Theta}'\boldsymbol{\Sigma}_U^{-1}\mathbf{M}\mathbf{f}(\mathbf{s},t)\mathbf{B}_J(\mathbf{s},t) \\
& - \mathbf{B}_J(\mathbf{s},t)\mathbf{A}'\boldsymbol{\Theta}'\boldsymbol{\Sigma}_U^{-1}\mathbf{M}\dot{\mathbf{F}}'(\mathbf{s},t) + \mathbf{f}'(\mathbf{s},t)\mathbf{M}'\boldsymbol{\Sigma}_U^{-1}\mathbf{M}\dot{\mathbf{F}}'(\mathbf{s},t) \\
& - \frac{\partial}{\partial \mathbf{A}}\log([\mathbf{A}]),
\end{aligned}$$

where $\dot{\mathbf{F}}(\mathbf{s},t)$ generically denotes $\frac{\partial}{\partial \mathbf{A}}\mathbf{f}(\mathbf{A},\cdot)$. For any member of the exponential family, we get a relatively simple form for $\nabla\mathscr{L}(\mathbf{A};\mathbf{s},t)$ given the parameters of the exponential family and a prior for $\mathbf{A}$. This can be extended to non-exponential family members, but there is no concise general form for all of the non-exponential family distribution functions. The extension to non-Gaussian data will enable the discovery of dynamics for systems with binary or count data. For example, from an ecological perspective this includes presence/absence or abundance monitoring and from an epidemiological perspective this includes monitoring the spread of disease.

Appendix A

Chapter 2 Appendix

All code can be found at https://github.com/jsnowynorth/Harmonics. This includes code for downloading and processing the data, the sampler (listed below), and processing the results.

Defining notation that will be used and restructuring equations to match those used in the sampler, let

$$\mathbf{z}_\ell(\mathbf{s}) = [\mathbf{z}_{1\ell}(\mathbf{s})', \mathbf{z}_{2\ell}(\mathbf{s})']' \sim N\left(\mathbf{X}_\ell \boldsymbol{\beta}_\ell(\mathbf{s}), \boldsymbol{\Sigma}_\varepsilon(\mathbf{s})\right), \quad \text{where}$$

$$\mathbb{X}_\ell = \begin{bmatrix} \mathbf{X}_\ell & 0 \\ 0 & \mathbf{X}_\ell \end{bmatrix}, \quad \text{and} \quad \boldsymbol{\Sigma}_{\varepsilon_\ell}(\mathbf{s}) = \begin{bmatrix} \sigma^2_{\varepsilon_1}(\mathbf{s}) I_{T_\ell} & 0 \\ 0 & \sigma^2_{\varepsilon_2}(\mathbf{s}) I_{T_\ell} \end{bmatrix}.$$

Writing the $\boldsymbol{\beta}$'s in block notation, which will be used in the update of the predictive process, let $\mathbf{B}_\ell = [\boldsymbol{\beta}_\ell(\mathbf{s}_1)', ..., \boldsymbol{\beta}_\ell(\mathbf{s}_n)']'$ and $\boldsymbol{\Sigma}_B = I_n \otimes \boldsymbol{\Sigma}_\beta$, resulting in

$$\mathbf{B}_\ell \sim N\left(\mathbf{B}_{\ell-1} + \mathbf{F}_\ell \widetilde{\mathbf{w}}^*_\ell, \boldsymbol{\Sigma}_B\right),$$

where $\mathbf{F}_\ell = \text{Block-Diag}[\mathbf{C}(\theta_{11})\mathbf{C}^*(\theta_{11}), ..., \mathbf{C}(\theta_{2p})\mathbf{C}^*(\theta_{2p})]$.

The Gibbs sampler algorithm for (2.6) is initialized by setting all parameters to some starting values. Then, for each iteration of the Gibbs sampler, parameters are updated:

1. For $\ell = 1, ..., L$, update $[\widetilde{\mathbf{w}}^*|\cdot] \sim N\left(\mathbf{V}_w^{-1}\mathbf{a}_w, \mathbf{V}_w^{-1}\right)$ where

$$\begin{aligned}\mathbf{a}_w &= \mathbf{F}_\ell' \mathbf{\Sigma}_B^{-1}\left(\mathbf{B}_\ell - \mathbf{B}_{\ell-1}\right),\\ \mathbf{V}_w &= \mathbf{F}_\ell' \mathbf{\Sigma}_B^{-1}\mathbf{F}_\ell + \mathbf{\Sigma}_W^{-1},\end{aligned}$$

   where $\mathbf{\Sigma}_W^{-1} = \text{Block-Diag}[\mathbf{C}^*(\boldsymbol{\theta}_{11})^{-1}, ..., \mathbf{C}^*(\boldsymbol{\theta}_{2p})^{-1}]$.

2. For $k = 1, ..., p$ and $j = 1, 2$, update $[\sigma_{jk}^2|\cdot] \sim IG(a, b)$, where

$$a = \frac{Lm}{2} + a_{jk} \quad \text{and} \quad b = b_{jk} + \frac{1}{2}\sum_{\ell=1}^{L} \widetilde{\mathbf{w}}_{jk\ell}^{*\prime}\mathbf{C}^*(\boldsymbol{\theta}_{jk})^{-1}\widetilde{\mathbf{w}}_{jk\ell}^*,$$

   where a_{jk} and $b + jk$ are chosen hyperpriors, and $\widetilde{\mathbf{w}}_{jk\ell}^*$ denotes the k^{th} predictive process for cycle j for all knot locations.

3. For $\mathbf{s} = \mathbf{s}_1, ..., \mathbf{s}_n$, update$[\boldsymbol{\beta}_0(\mathbf{s}_i)|\cdot] \sim N\left(\mathbf{V}_0^{-1}\mathbf{a}_0, \mathbf{V}_0^{-1}\right)$ where

$$\begin{aligned}\mathbf{a}_0 &= \mathbf{\Sigma}_\beta^{-1}(\boldsymbol{\beta}_1(\mathbf{s}_i) - \mathbf{w}_1(\mathbf{s}_i)) + \mathbf{\Sigma}_0^{-1}\boldsymbol{\mu}_0,\\ \mathbf{V}_0 &= \mathbf{\Sigma}_\beta^{-1} + \mathbf{\Sigma}_0^{-1}.\end{aligned}$$

4. For $\ell = 1,...,L$ and $\mathbf{s} = \mathbf{s}_1,...,\mathbf{s}_n$, update $[\boldsymbol{\beta}_\ell(\mathbf{s}_i)|\cdot] \sim N\left(\mathbf{V}^{-1}\mathbf{a}, \mathbf{V}^{-1}\right)$, where

$$\mathbf{a} = \mathbf{X}_\ell(\mathbf{s}_i)'\boldsymbol{\Sigma}_{\varepsilon_\ell}^{-1}(\mathbf{s}_i)\mathbf{z}_\ell(\mathbf{s}_i) + \boldsymbol{\Sigma}_\beta^{-1}\left(\boldsymbol{\beta}_{\ell-1}(\mathbf{s}_i) + \mathbf{w}_\ell(\mathbf{s}_i)\right) + \boldsymbol{\Sigma}_\beta^{-1}\left(\boldsymbol{\beta}_{\ell+1}(\mathbf{s}_i) + \mathbf{w}_{\ell+1}(\mathbf{s}_i)\right),$$

$$\mathbf{V} = \mathbf{X}_\ell(\mathbf{s}_i)'\boldsymbol{\Sigma}_{\varepsilon_\ell}^{-1}(\mathbf{s}_i)\mathbf{X}_\ell(\mathbf{s}_i) + 2\boldsymbol{\Sigma}_\beta^{-1}$$

