DYNAMIC ACTIVE SUBSPACES: A DATA-DRIVEN APPROACH TO COMPUTING TIME-DEPENDENT ACTIVE SUBSPACES IN DYNAMICAL SYSTEMS

by

IZABEL PIRIMAI AGUIAR

B.S., Colorado School of Mines, 2017

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Master of Science

Department of Computer Science

2018

This thesis entitled:
DYNAMIC ACTIVE SUBSPACES: A DATA-DRIVEN APPROACH TO COMPUTING TIME-DEPENDENT ACTIVE SUBSPACES IN DYNAMICAL SYSTEMS
written by Izabel Pirimai Aguiar
has been approved for the Department of Computer Science

_____
Dr. Paul Constantine

_____
Dr. Elizabeth Bradley

_____
Dr. Jim Curry

Date _____

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Aguiar, Izabel Pirimai (M.S. Computer Science)

Dynamic active subspaces: a data-driven approach to computing time-dependent active subspaces in dynamical systems

Thesis directed by Dr. Paul Constantine

Computational models are aiding in the advancement of science – from biological, to engineering, to social systems. To trust the predictions of computational models, however, we must understand how the errors in the models' inputs (i.e., through measurement error) affect the output of the systems: we must quantify the uncertainty that results from these input errors. Uncertainty quantification (UQ) becomes computationally complex when there are many parameters in the model. In such cases it is useful to reduce the dimension of the problem by identifying unimportant parameters and disregarding them for UQ studies. This makes an otherwise intractable UQ problem tractable. *Active subspaces* extend this idea by identifying important linear combinations of parameters, enabling more powerful and effective dimension reduction. Although active subspaces give model insight and computational tractability for scalar-valued functions, it is not enough. This analysis does not extend to time-dependent systems. In this thesis we discuss time-dependent, dynamic active subspaces. We develop a methodology by which to compute and approximate dynamic active subspaces, and introduce the analytical form of dynamic active subspaces for two cases. To highlight these methods we find dynamic active subspaces for a linear harmonic oscillator and a nonlinear enzyme kinetics system.

Dedication

For Mum and Dad.

Acknowledgements

I would primarily like to thank my advisor, Paul Constantine. Paul has shown me, through the opportunities and support he's given me, how much he believes in my ability to succeed. In doing so, he has given me some of the most positive and productive experiences I've ever had. He has continually emboldened me to continue on my own path. Professionally, no one else has impacted and influenced me as much as he has. I am extremely grateful for the advocacy he has given me.

I would also like to sincerely thank the members of my research group: Zachary Grey, Andrew Glaws, and Jeffrey Hokanson, for their engaging discussions, cups of coffee, critical feedback, and overwhelming kindness. Many thanks to my thesis committee, Elizabeth Bradley, Gianluca Iaccarino, and Jim Curry for their support as well as to Nathan Kutz, Steven Brunton, John Butcher, Kyle Niemeyer, and Peter Schmid, for their interest and encouragement. I am also extremely grateful for the dozens of kind and enthusiastic conference attendees who have inspired new questions and driven further research. I am excited and grateful to be a part of such a community.

Certainly not least, I am thankful to my family: Aristotle Johns, Jojo Clark, Sharon Aguiar, Laura Aguiar, Matt Aguilar, Christina Whippen, Amanda Evans, Julie Clark, and Xenia Johns. I would be nowhere without their unparalleled support, and I would know nothing without their boundless Love.

CONTENTS

CHAPTER

TABLES

Table

FIGURES

Figure

CHAPTER 1

1.1 INTRODUCTION

Computational models are aiding in the advancement of science – from biological, to engineering, to social systems. To trust the predictions of computational models, however, we must understand how the errors in the models' inputs (i.e., through measurement error) affect the output of the systems: we must quantify the uncertainty that results from these input errors. Uncertainty quantification (UQ) becomes computationally expensive when there are many parameters in the model. In such cases it is useful to reduce the dimension of the model by identifying unimportant parameters and disregarding them for UQ studies. This makes an otherwise intractable UQ problem tractable. For time-dependent systems, these analyses are limited to local metrics that do not fully explore the parameter space, or global metrics that are either limited to a static point in time or to the sensitivity of individual parameters.

Active subspaces [11] identify linear combinations of the parameters that change the quantity of interest the most on average. The active subspace of a system can be exploited to identify coordinate-based global sensitivity metrics called *activity scores* [13] that provide comparable sensitivity analysis to existing metrics. The existing approach for extending active subspaces to time-dependent systems involves recomputing the active subspace at each time step of interest [33, 14]. This analysis requires sampling across the parameter space, approximating gradients, approximating an expectation, and eigendecomposing a matrix at

each time step. For complex time-dependent systems or for those with many parameters this can become computationally expensive.

Extending active subspaces to time-dependent systems will enable uncertainty quantification, sensitivity analysis, and parameter estimation for computational models that have explicit dependence on time. Understanding how *linear combinations* of the parameters change the quantity of interest *over time* could inform important discipline-specific insights. Such analysis would identify which linear combinations change the quantity of interest the most on average, and how those linear combinations change in time.

The contributions of this work. In this thesis we present a novel approach for uncertainty quantification and sensitivity analysis in time-dependent systems. We prove the existence of a dynamical system describing the dynamic active subspace in the cases of three linear dynamical systems.

For other time-dependent quantities of interest, we present and develop a methodology for approximating the dynamic active subspace with "data" in the form of the one-dimensional active subspace of a system at different points in time. We approximate the dynamic active subspace using this methodology for (i) a linear harmonic oscillator and (ii) the enzyme kinetics system. Our methodology provides a means by which to identify the linear combination of parameters that will change the function the most on average, and how the linear combination changes in time.

The concept of using data-driven recovery techniques to approximate a time-dependent sensitivity metric given snapshots of the sensitivity metric is, to the best of our knowledge, novel. Furthermore, the methodology presented in this work can be extended to develop time-dependent insight for other global sensitivity metrics. This intellectual and technical contribution of the novel ideas and methodology developed in this work lend to the advancement of uncertainty quantification and sensitivity analysis for time-dependent functions, with applications ranging from epidemiological studies, to chemical kinetics, to biological systems.

The remainder of this thesis proceeds as follows. We begin by reviewing existing global sensitivity analysis tools and related work and follow by defining the notation and vocabulary used in the paper. In Section 1.4 we detail the background information on active subspaces, dynamic mode decomposition (DMD), and sparse identification for nonlinear dynamical systems (SINDy). In Section 2.2 we prove the existence of the analytic form of three dynamic active subspaces. In Section 2.3 we propose the method of approximating dynamic active subspaces with DMD and SINDy. We discuss the benefits and drawbacks of such methods and conduct numerical experiments on the dynamic active subspace for an output of a linear and nonlinear dynamical system. We finish by drawing conclusions on the preceding work and detailing directions for future work.

A note to the reader The pronoun *we* will be used from the perspective of the authors. This choice is intended to directly include the reader in the instruction, analysis, and discussion of the following ideas.

1.2 EXISTING TOOLS AND RELATED WORK

Sensitivity analysis methods can be separated into local and global metrics. As discussed in [34], local sensitivity analysis is appropriate when the parameters of a model are known with little uncertainty. In such cases we can evaluate the partial derivative of the quantity of interest (QoI) with respect to its parameters to evaluate the relative change in the QoI with respect to each parameter. However, in many biological and engineering systems the input parameters are associated with a range of uncertainty or possibility; e.g. glycosis modeling [46], molecular systems [43, 3], battery chemistries [37], and HIV modeling [38]. In such cases a **global sensitivity metric** is necessary to understand the sensitivity of the system to its range of inputs. There are many existing global sensitivity analysis tools, including "One-At-A-Time" screening methods [23] such as the Morris method [36], the full factorial method [18], the analysis of variance (ANOVA) decomposition [29], Fourier Amplitude Sensitivity Test (FAST) [17], and active subspaces [13]. For a more detailed review of

global sensitivity analysis see [23] and [41].