5. $[\boldsymbol{\Sigma}_\beta|\cdot] \sim IW(\Lambda, \Xi)$, where

$$\Lambda = V + \sum_{i=1}^{n}\sum_{\ell=1}^{L}\left(\boldsymbol{\beta}_\ell(\mathbf{s}_i) - \boldsymbol{\beta}_{\ell-1}(\mathbf{s}_i) - \mathbf{w}_\ell(\mathbf{s}_i)\right)'\left(\boldsymbol{\beta}_\ell(\mathbf{s}_i) - \boldsymbol{\beta}_{\ell-1}(\mathbf{s}_i) - \mathbf{w}_\ell(\mathbf{s}_i)\right),$$

$$\Xi = nL + \xi.$$

6. For $\mathbf{s} = \mathbf{s}_1,...,\mathbf{s}_n$ and $j = 1,2$, update $[\sigma_{\varepsilon_j}^2(\mathbf{s}_i)|\cdot] \sim IG(a_\varepsilon, b_\varepsilon)$, where

$$a_\varepsilon = \frac{L*n}{2} + a \quad \text{and} \quad b_\varepsilon = b + \frac{1}{2}\sum_{\ell=1}^{L}(\mathbf{z}_{j\ell}(\mathbf{s}_i) - \mathbf{X}_\ell\boldsymbol{\beta}_{j\ell}(\mathbf{s}_i))'(\mathbf{z}_{j\ell}(\mathbf{s}_i) - \mathbf{X}_\ell\boldsymbol{\beta}_{j\ell}(\mathbf{s}_i)).$$

Appendix B

Chapter 3 Appendix

Algorithm 2: Sequential Threshold Least-Squares: SINDy

```
Input: $K, \kappa$
Data: $\mathbf{U}_{t(J)}, \mathbf{F}$
Result: $\mathbf{M}$
Initialize: $\mathbf{M} = (\mathbf{F}'\mathbf{F} + \lambda \mathbf{I})^{-1}\mathbf{F}'\mathbf{U}_{t(J)}$
for $k = 1$ to $K$ do
    $\gamma = |\mathbf{M}| < \kappa$ ;    /* Matrix identifying small coefficients */
    $\mathbf{M}(\gamma) = 0$ ;    /* Threshold $\mathbf{M}$ */
    for $n = 1, \ldots, N$ do
        $i := \gamma(n) == 0$ ;    /* Identify non-zero columns */
        $\mathbf{m}(n) = (\mathbf{F}(i)'\mathbf{F}(i))^{-1}\mathbf{F}(i)'\mathbf{U}_{t(J)}$ ;    /* Regress */
    end
end
```

Algorithm 3: Sequential Threshold Ridge Regression: PDE-FIND

```
Input: $K, \kappa, \lambda$
Data: $\mathbf{U}_{t^{(J)}}, \mathbf{F}$
Result: $\mathbf{M}$
Initialize: $\mathbf{M} = (\mathbf{F}'\mathbf{F} + \lambda\mathbf{I})^{-1}\mathbf{F}'\mathbf{U}_{t^{(J)}}$
for $k = 1$ to $K$ do
    $\gamma = |\mathbf{M}| < \kappa$ ;    /* Matrix identifying small coefficients */
    $\mathbf{M}(\gamma) = 0$ ;                              /* Threshold $\mathbf{M}$ */
    for $n = 1, ..., N$ do
        $i := \gamma(n) == 0$ ;          /* Identify non-zero columns */
        $\mathbf{m}(n) = (\mathbf{F}(i)'\mathbf{F}(i) + \lambda\mathbf{I})^{-1}\mathbf{F}(i)'\mathbf{U}_{t^{(J)}}$ ;      /* Regress */
    end
end
```

Algorithm 4: Sparse Relaxed Regularized Regression: SR3

```
Input: $K, \kappa, \lambda, tolerance$
Data: $\mathbf{U}_{t^{(J)}}, \mathbf{F}, \mathbf{W}^0$
Result: $\mathbf{M}$
Initialize: $k = 0$, $err = 2 * tolerance$, $\mathbf{M} = (\mathbf{F}'\mathbf{F} + \lambda\mathbf{I})^{-1}\mathbf{F}'\mathbf{U}_{t^{(J)}}$
while $err > tolerance$ do
    $k = k + 1$;
    $\mathbf{M}^k = \underset{\widehat{\mathbf{M}}}{argmin} \frac{1}{2}\|\mathbf{U}_{t^{(J)}} - \mathbf{F}\widehat{\mathbf{M}}'\|^2 + \frac{1}{2\nu}\|\widehat{\mathbf{M}} - \mathbf{W}^{k-1}\|^2$;
    $\mathbf{W}^k = \mathrm{prox}_{\lambda,\nu,R}(\mathbf{M}^k)$ ;    /* prox is the proximal gradient */
    $err = \|\mathbf{W}^k - \mathbf{W}^{k-1}\|/\nu$;
end
```

Algorithm 5: General Genetic Algorithm

```
Input: Stopping criteria - $\xi$, function set, fitness function - $f()$, summary statistic - $T()$
Result: Best individual
Initialize: P = Randomly generate the initial population based on the defined functional set, $\Delta_C = 2\xi, \Delta_N = 0$
while $|T(\Delta_C) - T(\Delta_N)| > \xi$ do
    $\Delta_C = f(P)$ ;   /* Evaluate fitness of current individuals */
    P = Generate new population based on reproduction, crossover, and mutation where individuals are chosen based on fitness level (i.e., higher fitness equals higher probability of being chosen) ;
    $\Delta_N = f(P)$ ;   /* Evaluate fitness of new individuals */
end
```

Appendix C

Chapter 4 Appendix

C.1 Chapter 4 Sampling Algorithm

For time point $t = 1, ..., T$, the full model is

$$\begin{aligned}
\mathbf{v}_t &= \mathbf{H}\mathbf{A}\boldsymbol{\phi}_{t^{(0)}} + \boldsymbol{\epsilon}_t, \quad \boldsymbol{\epsilon}_t \sim N_{L(t)}(\mathbf{0}, \boldsymbol{\Sigma}_V(t)) \\
\mathbf{A}\boldsymbol{\phi}_{t^{(J)}} &= \mathbf{M}\mathbf{f}(\mathbf{A}\boldsymbol{\phi}_{t^{(0)}}) + \boldsymbol{\eta}_t, \quad \boldsymbol{\eta}_t \sim N_N(\mathbf{0}, \boldsymbol{\Sigma}_U) \\
vec(\mathbf{M}) &\sim N_{ND}(\mathbf{0}, \boldsymbol{\Sigma}_M) \\
\boldsymbol{\Sigma}_M &= diag(\gamma_1^{(c_1)}, ..., \gamma_{ND}^{(c_{ND})}) \\
\boldsymbol{\Sigma}_V(t) &= \mathbf{H}(t) diag(\sigma^2(1), ..., \sigma^2(N))\mathbf{H}(t)' \\
\sigma^2(k) &\sim IG(\nu_r/2, 2\nu_r/b_{rk}), \quad k = 1, ..., N \\
b_{rk} &\sim IG(1/2, 1/B_{rk}^2), \quad k = 1, ..., N \\
\boldsymbol{\Sigma}_U &\sim IW(\nu_q - n - 1, 2\nu_q diag(1/b_{q1}, ..., 1/b_{qn})) \\
b_{qk} &\sim IG(1/2, 1/B_{qk}^2), \quad k = 1, ..., N
\end{aligned}$$

Defining notation that will be used in the sampler, let $\mathscr{X}_{1:T} \equiv [\mathbf{u}(1),...,\mathbf{u}(T)]$, $\mathscr{X}_{1:T}^{(J)} \equiv [\mathbf{u}_{t^{(J)}}(1),...,\mathbf{u}_{t^{(J)}}(T)]$, $\mathscr{F}_{1:T} \equiv [\mathbf{f}(1),...,\mathbf{f}(T)]$, and $\mathbf{m} \equiv vec(\mathbf{M})$. For each iteration of the Gibbs sampler, parameters are updated:

1. Update $[\mathbf{M}|\cdot] \sim Gau(\mathbf{V}_m^{-1}\mathbf{a}_m, \mathbf{V}_m^{-1})$, where

$$\mathbf{V}_m = \left(\mathscr{F}_{1:T}' \otimes \mathbf{I}_N\right)' (\mathbf{I}_T \otimes \boldsymbol{\Sigma}_U)^{-1} \left(\mathscr{F}_{1:T}' \otimes \mathbf{I}_N\right) + \boldsymbol{\Sigma}_M^{-1},$$
$$\mathbf{a}_m = \left(\mathscr{F}_{1:T}' \otimes \mathbf{I}_N\right)' (\mathbf{I}_T \otimes \boldsymbol{\Sigma}_U)^{-1} vec(\mathscr{X}_{1:T}^{(J)}).$$

2. For $l = 1, \ldots ND$, update $\gamma_l^{(c_l)} = v_1$ if $c_l = 1$ and $\gamma_l^{(c_l)} = v_0$ if $c_l = 0$ where $[c_l|\cdot] \sim Bern(p_l)$ and

$$p_l = \frac{\pi[\mathbf{m}(l)|c_l = 1, \cdot]}{\pi[\mathbf{m}(l)|c_l = 1, \cdot] + (1-\pi)[\mathbf{m}(l)|c_l = 0, \cdot]}$$

3. For $k = 1,...,N$, update $[\sigma^2(k)|\cdot] \sim IG(\widehat{a}_{r_k}, \widehat{b}_{r_k})$ where

$$\widehat{a}_{r_k} = (T + \nu_{r_k})/2,$$
$$\widehat{b}_{r_k} = (\nu_{r_k}/b_{r_k}) + \frac{1}{2}\sum_{t=1}^{T}(v(k,t) - H(k,t)\mathbf{A}(k)\phi_{t^{(0)}}(t))^2,$$

where $H(k,t)$ is the kth row of $\mathbf{H}(t)$.

4. For $k = 1,...,N$, update $[b_{r_k}|\cdot] \sim IG(\widehat{A}_{r_k}, \widehat{B}_{r_k})$ where

$$\widehat{A}_{r_k} = (\nu_{r_k} + 1)/2,$$
$$\widehat{B}_{r_k} = (\nu_{r_k}/\sigma^2(k)) + 1/B_{r_k}.$$

5. Update $[\boldsymbol{\Sigma}_U|\cdot] \sim IW(\widehat{\nu}, \widehat{\boldsymbol{\Psi}})$ where

$$\widehat{\nu} = \nu_q + T - 1,$$

$$\widehat{\boldsymbol{\Psi}} = 2 * \nu_q * diag(1/b_{q_1}, ..., 1/b_{q_n}) + (\mathscr{X}^{(J)}_{1:T} - \mathbf{M}\mathscr{F}_{1:T})'(\mathscr{X}^{(J)}_{1:T} - \mathbf{M}\mathscr{F}_{1:T}).$$

6. For $k = 1, ..., N$, update $[b_{q_k}|\cdot] \sim IG(\widehat{A}_{q_k}, \widehat{B}_{q_k})$ where

$$\widehat{A}_{q_k} = (\nu_{q_k} + 1)/2,$$

$$\widehat{B}_{q_k} = (\nu_{q_k}(\boldsymbol{\Sigma}_U)^{-1}_{kk}) + 1/B_{q_k}.$$

7. Update $[\mathbf{A}|\cdot]$ from (4.5).

C.2 Chapter 4 Tables

	System	Equation
True Equation	dx/dt	$-10x + 10y$
	dy/dt	$28x - 1y - 1xz$
	dz/dt	$-2.667z + 1xy$
Posterior Mean	dx/dt	$-8.894x + 9.321y$
	dy/dt	$25.438x - 0.914xz$
	dz/dt	$-2.669z + 0.942xy$
95% Credible Interval	dx/dt	$(-10.325, -7.538)x + (8.393, 10.305)y$
	dy/dt	$(23.697, 27.060)x + (-1.035, -0.791)xz$
	dz/dt	$(-2.822, -2.518)z + (0.810, 1.063)xy$

Table C.1: Parameter estimates and 95% credible intervals (lower bound, upper bound) for the Lorenz-63 simulation under scenario (2) with measurement noise ($c = 1$).

	System	Equation
True Equation	dx/dt	$-10x+10y$
	dy/dt	$28x-1y-1xz$
	dz/dt	$-2.667z+1xy$
Posterior Mean	dx/dt	$-8.067x+8.847y$
	dy/dt	$24.749x-0.883xz$
	dz/dt	$-2.696z+0.875xy$
95% Credible Interval	dx/dt	$(-9.695,-6.533)x+(7.791,9.997)y$
	dy/dt	$(22.432,26.811)x+(-1.028,-0.728)xz$
	dz/dt	$(-2.886,-2.516)z+(0.713,1.019)xy$

Table C.2: Parameter estimates and 95% credible intervals (lower bound, upper bound) for the Lorenz-63 simulation under scenario (3) with measurement noise ($c=5$).

	System	Equation
True Equation	dx/dt	$-10x+10y$
	dy/dt	$28x-1y-1xz$
	dz/dt	$-2.667z+1xy$
Posterior Mean	dx/dt	$-7.154x+8.260y$
	dy/dt	$24.346x-0.865xz$
	dz/dt	$-2.674z+0.780xy$
95% Credible Interval	dx/dt	$(-8.653,-5.609)x+(7.095,9.445)y$
	dy/dt	$(22.339,26.409)x+(-1.005,-0.723)xz$
	dz/dt	$(-2.868,-2.472)z+(0.594,0.942)xy$

Table C.3: Parameter estimates and 95% credible intervals (lower bound, upper bound) for the Lorenz-63 simulation under scenario (4) with measurement noise ($c=10$).

	System	Equation
True Equation	dx/dt	$-10x+10y$
	dy/dt	$28x-1y-1xz$
	dz/dt	$-2.667z+1xy$
Posterior Mean	dx/dt	$-8.587x+9.088y$
	dy/dt	$+26.086x-0.911xz$
	dz/dt	$-2.465z+0.918xy$
95% Credible Interval	dx/dt	$(-10.173,-6.942)x+(7.931,10.132)y$
	dy/dt	$(24.037,28.186)x+(-1.052,-0.774)xz$
	dz/dt	$(-2.658,-2.285)z+(0.722,1.093)xy$

Table C.4: Parameter estimates and 95% credible intervals (lower bound, upper bound) for the Lorenz-63 simulation under scenario (5) with measurement noise ($c=1$) and missing data.

Principal	ENSO97		ENSO15	
Component	Linear	Polynomial	Linear	Polynomial
1	5	12	8	11
2	4	10	4	11
3	3	5	6	12
4	3	13	2	4
5	2	9	3	12
6	3	7	2	7
7	3	5	2	6
8	2	5	1	7
9	2	5	0	8
10	1	8	2	4

Table C.5: Number of linear and polynomial terms that were included with greater than 99% probability for the model fit to ENSO97 (left) and ENSO15 (Right).

Appendix D

Chapter 5 Appendix

D.1 Inclusion Probabilities for Spike and Slab

Let the residual sum of squares for the model with the $M(n,d)$ term included be RSS_γ and the model without the $M(n,d)$ term included be $RSS_{\setminus\gamma}$. The probability any element is included is given as

$$p(\gamma(n,d) = 1|\cdot) = \frac{1}{1 + \frac{1-\pi(n)}{\pi(n)} R_\gamma(n,d)}$$

where

$$R_\gamma(n,d) = (g+1)^{1/2} \left(\frac{RSS_\gamma}{RSS_{\setminus\gamma}} \right)^{ST/2-1} .$$

Denote $\beta = \frac{RSS_\gamma}{RSS_{\setminus\gamma}}$ and solving for the number of observations ST,

$$n_{obs} := ST = 2\left(\log\left(\frac{R_\gamma(n,d)}{(g+1)^{1/2}}\right) \Big/ \log(\beta) + 1\right).$$

We then use the value of n_{obs} to inform our subsample size based on g, the ratio of the RSSs, and an informed value of R. For example, assume $\pi = 0.5$ such that every parameter has a 50% chance of being included in the model. We would take $R = 1$ resulting in $p(\gamma(n,d) = 1|\cdot) = 0.5$. n_{obs} is then chosen by solving the equation under a hypothetical β (e.g., 0.99 or 0.95).

In choosing the value β there are a couple things to consider. If the terms in the library are highly correlated, there is likely to be confounding and incorrect variables may have a larger impact on the RSS. This issue is detected using the condition number of the correlation matrix (e.g., $\mathbf{F}^{*'}\mathbf{F}^{*}$ where $\mathbf{F}^{*}$ is the normalized version of $\mathbf{F}$), where a large condition number (e.g., greater than 1000) indicates multicollinearity. Under the scenario where the condition number is large, β should be chosen to be smaller (i.e., 0.9 or 0.95), resulting in a smaller subsampled size. Alternatively, if the variables are less correlated (i.e., condition number smaller than 1000), the impact of an incorrect variable on the RSS will be less. In this case β can be chosen to be closer to one (e.g., 0.99 or 0.999), resulting in a larger subsample size. When fitting the model, the parameter π will be estimated and will (likely) not be 0.5. However, empirically we have found taking $\pi = 0.5$ and $R = 1$ to solve for n_{obs} works well.