Local and global metrics exist for sensitivity analysis of dynamical systems, as well. For a system of ordinary differential equations (ODEs), time-dependent local sensitivity metrics are given by the augmented ODE with additional terms for the partials of the states with respect to each parameter [21]. Similarly, [47] and [7] develop a time-dependent local sensitivity metric for evaluating the parameter dependence of dynamic population models. These metrics, along with others reviewed in [51], are local and only assess small perturbations from the nominal parameter values. Global sensitivity analyses of dynamical systems are often limited to analyzing an aspect of the system that is not time-dependent, such as the steady state [1] or known stages [28] of the system. For time-dependent analyses, the Fourier Amplitude Sensitivity Test (FAST) computes the time dependent variances of each state of a system. In [34], the Pearson Rank Correlation Coefficient [39] is computed at multiple points in time. These time-dependent analyses are limited to assessing the variance in or correlation between the QoI and each parameter in the system. In the case of a high-dimensional parameter space, these individual variances become difficult to assess.

1.3 NOTATION AND VOCABULARY

In this section we define the notation and vocabulary used throughout the remainder of the paper to create a means by which definitions and notations can easily be referenced. Although we define these terms below, they will also be explicitly detailed in the following sections.

$\mathbf{p} \in \mathbb{R}^m$	The m real-valued parameters of a scalar-valued quantity of interest. The upper and lower bounds of these parameters are given by $\mathbf{p}_\ell \in \mathbb{R}^m$ and $\mathbf{p}_u \in \mathbb{R}^m$, respectively.
$f(t, \mathbf{p})$	$f : \mathbb{R}^m \to \mathbb{R}$ The real-valued scalar quantity of interest dependent on m parameters $\mathbf{p}$ and time. In the discussions of *static* active subspaces, this quantity of interest will not be dependent on time.
$\rho(\mathbf{p})$	The joint probability density for the parameter space of f. For example, in the below computations we assume that the parameters are distributed uniformly between $[\mathbf{p}_\ell, \mathbf{p}_u]$.
$\nabla f \in \mathbb{R}^m$	The gradient of $f(\mathbf{p}, t)$ with respect to parameters $\mathbf{p}$.
$\boldsymbol{C} \in \mathbb{R}^{m \times m}$	The matrix given by $\int \nabla f \nabla f^T \rho(\mathbf{p}) d\mathbf{p}$.
$\boldsymbol{W}\boldsymbol{\Lambda}\boldsymbol{W}^T$	The eigendecomposition of $\boldsymbol{C}$ with eigenvectors $\mathbf{w}_i$ and eigenvalues λ_i for $i = 1, 2, \ldots, m$.
$\boldsymbol{W}_1 \in \mathbb{R}^{m \times k}$	The k first eigenvectors of $\boldsymbol{C}$ corresponding to the k *largest* eigenvalues. This is also referred to as *the active subspace* of $f(t, \mathbf{p})$.
DMD	Shorthand for *dynamic mode decomposition* and broadly used to refer to the method presented in [44] for approximating dynamics of dynamical systems from data.
$\mathbf{u} \in \mathbb{R}^q$	The q real-valued states of a dynamical system.
$\dot{\mathbf{u}}$	The q real-valued time derivatives of the states of a dynamical system.
$g(\mathbf{u})$	$g : \mathbb{R}^q \to \mathbb{R}^q$ The governing equations describing the time dependence of the states of the dynamical system
$\boldsymbol{X}_{\boldsymbol{DMD}} \in \mathbb{R}^{q \times (n-1)}$	Data matrix used in DMD consisting of observations of $\mathbf{u} \in \mathbb{R}^q$ from time 0 to time $n - 1$. Note that in this data matrix, the observations are columns.

$\boldsymbol{X'_{DMD}} \in \mathbb{R}^{q\times(n-1)}$	Data matrix used in DMD consisting of observations of $\mathbf{u}$ from time 1 to time n. Note that in this data matrix, the observations are columns.
SINDy	Shorthand for *sparse identification for nonlinear dynamical systems* and broadly used to refer to the method presented in [4] for recovering the governing system of dynamical systems from data.
$\boldsymbol{X_{SINDy}} \in \mathbb{R}^{n\times q}$	Data matrix used in SINDy consisting of observations of $\mathbf{u} \in \mathbb{R}^q$ at n points in time. Note that in this data matrix, the observations are rows.
$\boldsymbol{\dot{X}_{SINDy}} \in \mathbb{R}^{n\times q}$	Data matrix used in SINDy consisting of time derivatives of $\mathbf{u} \in \mathbb{R}^q$ at n points in time.
$\boldsymbol{\Theta} \in \mathbb{R}^{n\times p}$	Library matrix used in SINDy consisting of p transformations of the columns of $\boldsymbol{X_{SINDy}} \in \mathbb{R}^{n\times q}$ according to *candidate functions* for the governing dynamical system.
$\boldsymbol{X}^{P_n}$	An nth order monomial of the columns of $\boldsymbol{X_{SINDy}}$. For example, for a system of three states, $\mathbf{u} \in \mathbb{R}^3 = [x\ y\ z]^T$, $\boldsymbol{X}^{P_2} = [x^2\ xy\ xz\ y^2\ yz\ z^2]$.
$\boldsymbol{\Xi} \in \mathbb{R}^{p\times q}$	Coefficient matrix used in SINDy defining the contributing weight of functions in $\boldsymbol{\Theta}$ to the governing system $g(\mathbf{u})$.
DyAS	Shorthand for *dynamic active subspaces* and broadly used to refer to the active subspace of a time-dependent quantity of interest.
$\mathbf{w}(t) \in \mathbb{R}^m$	Used to refer to the one-dimensional dynamic active subspace of $f(t, \mathbf{p})$, corresponding to the first eigenvector of matrix $\boldsymbol{C}$ in time.

1.4 BACKGROUND

1.4.1 ACTIVE SUBSPACES

Consider a function $f : \mathbb{R}^m \to \mathbb{R}$ that maps the inputs of a system, $\mathbf{p} \in \mathbb{R}^m$, to a scalar output of interest (e.g., cell count, drug concentration, temperature). We assume that the parameters $\mathbf{p}$ have joint probability density $\rho(\mathbf{p}) : \mathbb{R}^m \mapsto \mathbb{R}^+$. This density represents the uncertainty in the parameters. For example, if it is equally probable that the parameters take any value in the range of its lower and upper bounds, $\mathbf{p}_\ell$ and $\mathbf{p}_u$, respectively, then $\mathbf{p} \sim \mathcal{U}(\mathbf{p}_\ell, \mathbf{p}_u)$.

We assume that f is continuously differentiable almost everywhere in the support of ρ, and has square integrable derivatives. *Active subspaces* [11] are eigenspaces of the matrix,

$$\boldsymbol{C} = \int \nabla f(\mathbf{p}) \nabla f(\mathbf{p})^T \rho(\mathbf{p}) d\mathbf{p} = \boldsymbol{W} \boldsymbol{\Lambda} \boldsymbol{W}^T. \tag{1.1}$$

The matrix $\boldsymbol{C}$ is the expectation of the outer product of the gradient of f, and is symmetric and positive semidefinite. Thus the eigendecomposition of $\boldsymbol{C}$ yields real non-negative eigenvalues, $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m$, whose relationship to the function f is described by the following lemma.