For the simulations, the condition number of the correlation matrix for Burgers' is approximately 28940, the Heat equation is approximately 1840, and the reaction-diffusion equation is approximately 6510. We chose β to be 0.9, 0.99, and 0.95 for Burgers', the Heat, and the reaction-diffusion equations, respectively. For the real-world example, the condition number of the correlation matrix is approximately 480 and we chose β to be 0.99.

A parallel can be drawn between the subsampling and sequential thresholded least squares (STLS; Brunton et al., 2016) or sequential threshold ridge regression (STRidge; Rudy et al., 2017). Because the inclusion probability is affected by the subsample size (i.e., reduces the inclusion probability or highly unlikely terms), this is analogous to a probabilistic extension of the thresholding approaches. That is, instead of assigning a hard threshold, where values less than a predetermined value are set to zero, the subsampling approach impacts the probability of a variable being included below a certain threshold based on a specified β.

D.2 Sampling Algorithm

To simplify notation, denote $\mathbf{B}_0(\mathbf{s},t) = \boldsymbol{\phi}_{t^{(0)}}(t) \otimes \boldsymbol{\psi}(\mathbf{s})$ and $\mathbf{B}_J(\mathbf{s},t) = \boldsymbol{\phi}_{t^{(J)}}(t) \otimes g(\boldsymbol{\psi}(\mathbf{s}))$. At time t and location $\mathbf{s}$, the full model is

$$
\begin{aligned}
\mathbf{v}(\mathbf{s},t) &\sim N(\mathbf{\Theta}\mathbf{A}\mathbf{B}_0'(\mathbf{s},t), \mathbf{\Sigma}_V(\mathbf{s},t)) \\
\mathbf{\Theta}\mathbf{A}\mathbf{B}_J'(\mathbf{s},t) &\sim N(\mathbf{M}\mathbf{f}(\mathbf{s},t), \mathbf{\Sigma}_U) \\
\mathbf{M}(n)|\gamma(n), \boldsymbol{\sigma}_U^2(n) &= \prod_{d=1}^{D}[(1-\gamma(n,d))\delta_0 + \gamma(n,d)p(M(n,d)|\boldsymbol{\sigma}_U^2(n),\cdot)] \\
p(\gamma_{nd} = 1|\pi_n) &= \pi_n \\
\pi &\sim Beta(a,b) \\
\mathbf{\Sigma}_V &= \mathbf{H}(\mathbf{s},t)diag(\sigma_{V1}^2,...,\sigma_{VN}^2)\mathbf{H}'(\mathbf{s},t) \\
\sigma_{Vn}^2 &\sim IG(\nu_V/2, \nu_y/a_{Vn}) \\
\mathbf{\Sigma}_U &= diag(\sigma_{U1}^2,...,\sigma_{UN}^2) \\
\sigma_{Un}^2 &\propto 1/\sigma_{Un}^2.
\end{aligned}
$$

For iteration $\ell = 1,...,\mathscr{L}$ do:

1. Obtain minibatch $\mathscr{Q}$.
2. Update $\boldsymbol{\gamma}$: Subsample data based on choice of n_{obs}. Denote $\mathbf{f}_\gamma$ as the design matrix consisting only of columns of $\mathbf{f}$ corresponding to non-zero effects, $\mathbf{G}_\gamma = \frac{g}{g+1}(\mathbf{f}_\gamma'\mathbf{f}_\gamma)^{-1}, \mathbf{g}_\gamma = \mathbf{G}_\gamma\mathbf{f}_\gamma'\mathbf{\Theta}\mathbf{A}(n)\mathbf{B}_J'$, and $\mathbf{y}_\gamma = \frac{1}{2}((\mathbf{\Theta}\mathbf{A}(n)\mathbf{B}_J')'(\mathbf{\Theta}\mathbf{A}(n)\mathbf{B}_J') - \mathbf{g}_\gamma'\mathbf{G}_\gamma^{-1}\mathbf{g}_\gamma)$. Let $\mathbf{G}_{\gamma,0}, \mathbf{g}_{\gamma,0}$, and $\mathbf{y}_{\gamma,0}$ correspond to $\gamma(d) = 0$ and $\mathbf{G}_{\gamma,1}, \mathbf{g}_{\gamma,1}$, and $\mathbf{y}_{\gamma,1}$ correspond to $\gamma(d) = 1$. Sample each element of $\gamma_{nd}, n = 1,...,N, d = 1,...,D$ of the indicator vector $\boldsymbol{\gamma}$ from

   $$p(\gamma_{nd} = 1|\boldsymbol{\gamma}_{\backslash nd}, \mathbf{\Theta}, \mathbf{A}(n), \mathbf{B}_J') = \frac{1}{1 + R_\gamma(n,d)\frac{1-\pi_n}{\pi_n}},$$

   where $R_\gamma(n,d) = (g+1)^{1/2}\frac{\mathbf{y}_{\gamma,1}}{\mathbf{y}_{\gamma,0}}^{n_{obs}/2-1}$.

3. Update $\boldsymbol{\pi}$: For $n = 1,...,N$, sample

$$[\pi_n|\cdot] \sim Beta\left(a + \sum_d \gamma_{nd}, b + D - \sum_d \gamma_{nd}\right).$$

4. Update $\mathbf{M}$: For $n = 1,...,N$, set $\mathbf{M}_{nd} = 0$ is $\gamma_{nd} = 0$. For non-zero elements, sample

$$[\mathbf{M}_n|\cdot] \sim Gau\left(\mathbf{g}_\gamma, \sigma^2_{Un}\mathbf{G}_\gamma\right).$$

5. Update $\boldsymbol{\Sigma}_U$: For $n = 1,...,N$, sample

$$[\sigma^2_{Un}|\cdot] \sim IG\left(\frac{N-1}{2}, \frac{1}{2}\left((\boldsymbol{\Theta}\mathbf{A}_{(3)}\mathbf{B}'_J(\mathbf{s},t))'(\boldsymbol{\Theta}\mathbf{A}_{(3)}\mathbf{B}'_J(\mathbf{s},t)) - \mathbf{g}'_\gamma\mathbf{G}^{-1}_\gamma\mathbf{g}_\gamma\right)\right).$$

6. Update $\boldsymbol{\Sigma}_V$: For $n = 1,...,N$, sample

$$[\sigma^2_{Vn}|\cdot] \sim IG\Bigg(\frac{TS+\nu_V}{2}, \frac{\nu_V}{a_V} + \frac{1}{2}\sum_{t=1}^{T}\sum_{s=1}^{S}(V(\mathbf{s},t,n) - \boldsymbol{\theta}(n)\mathbf{A}B'_0(\mathbf{s},t))(V(\mathbf{s},t,n) - \boldsymbol{\theta}(n)\mathbf{A}B'_0(\mathbf{s},t))'\Bigg)$$

and

$$[a_{Vn}|\cdot] \sim IG\left(\frac{\nu_V+1}{2}, \frac{\nu_V}{\sigma^2_{Vn} + \frac{1}{A^2_V}}\right)$$

7. Update $\mathbf{A}$: Use (4.5) to update $\mathbf{A}$.

D.3 Proof of Propositions

Proposition 1. *The mode-3 decomposition of* $[\![\mathcal{A};\boldsymbol{\Psi},\boldsymbol{\Phi}_{t^{(J)}},\boldsymbol{\Theta}]\!] = \mathcal{F}\times_3\mathbf{M}+\widetilde{\boldsymbol{\eta}}$ *where* $\boldsymbol{\eta}(\mathbf{s},t)\overset{i.i.d.}{\sim} N_N(\mathbf{0},\Sigma_U)$ *in space and time at location* $\mathbf{s}$ *and time* t *is*

$$\boldsymbol{\Theta}\mathbf{A}(\boldsymbol{\phi}_{t^{(J)}}(t)\otimes\boldsymbol{\psi}(\mathbf{s}))' =$$
$$\mathbf{M}\mathbf{f}(\mathbf{A},\boldsymbol{\psi}(\mathbf{s}),\boldsymbol{\psi}_x(\mathbf{s}),\boldsymbol{\psi}_y(\mathbf{s}),\boldsymbol{\psi}_{xy}(\mathbf{s}),...,\boldsymbol{\phi}_{t^{(0)}}(t),...,\boldsymbol{\phi}_{t^{(J)}}(t),\boldsymbol{\omega}(\mathbf{s},t))+\boldsymbol{\eta}(\mathbf{s},t),$$

where $\mathbf{A}$ *is a* $R\times PQ$ *matrix of basis coefficients,* $\boldsymbol{\psi}(\mathbf{s})$ *is a length-P vector of spatial basis functions,* $\boldsymbol{\phi}(t)$ *is a length-Q vector of temporal basis functions, and* $\boldsymbol{\Theta}$ *is a* $N\times R$ *matrix of component basis functions.*

Proof of Proposition 1. For the LHS, $\boldsymbol{\Theta}\mathbf{A}(\boldsymbol{\Phi}_{t^{(J)}}\otimes\boldsymbol{\Psi})'$ is the mode-3 matricization of $\mathcal{A}\times_1\boldsymbol{\Psi}\times_2\boldsymbol{\Phi}_{t^{(J)}}\times_3\boldsymbol{\Theta}$ (see Kolda, 2006, for a proof of this property). For the RHS, from the property of the n-mode product, $\mathcal{F}\times_3\mathbf{M}=\mathbf{M}\mathbf{f}(\cdot)$, where $\mathbf{f}(\cdot)$ is the mode-3 matricization of $\mathcal{F}$. The arguments of $\mathbf{f}(\cdot)$, namely $\mathbf{U},\mathbf{U}_x,\mathbf{U}_y,...,\mathbf{U}_{t^{(0)}},...,\mathbf{U}_{t^{(J-1)}}$, are represented using their basis expansion, resulting in $\mathbf{f}(\cdot)$ depending on $\boldsymbol{\Psi},\boldsymbol{\Phi},\boldsymbol{\Theta},\mathcal{A}$ and any derivatives of $\boldsymbol{\Psi}$ and $\boldsymbol{\Phi}$ needed for the library. The value at a specific space-time location is determined from the $\mathbf{s}$th and tth column of $\boldsymbol{\Psi}$ and $\boldsymbol{\Phi}$, respectively. The last term on the RHS, $\boldsymbol{\eta}(\mathbf{s},t)$, is a the mode-3 matricization of the uncertainty tensor $\widetilde{\boldsymbol{\eta}}$, where each space-time location has the same variance-covariance matrix Σ_U.

Proposition 2. *Let* $g(\cdot)$ *be a linear differential operator. The basis formulation of a PDE* *with a space-time response* $g(\mathbf{u}_{t^{(J)}}(\mathbf{s},t))$ *is*

$$\mathbf{\Theta A}(\boldsymbol{\phi}_{t^{(J)}}(t) \otimes g(\boldsymbol{\psi}(\mathbf{s})))'.$$

Proof of Proposition 2. Let $g(\mathcal{U}) = \{g(u(\mathbf{s},t,n)) : \mathbf{s} \in D_s, t = 1,...,T, n = 1,...,N\}$ be a function of the tensor of the continuous process observed at discrete space-time locations. Decomposing $g(\mathcal{U})$ in terms of spatial, temporal, and component basis functions, $g(\mathcal{U}) \approx g([\![\mathcal{A}; \mathbf{\Psi}, \mathbf{\Phi}_{t^{(J)}}, \mathbf{\Theta}]\!])$. From Proposition 1, the mode-3 basis decomposition of $g([\![\mathcal{A}; \mathbf{\Psi}, \mathbf{\Phi}_{t^{(J)}}, \mathbf{\Theta}]\!])$ is $g(\mathbf{\Theta A}(\mathbf{\Phi}_{t^{(J)}} \otimes \mathbf{\Psi}))'$. By linearity of $g(\cdot)$ and properties of the Kronecker product,

$$g(\mathbf{\Theta A}(\mathbf{\Phi}_{t^{(J)}} \otimes \mathbf{\Psi}))' = \mathbf{\Theta A}(\mathbf{\Phi}_{t^{(J)}} \otimes g(\mathbf{\Psi}))'.$$

The function at time t and location $\mathbf{s}$ is $\mathbf{\Theta A}(\boldsymbol{\phi}_{t^{(J)}}(t) \otimes g(\boldsymbol{\psi}(\mathbf{s})))'$.

Bibliography

Ahvanooey, M. T., Li, Q., Wu, M., and Wang, S. (2019). A survey of genetic programming and its applications. *KSII Transactions on Internet and Information Systems*, 13(4):1765–1794.

Amir Haeri, M., Ebadzadeh, M. M., and Folino, G. (2017). Statistical genetic programming for symbolic regression. *Applied Soft Computing Journal*, 60:447–469.

Arnell, N. W. and Gosling, S. N. (2016). The impacts of climate change on river flood risk at the global scale. *Climatic Change*, 134(3):387–401.

Atkinson, S., Subber, W., Wang, L., Khan, G., Hawi, P., and Ghanem, R. (2019). Data-driven discovery of free-form governing differential equations. *arXiv preprint arXiv:1910.05117*, pages 1–7.

Banerjee, S., Carlin, B. P., and Gelfand, A. E. (2014). *Hierarchical modeling and analysis for spatial data*. Chapman and Hall/CRC, 2 edition.

Barnston, A. G., He, Y., and Glantz, M. H. (1999). Predictive skill of statistical and dynamical climate models in SST forecasts during the 1997—98 El Niño episode and the 1998 La Niña onset. *Bulletin of the American Meteorological Society*, 80(2):217–243.

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

Bentz, B. J., Régnière, J., Fettig, C. J., Hansen, E. M., Hayes, J. L., Hicke, J. A., Kelsey, R. G., Negrón, J. F., and Seybold, S. J. (2010). Climate change and Bark Beetles of the Western United States and Canada: Direct and indirect effects. *BioScience*, 60(8):602–613.

Berliner, L. M. (1996). Hierarchical Bayesian time series models. In *Maximum Entropy and Bayesian Methods*, pages 15–22. Springer Netherlands, Dordrecht.

Berliner, L. M. (2003). Physical-statistical modeling in geophysics. *Journal of Geophysical Research: Atmospheres*, 108(D24).

Bezanson, J., Edelman, A., Karpinski, S., and Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1):65–98.

Bolker, B. and Grenfell, B. (1995). Space, persistence and dynamics of measles epidemics. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 348(1325):309–320.

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

Boninsegna, L., Nüske, F., and Clementi, C. (2018). Sparse learning of stochastic dynamical equations. *The Journal of Chemical Physics*, 148(24):241723.

Both, G.-J., Choudhury, S., Sens, P., and Kusters, R. (2021). DeepMoD: Deep learning for model discovery in noisy data. *Journal of Computational Physics*, 428(1):109985.

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.

Bulmer, M. G. (1974). A statistical analysis of the 10-year cycle in Canada. *Journal of Animal Ecology*, 43(3):701–718.

Burgers, J. (1948). A mathematical model illustrating the theory of turbulence. In *Advances in Applied Mechanics*, volume 1, pages 171–199. Elsevier.

Carvalho, C. M., Polson, N. G., and Scott, J. G. (2010). The horseshoe estimator for sparse signals. *Biometrika*, 97(2):465–480.

Champion, K., Lusch, B., Kutz, J. N., and Brunton, S. L. (2019). Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, 116(45):22445–22451.

Champion, K., Zheng, P., Aravkin, A. Y., Brunton, S. L., and Kutz, J. N. (2020). A unified sparse optimization framework to learn parsimonious physics-informed models from data. *IEEE Access*, 8:169259–169271.

Charney, J. G., FjÖrtoft, R., and Neumann, J. V. (1950). Numerical integration of the barotropic vorticity equation. *Tellus*, 2(4):237–254.

Chartrand, R. (2011). Numerical differentiation of noisy, nonsmooth data. *ISRN Applied Mathematics*, 2011:1–11.

Chavez-Demoulin, V. and Davison, A. C. (2005). Generalized additive modelling of sample extremes. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 54(1):207–222.

Chen, Q., Zhang, M., and Xue, B. (2017). Feature selection to improve generalization of genetic programming for high-dimensional symbolic regression. *IEEE Transactions on Evolutionary Computation*, 21(5):792–806.