Lemma 1.4.1 (Lemma 3.1, Constantine, 2015 [11])**.** *The mean-squared directional derivative of f with respect to the eigenvector $\mathbf{w}_i$ is equal to the corresponding eigenvalue,*

$$\int \left((\nabla f)^T \mathbf{w}_i \right)^2 \rho(\mathbf{p}) d\mathbf{p} = \lambda_i \tag{1.2}$$

Assuming the largest eigenvalue λ_1 is unique, it indicates that $\mathbf{w}_1$ is the direction in the parameter space along which f, changes the *most* on average. Let the eigendecomposition be partitioned by

$$\boldsymbol{\Lambda} = \begin{bmatrix} \boldsymbol{\Lambda}_1 & \\ & \boldsymbol{\Lambda}_2 \end{bmatrix}, \quad \boldsymbol{W} = \begin{bmatrix} \boldsymbol{W}_1 & \boldsymbol{W}_2 \end{bmatrix}, \tag{1.3}$$

where $\mathbf{\Lambda}_1$ contains the $k < m$ largest eigenvalues and $\boldsymbol{W}_1$ the corresponding k eigenvectors. We define new parameters $\mathbf{y}$ and $\mathbf{z}$

$$\mathbf{y} = \boldsymbol{W}_1^T \mathbf{p} \in \mathbb{R}^k, \quad \mathbf{z} = \boldsymbol{W}_2^T \mathbf{p} \in \mathbb{R}^{m-k}. \tag{1.4}$$

The new parameters $\mathbf{y}$ and $\mathbf{z}$ are linear combinations of the original parameters $\mathbf{p}$. The parameters $\mathbf{y}$ are those that are *most important* in the following sense.

Lemma 1.4.2 (Constantine, 2015)**.** *The mean-squared gradients of f with respect to $\mathbf{y}$ and $\mathbf{z}$ satisfy*

$$\int (\nabla_{\mathbf{y}} f)^T (\nabla_{\mathbf{y}} f) \rho(\mathbf{p}) d\mathbf{p} = \lambda_1 + \ldots + \lambda_k \tag{1.5}$$

$$\int (\nabla_{\mathbf{z}} f)^T (\nabla_{\mathbf{z}} f) \rho(\mathbf{p}) d\mathbf{p} = \lambda_{k+1} + \ldots + \lambda_m \tag{1.6}$$

Thus the parameters defined by $\mathbf{y}$ are those for which the mean squared gradients of f are largest in the 2-norm. Consider, for example, the case where $\lambda_{k+1} = \ldots = \lambda_m = 0$. In this case, the only directions in which the function changes are those defined by $\boldsymbol{W}_1$.

Although finding the active subspace and principal component analysis (PCA) [24] both involve the process of eigendecomposing a matrix, it is important to note that the two analyses are *not* the same. The matrix to be eigendecomposed in PCA is the covariance of the parameters $\mathbf{p}$, whereas the matrix used to find the active subspace is that defined by (1.1). The most critical distinction between these two analyses is that PCA gives insight into the relationship between *one parameter and another*; active subspaces give insight into the relationship between *the parameters and the function's output.*

1.4.1.1 Practicalities

In this section we discuss the practicalities of finding and estimating the active subspace. We address the convention of centering and normalizing the parameters $\mathbf{p}$ and when to take the log-transform of the parameters. Additionally, we discuss the methods by which

we estimate (i) the gradient of f with respect to $\mathbf{p}$ and (ii) the integral (1.1) defining the active subspace.

Normalizing the parameters In the analysis we assume that $\mathbf{p} \in [-1,1]^m$. To satisfy this assumption in practice we often need to consider an initial map that shifts and scales the model's natural parameter space to the normalized space. For parameters $\mathbf{p} \in \mathbb{R}^m$ with upper and lower bounds $\mathbf{p}_u, \mathbf{p}_\ell$ respectively, the appropriate shifted and scaled parameters are given by

$$\mathbf{p}^* = \frac{\text{diag}(\mathbf{p}_u - \mathbf{p}_l)\mathbf{p} + (\mathbf{p}_u + \mathbf{p}_\ell)}{2}, \quad \mathbf{p}^* \in [-1,1]^m. \tag{1.7}$$

Log-transform of parameter space When the inputs of f have units, as in the enzyme-kinetics system we study in Section 2.3.3, we take a log-transform of the parameter space. Such a decision is made from experience and insight with other dimensionalized systems. There is an interesting connection between the classical process of finding non-dimensional parameters for a physical system and identifying important linear combinations of parameters. Since the non-dimensional parameters are products of powers of the original parameters, a log-transform produces a linear combination of the logs of the original parameters. This implies that, for systems where the original parameters are amenable to non-dimensionalization, the log-transformed system depends only on a few linear combinations. Such structure is closely related to active subspaces; see [12] for a detailed discussion.

Estimating gradients In practice we do not have the analytic form of $\nabla f(\mathbf{p})$ and thus must approximate it. If this is the case one may approximate the gradient using first order forward finite differences,

$$\frac{\partial f}{\partial p_i} \approx \frac{f(p_i + h) - f(p_i)}{h}, \;\; i = 1, 2, ..., m. \tag{1.8}$$

For dynamical systems with an analytical form $\nabla f(t, \mathbf{p})$ can be approximated by augmenting the original ODE system with the time derivative of the states' partial derivatives. These

will be referred to as *Gronwall Gradients* [21] and are detailed in Section 1.4.2.

Estimating the expectation The expectation defined by the integral in (1.1) must be approximated. For this paper we employ two methods for approximating this integral using (i) Monte Carlo and (ii) Gauss-Legendre quadrature. Both approximation rules are achieved by the following algorithm.

Algorithm 1 Estimation of the active subspace.

Given M quadrature weights $\omega_i \in \mathbb{R}$ and points $\mathbf{p}_i \in \mathbb{R}^m$:

(1) For each point $\mathbf{p}_i$ compute the quantity of interest $f(\mathbf{p}_i)$

(2) Estimate the gradient $\nabla_{\mathbf{p}} f_i = \nabla_{\mathbf{p}} f(\mathbf{p}_i)$.

(3) Compute the matrix $\hat{\boldsymbol{C}}$ and its eigenvalue decomposition,

$$\hat{\boldsymbol{C}} = \sum_{i=1}^{M} \omega_i \nabla_{\mathbf{p}} f_i \nabla_{\mathbf{p}} f_i^T = \hat{\boldsymbol{W}} \hat{\boldsymbol{\Lambda}} \hat{\boldsymbol{W}} \tag{1.9}$$

The choice of M depends on f and the dimension of the parameter space. With Monte Carlo sampling, the points $\mathbf{p}_i$ are randomly drawn according to density $\rho(\mathbf{p})$ with weights $\omega_i = \frac{1}{M}$. The estimate for $\hat{\boldsymbol{C}}$ converges to $\boldsymbol{C}$ like $\mathcal{O}(M^{-1/2})$. When the dimension m of f is low, and when f has sufficiently smooth derivatives we can achieve a much better approximation using a Gauss quadrature rule to approximate $\boldsymbol{C}$. Here, M nodes $\mathbf{p}_i$ and weights ω_i are drawn *per dimension* according to the tensor product Gauss-Legendre quadrature scheme. It is important to note that the number of nodes M per dimension m leads to M^m total function evaluations and gradient approximations. The choice of M again depends on f and in practice one should conduct numerical convergence studies to find an appropriate M. We implement and focus on these two methods for approximating the active subspace, although additional approaches can be found in [11].