Cleveland, W. S. and Devlin, S. J. (1988). Locally weighted regression: An approach to regression analysis by local fitting. *Journal of the American Statistical Association*, 83(403):596–610.

Combettes, P. L. and Pesquet, J.-C. (2011). Proximal splitting methods in signal processing. In *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pages 185–212. Springer New York.

Cooley, D. and Sain, S. R. (2010). Spatial hierarchical modeling of precipitation extremes from a regional climate model. *Journal of Agricultural, Biological, and Environmental Statistics*, 15(3):381–402.

Cressie, N. A. C. (1993). *Statistics for spatial data*. Wiley Series in Probability and Statistics. John Wiley & Sons, Inc., Hoboken, NJ, USA.

Cressie, N. A. C. and Wikle, C. K. (2011). *Statistics for spatio-temporal data*. John Wiley & Sons.

de Silva, B., Champion, K., Quade, M., Loiseau, J.-C., Kutz, J., and Brunton, S. (2020). PySINDy: A Python package for the sparse identification of nonlinear dynamical systems from data. *Journal of Open Source Software*, 5(49):2104.

Defriez, E. J., Sheppard, L. W., Reid, P. C., and Reuman, D. C. (2016). Climate change-related regime shifts have altered spatial synchrony of plankton dynamics in the North Sea. *Global Change Biology*, 22(6):2069–2080.

Dwyer, J. G., Biasutti, M., and Sobel, A. H. (2012). Projected changes in the seasonal cycle of surface temperature. *Journal of Climate*, 25(18):6359–6374.

Elton, C. and Nicholson, M. (1942). The ten-year cycle in numbers of the lynx in Canada. *Journal of Animal Ecology*, 11(2):215–244.

Fasel, U., Kutz, J. N., Brunton, B. W., and Brunton, S. L. (2021). Ensemble-SINDy: Robust sparse model discovery in the low-data, high-noise limit, with active learning and control. *arXiv preprint arXiv:2111.10992*, pages 1–18.

Finley, A. O., Banerjee, S., and Gelfand, A. E. (2012). Bayesian dynamic modeling for large space-time datasets using Gaussian predictive processes. *Journal of Geographical Systems*, 14(1):29–47.

Galioto, N. and Gorodetsky, A. A. (2020). Bayesian system ID: Optimal management of parameter, model, and measurement uncertainty. *Nonlinear Dynamics*, 102(1):241–267.

Garg, A. and Tai, K. (2012). Review of genetic programming in modeling of machining processes. *Proceedings of 2012 International Conference on Modelling, Identification and Control, ICMIC 2012*, pages 653–658.

Gauss, C. F. (1809). Theoria motus corporum coelestium in sectionibus conicis solem ambientium.

Gelfand, A. E., Diggle, P., Guttorp, P., and Fuentes, M. (2010). *Handbook of spatial statistics*. CRC press.

Gelfand, A. E., Fuentes, M., Hoeting, J. A., and Smith, R. L. (2017). *Handbook of environmental and ecological statistics*. CRC Press.

Gelman, A. (2006). Prior distributions for variance parameters in hierarchical models (comment on article by Browne and Draper). *Bayesian Analysis*, 1(3):515 – 534.

George, E. I. and McCulloch, R. E. (1997). Approaches for Bayesian variable selection. *Statistica Sinica*, 7(2):339–373.

George, E. I., Mcculloch, R. E., George, E. I., and Mcculloch, R. E. (1993). Variable selection via Gibbs sampling. *Journal of the American Statistical Association*, 88(423):881–889.

George, E. I., Sun, D., and Ni, S. (2008). Bayesian stochastic search for VAR model restrictions. *Journal of Econometrics*, 142(1):553–580.

Gladish, D. and Wikle, C. (2014). Physically motivated scale interaction parameterization in reduced rank quadratic nonlinear dynamic spatio-temporal models. *Environmetrics*, 25(4):230–244.

Goutelle, S., Maurin, M., Rougier, F., Barbaut, X., Bourguignon, L., Ducher, M., and Maire, P. (2008). The Hill equation: A review of its capabilities in pharmacological modelling. *Fundamental & Clinical Pharmacology*, 22(6):633–648.

Hastings, A. (1996). Models of spatial spread: Is the theory complete? *Ecology*, 77(6):1675–1679.

Hewitt, E. and Hewitt, R. E. (1979). The Gibbs-Wilbraham phenomenon: An episode in Fourier analysis. *Archive for History of Exact Sciences*, 21(2):129–160.

Higham, N. J., Dennis, M. R., Glendinning, P., Martin, P. A., Santosa, F., and Tanner, J. (2016). *The Princeton companion to applied mathematics*. Princeton University Press.

Hirsh, S. M., Barajas-Solano, D. A., and Kutz, J. N. (2021). Sparsifying priors for Bayesian uncertainty quantification in model discovery. *arXiv preprint arXiv:2107.02107*, pages 1–22.

Holmes, E. E., Lewis, M. A., Banks, J. E., and Veit, R. R. (1994). Partial differential equations in ecology: Spatial interactions and population dynamics. *Ecology*, 75(1):17–29.

Holton, J. R. and Hakim, G. J. (2012). *An introduction to dynamic meteorology*. Academic Press, 5 edition.

Hooten, M. B. and Wikle, C. K. (2008). A hierarchical Bayesian non-linear spatio-temporal model for the spread of invasive species with application to the Eurasian Collared-Dove. *Environmental and Ecological Statistics*, 15(1):59–70.

Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366.

Hsu, C.-P. F. and Wallace, J. M. (1976a). The global distribution of the annual and semiannual cycles in precipitation. *Monthly Weather Review*, 104(9):1093 – 1101.

Hsu, C.-P. F. and Wallace, J. M. (1976b). The global distribution of the annual and semi-annual cycles in sea level pressure. *Monthly Weather Review*, 104(12):1597–1601.

Huang, A. and Wand, M. P. (2013). Simple marginally noninformative prior distributions for covariance matrices. *Bayesian Analysis*, 8(2):439–452.

Huang, B., Thorne, P. W., Banzon, V. F., Boyer, T., Chepurin, G., Lawrimore, J. H., Menne, M. J., Smith, T. M., Vose, R. S., and Zhang, H.-M. (2017). Extended reconstructed sea surface temperature, version 5 (ERSSTv5): Upgrades, validations, and Intercomparisons. *Journal of Climate*, 30(20):8179–8205.

Icke, I. and Bongard, J. C. (2013). Improving genetic programming based symbolic regression using deterministic machine learning. In *2013 IEEE Congress on Evolutionary Computation*, pages 1763–1770. IEEE.

Jan van Oldenborgh, G., Balmaseda, M. A., Ferranti, L., Stockdale, T. N., and Anderson, D. L. T. (2005). Did the ECMWF seasonal forecast model outperform statistical ENSO forecast models over the last 15 years? *Journal of Climate*, 18(16):3240–3249.

Jin, Y., Fu, W., Kang, J., Guo, J., and Guo, J. (2019). Bayesian symbolic regression. *arXiv preprint arXiv:1910.08892*.

Katzfuss, M., Stroud, J. R., and Wikle, C. K. (2020). Ensemble Kalman methods for high-dimensional hierarchical dynamic space-time models. *Journal of the American Statistical Association*, 115(530):866–885.

Kermack, W. O. and McKendrick, A. G. (1927). A contribution to the mathematical theory of epidemics. *Proceedings of the Royal Society of London Series A*, 115(772):700–721.

Khoo, Y., Lu, J., and Ying, L. (2021). Solving parametric PDE problems with artificial neural networks. *European Journal of Applied Mathematics*, 32(3):421–435.

Kim, S., Lu, P. Y., Mukherjee, S., Gilbert, M., Jing, L., Ceperic, V., and Soljacic, M. (2021). Integration of neural network-based symbolic regression in deep learning for scientific discovery. *IEEE Transactions on Neural Networks and Learning Systems*, 32(9):4166–4177.

Knowles, I. and Renka, R. J. (2012). Methods for numerical differentiation of noisy data. *Electronic Journal of Differential Equations Conference*, 21(2012):235–246.

Kolda, T. (2006). Multilinear operators for higher-order decompositions. Technical report, Sandia National Laboratories (SNL), Albuquerque, NM, and Livermore, CA (United States).

Kolda, T. G. and Bader, B. W. (2009). Tensor decompositions and applications. *SIAM Review*, 51(3):455–500.