1.4.2 GRONWALL GRADIENTS

Assume that the q states of the dynamical system (1.12) are dependent upon at least one parameter p. In [21], Gronwall derives a system of ordinary differential equations that includes the dynamics of both the original state variables and of their partial derivatives, $\frac{\partial u_i}{\partial p}$. Numerically integrating the system provides an approximation for the partial derivatives of the state with respect to a parameter, p as a function of time. In the case where there is more than one parameter, the system is easily extended to include the dynamics of these partial derivatives. This approach for approximating partial derivatives was developed as a local sensitivity metric, but can be used as an efficient way to approximate the gradient of a function.

1.4.2.1 Description of variables

Let the following system of q differential equations describe the dynamics of the states u_i over time for $i = 1, \ldots, q$ with parameters p_k for $k = 1, \ldots, m$:

$$\frac{du_i}{dt} = g_i(t; u_1, \ldots, u_q; p_1, \ldots p_m). \tag{1.10}$$

Let the initial conditions $u_i = u_i^0$ for $t = t_0$ and assume that for $j = 1, \ldots, q$ the partials $\frac{\partial g_i}{\partial u_j}$ and $\frac{\partial g_i}{\partial p_k}$ are continuous [21].

1.4.2.2 Extended dynamical system with partial derivatives

The partial derivative of state u_i with respect to parameter p_k is written $\frac{\partial u_i}{\partial p_k}$. The differential equation that describes the dynamics of these partials is as follows.

$$\frac{d}{dt}\left(\frac{\partial u_i}{\partial p_k}\right) = \sum_j \frac{\partial g_i}{\partial u_j}(t; u_1, \ldots, u_q; p_1, \ldots p_m)\frac{\partial u_j}{\partial p_k} + \frac{\partial g_i}{\partial p_k}(t; u_1, \ldots, u_q; p_1, \ldots p_m) \tag{1.11}$$

As defined, the initial conditions for all of the partial derivatives are zero. The dynamical system resulting from augmenting (1.10) with (1.11) has $q+(m\times q)$ states. The integration of this system will result in the solutions of the states in time and the solutions of the partials

in time. For our purposes we approximate the integration of the system using python's `scipy.integrate.odeint` function with parameter values defined by the integration nodes used to approximate $\boldsymbol{C}$.

1.5 DATA-DRIVEN RECOVERY OF DYNAMICAL SYSTEMS

A parameterized dynamical system is a system in which the time dependence of the *states* is described by governing equations dependent on states and parameters. The vector form of (1.10) is written,

$$\dot{\mathbf{u}} = g(\mathbf{u}, \mathbf{p}), \quad \mathbf{u} \in \mathbb{R}^q, \ \mathbf{p} \in \mathbb{R}^m, \quad g : \mathbb{R}^{q+m} \longrightarrow \mathbb{R}^q. \tag{1.12}$$

In this form, $\dot{\mathbf{u}}$ denotes the time derivative of the states $\mathbf{u}$, and g denotes a continuous function of the states describing the governing time-dependence. The governing equations of dynamical systems enables in-depth analysis and computational simulations of the physical systems they model. However, in many systems the governing equations are not known. To address such situations, methods exist to recover the underlying dynamical system and predict future states from observations of the system with a local linear approximation [44]; recovering governing equations by solving a constrained optimization problem [49]; recovering governing equations through a sparse least squares problem [4]; nonlinear regression analysis [52]; iterative symbolic regression [45]; nonlinear adaptive projection onto a sparse basis [42]; and with embedding dimension and global equations of motion analysis [16]. Below we present the methods of: (i) *dynamic mode decomposition* (DMD) developed by Schmid [44] and further analyzed by Kutz, et al. in [30]; and (ii) *sparse identification of nonlinear dynamical systems* (SINDy) developed by Brunton et al. in [4]. We review these data-driven recovery methods to motivate their use for recovering and approximating the dynamics of time-dependent active subspaces.

1.5.1 DYNAMIC MODE DECOMPOSITION

Assume that we can observe a dynamic process with q state variables from which we can collect n snapshots of data. We further assume that the dynamical system (1.12) has corresponding discrete-time map given by

$$\mathbf{u}_{k+1} = \mathbf{F}(\mathbf{u}_k). \tag{1.13}$$

for $k = 1, 2, 3, \ldots n-1$ time steps. DMD approximates this discrete-time map by

$$\mathbf{u}_{k+1} \approx \boldsymbol{A}\mathbf{u}_k \tag{1.14}$$

where $\boldsymbol{A}$ minimizes $||\mathbf{u}_{k+1} - \boldsymbol{A}\mathbf{u}_k||_2$ over the data snapshots. Although DMD can be used to provide insight into the dynamical system, we implement the algorithm to exploit only its ability to estimate states in between data snapshots and predict future states using the approximate discrete-time map given by (1.14).

The definition of the dynamic mode decomposition as given by [50] is:

Suppose we have a dynamical system (1.12) *and two sets of data,*

$$\boldsymbol{X} = \begin{bmatrix} | & | & & | \\ \mathbf{u}(t_1) & \mathbf{u}(t_2) & \ldots & \mathbf{u}(t_{n-1}) \\ | & | & & | \end{bmatrix}, \tag{1.15}$$

$$\boldsymbol{X'} = \begin{bmatrix} | & | & & | \\ \mathbf{u}'(t_2) & \mathbf{u}'(t_3) & \ldots & \mathbf{u}'(t_n) \\ | & | & & | \end{bmatrix} \tag{1.16}$$

so that $\mathbf{u}'_k = \mathbf{F}(\mathbf{u}_k)$, *where* $\mathbf{F}$ *is the map in* (1.13) *corresponding to the evolution of* (1.12) *for time* Δt. *DMD computes the leading eigendecomposition of the best-fit linear operator* $\boldsymbol{A}$ *relating the data* $\boldsymbol{X'} \approx \boldsymbol{AX}$:

$$\boldsymbol{A} = \boldsymbol{X'}\boldsymbol{X}^{\dagger}. \tag{1.17}$$

The DMD nodes, also called dynamic modes, are the eigenvectors of $\boldsymbol{A}$, *and each DMD node corresponds to a particular eigenvalue of* $\boldsymbol{A}$.

1.5.1.1 The method

We implement a simple approach to the DMD algorithm presented in [30]. The algorithm as-is in [30] consists of using a low-rank projection of $\boldsymbol{A}$ onto its proper orthogonal

decomposition (POD) modes. For our application of DMD we want the full-dimensional representation of the dynamics and thus avoid this step, however it should be noted that for other applications Algorithm 1.1 in [30] can be used exactly.

Algorithm 2 Dynamic mode decomposition

Given matrices of snapshots $\boldsymbol{X}$ and $\boldsymbol{X'}$:

(1) Take the singular value decomposition (SVD) of $\boldsymbol{X}$:

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\boldsymbol{T}}, \tag{1.18}$$

(2) Construct matrix $\boldsymbol{A}$ from (1.17) with the pseudoinverse of $\boldsymbol{X}$ given by the above SVD:

$$\boldsymbol{X}^{\dagger} = \boldsymbol{V}\boldsymbol{\Sigma}^{-1}\boldsymbol{U}^{T}, \tag{1.19}$$

$$\boldsymbol{A} = \boldsymbol{X'}\boldsymbol{V}\boldsymbol{\Sigma}^{-1}\boldsymbol{U}^{T}. \tag{1.20}$$

(3) Compute the eigendecomposition of $\boldsymbol{A}$:

$$\boldsymbol{A}\boldsymbol{W} = \boldsymbol{W}\boldsymbol{\Lambda}, \tag{1.21}$$

where the eigenvectors of $\boldsymbol{A}$ are the columns of $\boldsymbol{W}$ and the eigenvalues λ_i are the diagonal entries of $\boldsymbol{\Lambda}$.

(4) Construct

$$\boldsymbol{\Phi} = \boldsymbol{X'}\boldsymbol{V}\boldsymbol{\Sigma}^{-1}\boldsymbol{W}. \tag{1.22}$$

(5) The future states of the system are approximated by

$$\mathbf{x}(t) \approx \Sigma_{k=1}^{r}\phi_k \exp{(\omega_k t)}b_k = \boldsymbol{\Phi}exp(\Omega t)\mathbf{b} \tag{1.23}$$

where $\omega_k = \ln{(\lambda_k)}/\Delta t$, and $\mathbf{b} = \boldsymbol{\Phi}^{\dagger}\mathbf{x}_1$ where $\mathbf{b}$ is the least-squares solution to $\mathbf{x}_1 = \boldsymbol{\Phi}\mathbf{b}$.

Thus, given snapshots of a dynamical system in time, we can approximate the dynamics

and future states with a locally linear approach given by dynamic mode decomposition.

1.5.2 SPARSE IDENTIFICATION OF NONLINEAR DYNAMICAL SYSTEMS

The method of *sparse identification of nonlinear dynamical systems* (SINDy) uses snapshots of a dynamical system to recover the governing equations of the system. We assume that we can observe and measure a time-dependent process of one or more states. For example, for the movement of an oscillator in three dimensional space, the state vector $\mathbf{u} \in \mathbb{R}^3$ would have three components x, y, z. We aim to recover the governing system of equations for the dynamical system of interest.

1.5.2.1 Setting up the overdetermined least squares problem

We form the data matrix $\boldsymbol{X} \in \mathbb{R}^{n \times q}$ by collecting observations of $\mathbf{u} \in \mathbb{R}^q$ at n points in time. We approximate the time derivative of these states in time using a differentiation scheme to form $\dot{\boldsymbol{X}}$. In this paper we implement a first order forward difference scheme,

$$\dot{\mathbf{u}}_i \approx \frac{\mathbf{u}(t_i + h) - \mathbf{u}(t_i)}{h} \tag{1.24}$$

where $\mathbf{u_i}$ are the observations at time t_i and h is the time-step used to compute derivatives. The authors of [4] suggest that when the snapshots are collected from a system with noise, a *total variation regularized derivative scheme* [8] should be used to compute $\dot{\boldsymbol{X}}$.

We next construct the *library matrix* $\boldsymbol{\Theta} \in \mathbb{R}^{n \times p}$. This library matrix is formed by taking p functional transformations of the columns of $\boldsymbol{X}$ and represents the possibilities of the functions which might be present in the *true* governing system g. Thus we have

$$\boldsymbol{X} = \begin{bmatrix} \mathbf{u}^T(t_1) \\ \mathbf{u}^T(t_2) \\ \vdots \\ \mathbf{u}^T(t_n) \end{bmatrix} \quad \dot{\boldsymbol{X}} = \begin{bmatrix} \dot{\mathbf{u}}^T(t_1) \\ \dot{\mathbf{u}}^T(t_2) \\ \vdots \\ \dot{\mathbf{u}}^T(t_n) \end{bmatrix} \tag{1.25}$$

$$\boldsymbol{\Theta}(\boldsymbol{X}) = \begin{bmatrix} 1 & \boldsymbol{X} & \boldsymbol{X}^{P_2} & \boldsymbol{X}^{P_3} \cdots \sin(\boldsymbol{X}) & \cos(\boldsymbol{X}) \cdots \end{bmatrix}. \tag{1.26}$$

The notation $\boldsymbol{X}^{P_i}$ indicates monomials of the states up to order i. For example, for the states x, y, z of a system, $\boldsymbol{X}^{P_2} = [x^2, y^2, z^2, xy, xz, yz]$.

We now seek a *coefficient matrix* $\boldsymbol{\Xi} \in \mathbb{R}^{p \times q}$,

$$\boldsymbol{\Xi} = \begin{bmatrix} \xi_1 & \xi_1 & \dots & \xi_q \end{bmatrix}, \ \xi_i \in \mathbb{R}^p \tag{1.27}$$

such that

$$\dot{\boldsymbol{X}} \approx \boldsymbol{\Theta}(\boldsymbol{X})\boldsymbol{\Xi} \tag{1.28}$$

The coefficients of $\boldsymbol{\Xi}$ will thus indicate which, and how much, of the functions of $\boldsymbol{\Theta}$ are in the *true* governing equations, $g(\mathbf{u})$. Given the various approximations present in the formulation of (1.28) (i.e., data observation errors, approximation of $\dot{\boldsymbol{X}}$, and errors in $\boldsymbol{\Theta}(\boldsymbol{X})$), an exact solution is unlikely or impossible. Thus we seek a solution to the minimization problem given by

$$\min_{\boldsymbol{\xi}_i} \left\{ ||\dot{\mathbf{x}}_i - \boldsymbol{\Theta}(\boldsymbol{X})\xi_i||_2^2 + \alpha||\xi||_1 \right\}, \ \ i = 1, 2, \cdots, q. \tag{1.29}$$

A note on the choice of functions in $\boldsymbol{\Theta}$ The transformations chosen can be informed by intuition of the underlying physical system or of the observed dynamics. The authors of [4], however, argue that the library matrix can contain *as many* functional transformations as the user chooses. This argument is based on the experience that unnecessary functions present in $\boldsymbol{\Theta}$ will result in a corresponding 0 coefficient in $\boldsymbol{\Xi}$. If, however, a function present in the *true* governing equations is not included in the library matrix, it will be impossible to recover with this algorithm. If we know nothing about the true governing equations of the system from which we collect data, then we have no way to verify whether or not we are actually recovering the same system.

1.5.2.2 Sequential thresholded least squares

The method of *sequential thresholded least squares* was developed in [4] to solve the problem given by (1.29). At the heart of this algorithm is an assumption of *sparsity* in the space of all functions. The *thresholding* aspect of the method imposes this sparsity and aids in the recovery of the underlying dynamical system. As an example of how this sparsity may be imposed, consider the function $e^x \approx 1 + x + \frac{x^2}{2} + \frac{x^3}{6}$. If we observed data from this function and formed the library matrix as $\boldsymbol{\Theta(X)} = \begin{bmatrix} e^{\boldsymbol{X}} & \mathbf{1} & \boldsymbol{X} & \boldsymbol{X^2} & \boldsymbol{X^3} \end{bmatrix}$ A *reasonable* solution to (1.29) could either be $\boldsymbol{\Xi} = [1\,0\,0\,0\,0]^T$ or $\boldsymbol{\Xi} = [0\,1\,1\,\frac{1}{2}\,\frac{1}{6}]^T$. The solution vector identifying only e^x, however, is most *sparse*. The following method has been developed for such examples and has been used to recover the governing dynamics of many systems [4].

Algorithm 3 Sequential thresholded least squares [4]

Given $\dot{\boldsymbol{X}}$, $\boldsymbol{\Theta(X)}$, and parameters λ and k:

(1) Let $\boldsymbol{\Xi}_0$ be the least squares solution to the problem,

$$\min_{\boldsymbol{\Xi}} ||\dot{\boldsymbol{X}} - \boldsymbol{\Theta(X)}\boldsymbol{\Xi}||_2^2. \tag{1.30}$$

(2) Repeat k times:

(a) For all entries $\xi_{i,j}$ of matrix $\boldsymbol{\Xi}$, if $\xi_{i,j} < \lambda$, update such that $\xi_{i,j} = 0$.

(b) for each column ξ_j of $\boldsymbol{\Xi}$ and column $, \dot{\mathbf{u}}_j$ of $\dot{\boldsymbol{X}}$:

(i) Let $r_1, \ldots r_\ell$ be the rows of ξ_j whose entries are not equal to zero. Let ξ^* be the column vector whose rows are the corresponding nonzero entries of ξ_j.