Korns, M. F. (2014). Extreme accuracy in symbolic regression. In *Genetic Programming Theory and Practice XI*, pages 1–30. Springer New York.

Koza, J. (1994). Genetic programming as a means for programming computers by natural selection. *Statistics and Computing*, 4(2):1–26.

Koza, J., Keane, M., and Rice, J. (1993). Performance improvement of machine learning via automatic discovery of facilitating functions as applied to a problem of symbolic system identification. In *IEEE International Conference on Neural Networks*, pages 191–198. IEEE.

Kraemer, B. M., Anneville, O., Chandra, S., Dix, M., Kuusisto, E., Livingstone, D. M., Rimmer, A., Schladow, S. G., Silow, E., Sitoki, L. M., Tamatamah, R., Vadeboncoeur, Y., and McIntyre, P. B. (2015). Morphometry and average temperature affect lake stratification responses to climate change. *Geophysical Research Letters*, 42(12):4981–4988.

Krebs, C. J., Boonstra, R., Boutin, S., and Sinclair, A. R. (2001). What drives the 10-year cycle of snowshoe hares? *BioScience*, 51(1):25–35.

Kühnert, D., Stadler, T., Vaughan, T. G., and Drummond, A. J. (2014). Simultaneous reconstruction of evolutionary history and epidemiological dynamics from viral sequences with the birth–death SIR model. *Journal of The Royal Society Interface*, 11(94):20131106.

Kuhnert, P. M. (2017). Physical-statistical modeling. In *Wiley StatsRef: Statistics Reference Online*, pages 1–5. Wiley.

Lagergren, J. H., Nardini, J. T., Michael Lavigne, G., Rutter, E. M., and Flores, K. B. (2020). Learning partial differential equations for biological transport models from noisy spatio-temporal data. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 476(2234):20190800.

Legendre, A. M. (1806). *Nouvelles méthodes pour la détermination des orbites des cometes*. F. Didot.

Li, Q. and Lin, N. (2010). The Bayesian elastic net. *Bayesian Analysis*, 5(1):151–170.

Liebhold, A., Koenig, W. D., and Bjørnstad, O. N. (2004). Spatial synchrony in population dynamics. *Annual Review of Ecology, Evolution, and Systematics*, 35(1):467–490.

Liu, H., Ye, Y., Wei, Y., Ma, W., Ma, M., and Zhang, K. (2019). Pattern formation in a reaction-diffusion predator-prey model with weak allee effect and delay. *Complexity*, 2019(1):1–14.

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. (2017). PDE-Net: Learning PDEs from data. *35th International Conference on Machine Learning, ICML 2018*, 7:5067–5078.

Lorenz, E. N. (1963). Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, 20(2):130–141.

Lotka, A. J. (1920). Analytical note on certain rhythmic relations in organic systems. *Proceedings of the National Academy of Sciences*, 6(7):410–415.

Mager, D. E., Wyska, E., and Jusko, W. J. (2003). Diversity of mechanism-based pharmacodynamic models. *Drug Metabolism and Disposition*, 31(5):510–518.

Malsiner-Walli, G. and Wagner, H. (2016). Comparing spike and slab priors for Bayesian variable selection. *Austrian Journal of Statistics*, 40(4).

Mandt, S., Hoffman, M., and Blei, D. (2016). A variational analysis of stochastic gradient algorithms. *Proceedings of The 33rd International Conference on Machine Learning*, 48:354–363.

Mangal, T. D., Paterson, S., and Fenton, A. (2008). Predicting the impact of long-term temperature changes on the epidemiology and control of Schistosomiasis: A mechanistic model. *PLoS ONE*, 3(1):e1438.

Mardt, A., Pasquali, L., Wu, H., and Noé, F. (2018). VAMPnets for deep learning of molecular kinetics. *Nature Communications*, 9(1):5.

Martius, G. and Lampert, C. H. (2016). Extrapolation and learning equations. *5th International Conference on Learning Representations, ICLR 2017 - Workshop Track Proceedings*, 1:1–13.

Maslyaev, M., Hvatov, A., and Kalyuzhnaya, A. (2019). Data-driven partial derivative equations discovery with evolutionary approach. In *Computational Science – ICCS 2019*, pages 635–641. Springer International Publishing.

Meier, L., Van De Geer, S., and Bühlmann, P. (2008). The group lasso for logistic regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(1):53–71.

Milliff, R. F., Bonazzi, A., Wikle, C. K., Pinardi, N., and Berliner, L. M. (2011). Ocean ensemble forecasting. Part I: Ensemble Mediterranean winds from a Bayesian hierarchical model. *Quarterly Journal of the Royal Meteorological Society*, 137(657):858–878.

Minnebo, W. and Stijven, S. (2011). *Empowering knowledge computing with variable selection - On variable importance and variable selection in regression random forests and symbolic regression*. PhD thesis, Antwerp University, Belgium.

Mitchell, T. J. and Beauchamp, J. J. (1988). Bayesian variable selection in linear regression. *Journal of the American Statistical Association*, 83(404):1023.

Nicolau, M. and Agapitos, A. (2018). On the effect of function set to the generalisation of symbolic regression models. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 272–273, New York, NY, USA. ACM.

Niven, R., Mohammad-Djafari, A., Cordier, L., Abel, M., and Quade, M. (2020). Bayesian identification of dynamical systems. *Proceedings*, 33(1):33.

North, J. S., Wikle, C. K., and Schliep, E. M. (2022). A Bayesian approach for data-driven dynamic equation discovery. *Journal of Agricultural, Biological, and Environmental Statistics*, 1(1):1–28.

Park, T. and Casella, G. (2008). The Bayesian lasso. *Journal of the American Statistical Association*, 103(482):681–686.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in PyTorch Adam. In *31st Conference on Neural Information Processing Systems (NIPS 2017)*, pages 1–4.

Philander, S. G. (1990). *El Niño, La Niña, and the Southern Oscillation*. Academic Press.

Piironen, J. and Vehtari, A. (2017). Sparsity information and regularization in the horseshoe and other shrinkage priors. *Electronic Journal of Statistics*, 11(2):5018–5051.

Qin, T., Wu, K., and Xiu, D. (2019). Data driven governing equations approximation using deep neural networks. *Journal of Computational Physics*, 395:620–635.

Rackauckas, C. and Nie, Q. (2017). DifferentialEquations.jl – A performant and feature-rich ecosystem for solving differential equations in Julia. *Journal of Open Research Software*, 5(1):15.

Raissi, M. (2018). Deep hidden physics models: Deep learning of nonlinear partial differential equations. *Journal of Machine Learning Research*, 19:1–24.

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. (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.

Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2017a). Physics informed deep learning (Part I): Data-driven solutions of nonlinear partial differential equations. *arXiv preprint arXiv:1711.10561*, Part I:1–22.

Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2017b). Physics informed deep learning (Part II): Data-driven discovery of nonlinear partial differential equations. *arXiv preprint arXiv:1711.10566*, Part II:1–19.

Raissi, M., Yazdani, A., and Karniadakis, G. E. (2020). Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. *Science*, 367(6481):1026–1030.

Ramsay, J. O. and Silverman, B. W. (2005). *Functional data analysis*. Springer Series in Statistics. Springer New York, New York, NY.

Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., and Prabhat (2019). Deep learning and process understanding for data-driven Earth system science. *Nature*, 566(7743):195–204.

Revels, J., Lubin, M., and Papamarkou, T. (2016). Forward-mode automatic differentiation in Julia. *arXiv preprint arXiv:1607.07892*.

Riley, W. J., Still, C. J., Torn, M. S., and Berry, J. A. (2002). A mechanistic model of $H_2^{18}O$ and $C_{18}OO$ fluxes between ecosystems and the atmosphere: Model description and sensitivity analyses. *Global Biogeochemical Cycles*, 16(4):42–1.

Ročková, V. and George, E. I. (2014). Negotiating multicollinearity with spike-and-slab priors. *METRON*, 72(2):217–229.

Royle, J. A., Berliner, L. M., Wikle, C. K., and Milliff, R. (1999). A Hierarchical spatial model for constructing wind fields from scatterometer data in the Labrador Sea. In *Case Studies in Bayesian Statistics.*, pages 367–382. Springer, New York, NY.