(ii) Let $\boldsymbol{\Theta}^*$ be the matrix whose columns are the columns $r_1, \ldots r_\ell$ of $\boldsymbol{\Theta}$.

(iii) Update $\xi^* = leastsquares(\dot{\mathbf{u}}_j, \boldsymbol{\Theta}^*)$.

The coefficient matrix, $\boldsymbol{\Xi}$ returned from Algorithm 3 indicates which functions of the library matrix are in the true governing equations, $g(\mathbf{x})$. For example, in a three state system where

$$\boldsymbol{\Xi} = \begin{bmatrix} 0 & 3 & 22 \\ 6 & 0 & 9 \\ 0 & 0.8 & 2 \end{bmatrix}, \boldsymbol{\Theta} = \begin{bmatrix} \sin x & xy & z^3 \end{bmatrix},$$

the governing equations recovered by SINDy would be

$$\dot{x} = 6xy$$

$$\dot{y} = 3\sin x + 0.8z^3$$

$$\dot{z} = 22\sin x + 9xy + z^3$$

Practicalities

Assumption of sparsity The choice of using this method to solve the overdetermined least squares problem is based on a few assumptions that are necessary to address. The first assumption is that the functions composing the governing system $g(\mathbf{x})$ are within the library matrix $\boldsymbol{\Theta}$. The second assumption is that the true governing system is *sparse* in the space of all functions. This means that of all possible functions, the governing equations of the dynamical system will only be composed of a few. The validity of this assumption for the case in which we will employ the algorithm will be discussed in Section 1.5.2.3.

Choice of tuning parameter λ The sequential thresholded least squares algorithm for solving the ℓ_1 minimization problem given by (1.29) is extremely and unpredictably sensitive to the choice of λ. Because the values of the coefficient matrix are thresholded according to this parameter at each step, the choice of λ will drastically affect the recovered dynamical system. This sensitivity and unpredictability of the choice of λ is also what makes the complexity analysis of this algorithm difficult. The cost of each least squares computation at each iteration in the sequential thresholded least squares approach to the problem is determined by the size of the matrix and vector. These sizes are dependent on the value of λ.

1.5.2.3 Least absolute shrinkage and selection operator (LASSO)

Although sequential thresholded least squares has been empirically shown to recover the governing dynamical system, the practicalities noted above make it difficult to understand how the method of SINDy can be translated to new problems. For example, to actually *recover* a dynamical system about which one has no intuition (stay tuned!), choosing an appropriate value for λ can be difficult. Alternatively, if there is no system governing the observed dynamics, one may want to model the dynamics instead. In such a case, this notion of sparsity may be incorrect. Thus we attempt to find a solution to the ℓ_1 minimization least squares problem (1.29) by taking advantage of the well-developed regression analysis methods developed to solve the problem of the *least absolute shrinkage and selection operator* (LASSO) [48] [22], defined as (1.29). For our purposes we use the `Lasso` function in python's `scikit-learn` package. This method for solving (1.29) involves the tuning of the parameter α which controls the relative importance of the 1-norm of the solution. The effect of this parameter on the solution is studied in Section 2.3.3.

Ill-conditioning The $\boldsymbol{\Theta}$ may be ill-conditioned. This ill conditioning affects the ability to recover a solution with least squares. To assist in the solution we impose the ℓ_1 regularization condition defined by the α term in (1.29).

Now that we've presented the necessary background information on active subspaces and the data-driven recovery of dynamical systems techniques of DMD and SINDy, we proceed to introduce the idea of *dynamic active subspaces.*

CHAPTER 2

2.1 DYNAMIC ACTIVE SUBSPACES

Active subspaces enable dimension reduction and sensitivity analysis for computational models. As is, this analysis does not explicitly extend to functions that are dependent on both parameter inputs and time. Extending active subspaces to time-dependent systems enables uncertainty quantification, sensitivity analysis, and parameter estimation for computational models that have explicit dependence on time.

2.1.1 The problem

To begin, consider a system of parameterized ordinary differential equations with q states $\mathbf{u}$ and m parameters $\mathbf{p}$,

$$\dot{\mathbf{u}} = g(\mathbf{u}, \mathbf{p}), \quad \mathbf{u} \in \mathbb{R}^q, \quad \mathbf{p} \in \mathbb{R}^m. \tag{2.1}$$

Let us consider a specific time-dependent quantity of interest (QoI) of these states, dependent on states $\mathbf{u} \in \mathbb{R}^q$ and parameters $\mathbf{p} \in \mathbb{R}^m$. This QoI is defined as

$$f(t, \mathbf{p}) = F(\mathbf{u}(t, \mathbf{p})) : \mathbb{R}^m \to \mathbb{R}. \tag{2.2}$$

Note that f can be a nonlinear function of the states, and that these states can be nonlinearly dependent on the parameters. In practice the solution $\mathbf{u}(t, \mathbf{p})$ is numerically computed.

2.1.2 The proposed solution

The main idea Suppose we want to identify the linear combinations of the parameters that change the QoI the most on average, and we want to know how these linear combinations change in time. Doing so would identify *the dynamic active subspaces* (DyAS) and would give quantifiable insight into the parameter sensitivity of time-dependent systems. We propose and discuss the following three approaches to identifying dynamic active subspaces.

Computing DyAS at independent times One option to identify these time-dependent active subspaces is to independently compute them at each step in time. Specifically, we could find the QoI at time t_0, compute the active subspace, find the QoI at time t_1, compute the active subspace, find the QoI at time t_2, compute the active subspace, and repeat this until we reach final time t_n. Such analysis and identification has been done in [33] and [14]. This approach to finding time-dependent active subspaces, however, is extremely expensive, especially in systems with many parameters. This computation can be reduced by only computing active subspaces at "interesting" points in time (i.e., stages of a disease, transient dynamics, final time, time of maximum concentration). However, what is considered an "interesting" point in time for nominal parameter values might not be for other values in the parameter space.

Analytically defining DyAS Rather than computing the active subspace independently throughout time, consider the possibility of defining a time-dependent $\boldsymbol{C}(t)$ with eigenvalues $\lambda_i(t)$ and a governing dynamical system for $\mathbf{w}_i(t)$. The computation involved in computing the active subspace at each time step could be bypassed by simply integrating the dynamical system for $\mathbf{w}_i(t)$. Classical analysis for dynamical systems could then give further insight into these dynamic active subspaces and, in turn, the physical systems from which they are derived. In Section 2.2 we present three cases for which we can use this approach to define DyAS.

Approximating the DyAS In cases for which there does not exist an analytical form for the DyAS or for which there is no analytical form for the dynamical system, we can consider *approximating* the dynamics of the active subspace. Below we propose the use of DMD and SINDy (see Section 1.5 for details) to approximate the one-dimensional DyAS from *snapshots* of the first eigenvector of $\boldsymbol{C}$ in time.

2.2 ANALYTIC DYNAMIC ACTIVE SUBSPACES

Just as we can describe the dynamics of the state space of a physical system with an ordinary differential equation, we propose similarly describing the dynamics of the state space of an eigenvector. An ODE describing the first eigenvector of $\boldsymbol{C}$ over time would give an analytical form of the time derivative for the one-dimensional dynamic active subspace. Below we prove the existence of two such dynamical systems for dynamic active subspaces.

2.2.1 DyAS for an homogeneous linear dynamical system

Consider a linear dynamical system,

$$\dot{\mathbf{u}} = \boldsymbol{A}\mathbf{u}, \;\; \mathbf{u}(0) = \mathbf{p}, \quad \mathbf{u}(t, \mathbf{p}), \;\; \mathbf{p} \in \mathbb{R}^q, \;\; \boldsymbol{A} \in \mathbb{R}^{q \times q}. \tag{2.3}$$

The solution to this dynamical system is,

$$\mathbf{u}(t, \mathbf{p}) = e^{\boldsymbol{A}t}\mathbf{p}. \tag{2.4}$$

Consider a time-dependent scalar output of the dynamical system, $f(t, \mathbf{p}) = \phi^T \mathbf{u}(t, \mathbf{p})$, for $\phi \in \mathbb{R}^q$. For example, where $f(t, \mathbf{p})$ is the ith state of the dynamical system, ϕ is the ith column of the identity matrix. Assume that the initial conditions of the dynamical system (2.3) have joint density $\rho(\mathbf{p})$. Thus the time-dependent scalar output of the parameterized dynamical system (2.3) is

$$f(t, \mathbf{p}) = \phi^T e^{\boldsymbol{A}t}\mathbf{p}. \tag{2.5}$$

The exact time-dependent gradient of f is

$$\nabla f(t, \mathbf{p}) = (e^{\boldsymbol{A}t})^T \phi, \tag{2.6}$$

and thus the time-dependent matrix $\boldsymbol{C}(t)$ is,

$$\begin{aligned} \boldsymbol{C}(t) &= \int \left((e^{\boldsymbol{A}t})^T \phi\right) \left((e^{\boldsymbol{A}t})^T \phi\right)^T \rho(\mathbf{p}) d\mathbf{p} && (2.7)\\ &= \int \left((e^{\boldsymbol{A}t})^T \phi\right) \left(\phi^T e^{\boldsymbol{A}t}\right) \rho(\mathbf{p}) d\mathbf{p} && (2.8)\\ &= (e^{\boldsymbol{A}t})^T \phi\phi^T e^{\boldsymbol{A}t} \int \rho(\mathbf{p}) d\mathbf{p} && (2.9)\\ &= (e^{\boldsymbol{A}t})^T \phi\phi^T e^{\boldsymbol{A}t}. && (2.10) \end{aligned}$$

Note that $\text{Rank}(\phi\phi^T) = 1$ and thus $\text{Rank}(\boldsymbol{C}(t)) = 1$. Therefore $\boldsymbol{C}(t)$ has only one nonzero eigenvalue with corresponding normalized eigenvector,

$$\mathbf{v}_1(t) = \frac{(e^{\boldsymbol{A}t})^T \phi}{||(e^{\boldsymbol{A}t})^T \phi||_2}. \tag{2.11}$$

This eigenvector is only unique up to a constant. Thus we seek a governing dynamical system for the unnormalized first eigenvector.

Let $\mathbf{w}(t) = (e^{\boldsymbol{A}t})^T \phi$. Then the dynamical system for the unnormalized first eigenvector of $\boldsymbol{C}(t)$ is described by the linear dynamical system,

$$\dot{\mathbf{w}}(t) = A^T \mathbf{w}(t), \;\; \mathbf{w}(0) = \phi. \tag{2.12}$$

2.2.2 DyAS for an inhomogeneous linear dynamical system

Consider the inhomogeneous linear dynamical system,

$$\dot{\mathbf{u}} = \boldsymbol{A}\mathbf{u} + \mathbf{p}, \;\; \mathbf{u}(0) = \eta, \quad \mathbf{u}(t, \mathbf{p}),\, \eta,\, \mathbf{p} \in \mathbb{R}^q, \;\; \boldsymbol{A} \in \mathbb{R}^{q \times q}. \tag{2.13}$$

For invertible $\boldsymbol{A}$ the solution to this dynamical system is given by *Duhamel's Principle* [32],

$$\mathbf{u}(t, \mathbf{p}) = e^{\boldsymbol{A}t}\eta + \boldsymbol{A}^{-1}(e^{\boldsymbol{A}t} - I)\mathbf{p}. \tag{2.14}$$

Consider a time-dependent scalar output of the dynamical system, $f(t, \mathbf{p}) = \phi^T \mathbf{u}(t, \mathbf{p})$, for $\phi \in \mathbb{R}^n$. Assume that $\mathbf{p}$ has a joint density $\rho(\mathbf{p})$. Thus the time-dependent scalar output of the parameterized dynamical system (2.13) is

$$f(t, \mathbf{p}) = \phi^T e^{\boldsymbol{A}t}\eta + \phi^T \boldsymbol{A}^{-1}(e^{\boldsymbol{A}t} - I)\mathbf{p}. \tag{2.15}$$

The exact time-dependent gradients of f are

$$\nabla f(t, \mathbf{p}) = ((e^{\boldsymbol{A}t})^T - I)\boldsymbol{A}^{-T}\phi, \tag{2.16}$$

and thus the time-dependent matrix $\boldsymbol{C}(t)$ is given by

$$\mathbf{C}(t) = \int \left(((e^{\boldsymbol{A}t})^T - I)\boldsymbol{A}^{-T}\phi\right)\left(((e^{\boldsymbol{A}t})^T - I)\boldsymbol{A}^{-T}\phi\right)^T \rho(\mathbf{p})d\mathbf{p} \tag{2.17}$$

$$= \int ((e^{\boldsymbol{A}t})^T - I)\boldsymbol{A}^{-T}\phi\phi^T\boldsymbol{A}^{-1}(e^{\boldsymbol{A}t} - I)\rho(\mathbf{p})d\mathbf{p} \tag{2.18}$$

$$= ((e^{\boldsymbol{A}t})^T - I)\boldsymbol{A}^{-T}\phi\phi^T\boldsymbol{A}^{-1}(e^{\boldsymbol{A}t} - I)\int \rho(\mathbf{p})d\mathbf{p} \tag{2.19}$$

$$= ((e^{\boldsymbol{A}t})^T - I)\boldsymbol{A}^{-T}\phi\phi^T\boldsymbol{A}^{-1}(e^{\boldsymbol{A}t} - I). \tag{2.20}$$

Note that $\text{Rank}(\phi\phi^T) = 1$ and thus $\text{Rank}(\boldsymbol{C}(t)) = 1$. Therefore $\boldsymbol{C}(t)$ has only one nonzero eigenvalue, with corresponding normalized eigenvector,

$$\mathbf{v}_1(t) = \frac{((e^{\boldsymbol{A}t})^T - I)\boldsymbol{A}^{-T}\phi}{||((e^{\boldsymbol{A}t})^T - I)\boldsymbol{A}^{-T}\phi||_2}. \tag{2.21}$$

This eigenvector is only unique up to a constant. Thus we seek a governing dynamical system for the unnormalized first eigenvector.

Let

$$\mathbf{w}(t) = ((e^{\boldsymbol{A}t})^T - I)\boldsymbol{A}^{-T}\phi. \tag{2.22}$$

Then the dynamical system of the unnormalized first eigenvector of $\boldsymbol{C}(t)$ is described by the inhomogeneous linear dynamical system,

$$\dot{\mathbf{w}}(t) = \boldsymbol{A}^T\mathbf{w}(t) + \phi, \;\; \mathbf{w}(0) = \mathbf{0}. \tag{2.23}$$

Note that Duhamel's principle on (2.23) yields

$$\mathbf{w}(t) = (e^{\boldsymbol{A}t})^T\mathbf{w}_0 + \boldsymbol{A}^{-T}((e^{\boldsymbol{A}t})^T - I)\phi = \boldsymbol{A}^{-T}((e^{\boldsymbol{A}t})^T - I)\phi.$$