Rudy, S., Alla, A., Brunton, S. L., and Kutz, J. N. (2019a). Data-driven identification of parametric partial differential equations. *SIAM Journal on Applied Dynamical Systems*, 18(2):643–660.

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.

Rudy, S. H., Nathan Kutz, J., and Brunton, S. L. (2019b). Deep learning of dynamics and signal-noise decomposition with time-stepping constraints. *Journal of Computational Physics*, 396:483–506.

Sahoo, S. S., Lampert, C. H., and Martius, G. (2018). Learning equations for extrapolation and control. *35th International Conference on Machine Learning, ICML 2018*, 10:7053–7061.

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

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

Shumway, R. H. and Stoffer, D. S. (2017). *Time series analysis and its applications with R examples*. Springer, 4 edition.

Stine, A. R., Huybers, P., and Fung, I. Y. (2009). Changes in the phase of the annual cycle of surface temperature. *Nature*, 457(7228):435–440.

Stroud, J. R., Katzfuss, M., and Wikle, C. K. (2018). A Bayesian adaptive ensemble Kalman filter for sequential state and parameter estimation. *Monthly Weather Review*, 146(1):373–386.

Sun, Y., Zhang, L., and Schaeffer, H. (2019). NeuPDE: Neural network based ordinary and partial differential equations for modeling time-dependent data. *arXiv preprint arXiv:1908.03190*, 107(2016):352–372.

Sutton, R. T., Dong, B., and Gregory, J. M. (2007). Land/sea warming ratio in response to climate change: IPCC AR4 model results and comparison with observations. *Geophysical Research Letters*, 34(2):L02701.

Thiébaux, H. J., Mitchell, H. L., and Shantz, D. W. (1986). Horizontal structure of hemispheric forecast error correlations for geopotential and temperature. *Monthly Weather Review*, 114(6):1048–1066.

Thompson, D. W. J. and Wallace, J. M. (1998). The Arctic oscillation signature in the wintertime geopotential height and temperature fields. *Geophysical Research Letters*, 25(9):1297–1300.

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

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

Tsitouras, C. (2011). Runge–Kutta pairs of order 5(4) satisfying only the first column simplifying assumption. *Computers & Mathematics with Applications*, 62(2):770–775.

Tucker, L. R. (1966). Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311.

Usui, T., Butchart, S. H. M., and Phillimore, A. B. (2017). Temporal shifts and temperature sensitivity of avian spring migratory phenology: A phylogenetic meta-analysis. *Journal of Animal Ecology*, 86(2):250–261.

van Loon, H. (1967). The half-yearly oscillations in middle and high southern latitudes and the coreless winter. *Journal of the Atmospheric Sciences*, 24(5):472–486.

Wallace, J. M., Zhang, Y., and Lau, K.-H. (1993). Structure and seasonality of interannual and interdecadal variability of the geopotential height and temperature fields in the northern hemisphere troposphere. *Journal of Climate*, 6(11):2063–2082.

Wang, J. L., Chiou, J. M., and Müller, H. G. (2016). Functional data analysis. *Annual Review of Statistics and Its Application*, 3:257–295.

Wang, Y., Wagner, N., and Rondinelli, J. M. (2019). Symbolic regression in materials science. *MRS Communications*, 9(3):793–805.

West, M. and Harrison, J. (2006). *Bayesian forecasting and dynamic models*. Springer Science & Business Media.

White, F. M. and Majdalani, J. (2006). *Viscous fluid flow*. McGraw-Hill New York, 3 edition.

White, G. H. and Wallace, J. M. (1978). The global distribution of the annual and semiannual cycles in surface temperature. *Monthly Weather Review*, 106(6):901–906.

Wikle, C. K. (2003). Hierarchical Bayesian models for predicting the spread of ecological processes. *Ecology*, 84(6):1382–1394.

Wikle, C. K., Berliner, L. M., and Cressie, N. (1998). Hierarchical Bayesian space-time models. *Environmental and Ecological Statistics*, 5(2):117–154.

Wikle, C. K. and Chen, T.-C. (1996). On the semiannual variation in the northern hemisphere extratropical height field. *Journal of Climate*, 9:2250 – 2258.

Wikle, C. K. and Holan, S. H. (2011). Polynomial nonlinear spatio-temporal integro-difference equation models. *Journal of Time Series Analysis*, 32(4):339–350.

Wikle, C. K. and Hooten, M. B. (2010). A general science-based framework for dynamical spatio-temporal models. *TEST*, 19(3):417–451.

Wikle, C. K., Milliff, R. F., Nychka, D., and Berliner, L. M. (2001). Spatiotemporal hierarchical Bayesian modeling: Tropical ocean surface winds. *Journal of the American Statistical Association*, 96(454):382–397.

Willis, M.-J. (1997). Genetic programming: an introduction and survey of applications. In *Second International Conference on Genetic Algorithms in Engineering Systems*, pages 314–319. IET.

Wu, K. and Xiu, D. (2019). Numerical aspects for approximating governing equations using data. *Journal of Computational Physics*, 384:200–221.

Wu, K. and Xiu, D. (2020). Data-driven deep learning of partial differential equations in modal space. *Journal of Computational Physics*, 408:109307.

Xu, H., Chang, H., and Zhang, D. (2019). DL-PDE: Deep-learning based data-driven discovery of partial differential equations from discrete and noisy data. *Communications in Computational Physics*, 29(3):698–728.

Xu, H., Chang, H., and Zhang, D. (2020). DLGA-PDE: Discovery of PDEs with incomplete candidate library via combination of deep learning and genetic algorithm. *Journal of Computational Physics*, 418:109584.

Xu, H., Zhang, D., and Zeng, J. (2021). Deep-learning of parametric partial differential equations from sparse and noisy data. *Physics of Fluids*, 33(3):037132.

Yang, H.-C., Hu, G., and Chen, M.-H. (2019). Bayesian variable selection for pareto regression models with latent multivariate log gamma process with applications to earthquake magnitudes. *Geosciences*, 9(4):169.

Yang, Y., Aziz Bhouri, M., and Perdikaris, P. (2020). Bayesian differential programming for robust systems identification under uncertainty. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 476(2243):20200290.

Zammit-Mangion, A., Ng, T. L. J., Vu, Q., and Filippone, M. (2021). Deep compositional spatial models. *Journal of the American Statistical Association*, 0(0):1–22.

Zellner, A. (1986). On assessing prior distributions and Bayesian regression analysis with g-prior distributions. *Bayesian Inference and Decision Techniques*, pages 233–243.

Zeng, N., Dickinson, R. E., and Zeng, X. (1996). Climatic impact of amazon deforestation — A mechanistic model study. *Journal of Climate*, 9(4):859–883.

Zhang, L. and Schaeffer, H. (2019). On the convergence of the SINDy algorithm. *Multiscale Modeling & Simulation*, 17(3):948–972.

Zhang, Q., Perra, N., Perrotta, D., Tizzoni, M., Paolotti, D., and Vespignani, A. (2017). Forecasting seasonal influenza fusing digital indicators and a mechanistic disease model. In *Proceedings of the 26th International Conference on World Wide Web*, pages 311–319, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

Zhang, S. and Lin, G. (2018). Robust data-driven discovery of governing physical laws with error bars. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 474(2217):20180305.

Zhang, Z., Tao, Y., and Li, Z. (2007). Factors affecting hare-lynx dynamics in the classic time series of the Hudson Bay Company, Canada. *Climate Research*, 34(2):83–89.

Zheng, P., Askham, T., Brunton, S. L., Kutz, J. N., and Aravkin, A. Y. (2019). A unified framework for sparse relaxed regularized regression: SR3. *IEEE Access*, 7:1404–1423.

VITA

Joshua Snowball North was born November 25, 1993 in Cambridge, Massachusetts to Monica Lynn and John North. He was raised in Albuquerque, New Mexico and graduated from Eldorado High School in 2012. He earned a B.A. in Ecology and Evolutionary Biology, a B.S. in Applied Mathematics, and a minor in Statistics from the University of Colorado Boulder in 2017. He began a doctoral degree in Statistics at the University of Missouri Columbia under Drs. Christopher K. Wikle and Erin M. Schliep. He completed a M.A. in Statistics in 2019 and a Ph.D. in Statistics in 2022. He has accepted a postdoctoral research position at the Lawrence Berkeley National Laboratory working with Dr. Mark Risser.