Consider $\boldsymbol{A} = \boldsymbol{W}\boldsymbol{\Lambda}\boldsymbol{W}^{-1}$ and recall the definition of the matrix exponential [32], $e^{\boldsymbol{A}t} = \boldsymbol{W}e^{\boldsymbol{\Lambda}t}\boldsymbol{W}^{-1}$. The expression for $\mathbf{w}(t)$ given by Duhamel's principle is equivalent to (2.22),

$$\begin{aligned}
\boldsymbol{A}^{-T}((e^{\boldsymbol{A}t})^T - I)\phi &= \boldsymbol{A}^{-T}(e^{\boldsymbol{A}t})^T\phi - \boldsymbol{A}^{-T}\phi \\
&= \boldsymbol{W}^{-T}\boldsymbol{\Lambda}^{-1}\boldsymbol{W}^T\boldsymbol{W}^{-T}e^{\boldsymbol{\Lambda}t}\boldsymbol{W}^T\phi - \boldsymbol{A}^{-T}\phi \\
&= \boldsymbol{W}^{-T}\boldsymbol{\Lambda}^{-1}e^{\boldsymbol{\Lambda}t}\boldsymbol{W}^T\phi - \boldsymbol{A}^{-T}\phi \\
&= \boldsymbol{W}^{-T}e^{\boldsymbol{\Lambda}t}\boldsymbol{\Lambda}^{-1}\boldsymbol{W}^T\phi - \boldsymbol{A}^{-T}\phi \\
&= e^{\boldsymbol{A}^Tt}\boldsymbol{A}^{-T}\phi - \boldsymbol{A}^{-T}\phi \\
&= ((e^{\boldsymbol{A}t})^T - I)\boldsymbol{A}^{-T}\phi.
\end{aligned}$$

The analyses of sections 2.2.1 and 2.2.2 can be extended for the case where the parameters of an inhomogeneous linear dynamical system (2.13) are the initial conditions η. Thus we prove the existence for the analytic form of the dynamic active subspace for three cases. For these specific problems, the cost of computing the active subspace at each time point of interest is avoided by simply evaluating a linear dynamical system which gives exactly the one-dimensional active subspace of the time-dependent QoI. The existence of the governing equations of the dynamic active subspace for these cases is directly comparable to the existence of an analytical static active subspace for a QoI with linear dependence on its parameters. As we saw in the above analyses, when the gradient of f is not parameter-dependent, the outer product in (1.1) can be pulled outside of the integral, and the resulting matrix will be rank-one. A rank-one $\boldsymbol{C}$ guarantees a one-dimensional active subspace.

However exciting as this may be (we think it's pretty cool), we must acknowledge that these analytic forms exist (so far!) for three very specific cases. We would like to have a form of the dynamic active subspace for *any* time dependent, differentiable QoI. Thus in the following section we propose a method for finding an approximate dynamic active subspace.

2.3 APPROXIMATING DYNAMIC ACTIVE SUBSPACES

In this section we propose the use of DMD (see Section 1.5.1) and SINDy (see Section 1.5.2) to approximate the dynamic active subspace.

2.3.1 Introduction to the methodology

The methods of DMD and SINDy to recover and approximate dynamics rely on a collection of *snapshots* of the system. In the case of dynamic active subspaces, these snapshots will be components of the first eigenvector of $\boldsymbol{C}$ in time. We propose collecting these snapshots and using DMD and SINDy to approximate the dynamic active subspace between snapshots and predict future DyAS. Below we present the methodology.

Methodology *Approximating the dynamic active subspace*

(1) Choose a quantity of interest from the parameterized dynamical system.

(2) Augment the dynamical system with states for the partial derivatives to compute Gronwall gradients (1.11) (Alternatively, use a finite difference method to approximate the gradients $\nabla f(\mathbf{p}, t)$).

(3) Identify a numerical integration rule for approximating $\boldsymbol{C}$.

(4) For each point $\mathbf{p}_i$ in the set of integration nodes, integrate the dynamical system to compute $f(t_i, \mathbf{p}_i)$ and $\nabla f(t_i, \mathbf{p}_i)$ for each $t_i \in [t_1, t_2, t_3, \cdots, t_{n-1}, t_n]$.

(5) For each t_i, estimate $\boldsymbol{C}(t_i)$ and compute the first eigenvector $\mathbf{w}(t_i)$.

   (a) If using SINDy to approximate the dynamics, estimate $\boldsymbol{C}(t_i)$ and $\mathbf{w}(t_i)$ at n additional points in time, $[t_1 + h,\, t_2 + h,\, t_3 + h,\, \cdots,\, t_{n-1} + h,\, t_n + h]$ for small h. Note that h is the time step used to approximate the time derivative for the dynamics of the eigenvector.

(6) Construct the data matrices with the first eigenvector $\mathbf{w}$ at each t_i

For DMD:

(a) Construct $\boldsymbol{X_{DMD}} \in \mathbb{R}^{m\times(n-1)}$ with the first $(n-1)$ eigenvectors:

$$\boldsymbol{X_{DMD}} = \begin{bmatrix} | & | & | & | & | \\ \mathbf{w}(t_1) & \mathbf{w}(t_2) & \cdots & \mathbf{w}(t_{n-2}) & \mathbf{w}(t_{n-1}) \\ | & | & | & | & | \end{bmatrix}$$

(b) Construct $\boldsymbol{X'_{DMD}} \in \mathbb{R}^{m\times(n-1)}$ with the last $(n-1)$ eigenvectors:

$$\boldsymbol{X'_{DMD}} = \begin{bmatrix} | & | & | & | & | \\ \mathbf{w}(t_2) & \mathbf{w}(t_3) & \cdots & \mathbf{w}(t_{n-1}) & \mathbf{w}(t_n) \\ | & | & | & | & | \end{bmatrix}$$

For SINDy:

(a) Construct $\boldsymbol{X_{SINDy}} \in \mathbb{R}^{n\times m}$ with the first eigenvector $\mathbf{w}$ at $[t_1, t_2, t_3, \cdots, t_{n-1}, t_n]$:

$$\boldsymbol{X_{SINDy}} = \begin{bmatrix} \mathbf{w}(t_1)^T \\ \mathbf{w}(t_2)^T \\ \vdots \\ \mathbf{w}(t_{n-1})^T \\ \mathbf{w}(t_n)^T \end{bmatrix}$$

(b) Use a first order finite difference scheme,

$$\dot{\mathbf{w}}(t_i) \approx \frac{\mathbf{w}(t_i + h) - \mathbf{w}(t_i)}{h}$$

to construct $\boldsymbol{\dot{X}_{SINDy}} \in \mathbb{R}^{n\times m}$:

$$\boldsymbol{\dot{X}_{SINDy}} = \begin{bmatrix} \dot{\mathbf{w}}(t_1)^T \\ \dot{\mathbf{w}}(t_2)^T \\ \vdots \\ \dot{\mathbf{w}}(t_{n-1})^T \\ \dot{\mathbf{w}}(t_n)^T \end{bmatrix}$$

(7) Let $\mathbf{T} \in \mathbb{R}^n$ be the vector of times at which to predict the dynamic active subspace. Use Algorithm 2 and/or Algorithm 3 to construct an approximation for the dynamic active subspace for all $t \in \mathbf{T}$.

Below we demonstrate this methodology to approximate the dynamic active subspace for two dynamical systems: (i) a linear harmonic oscillator and (ii) an enzyme kinetics system.

2.3.2 Example: Linear Harmonic Oscillator

Consider a linear harmonic oscillator whose dynamics are given by

$$\dot{\mathbf{u}} = \begin{bmatrix} -0.1 & -2 & 0 \\ 2 & -0.1 & 0 \\ 0 & 0 & -0.3 \end{bmatrix} \mathbf{u}, \ \mathbf{u}(t_0) = \mathbf{p} \tag{2.24}$$

The initial conditions, $\mathbf{p} \in \mathbb{R}^3$ are uncertain between the lower and upper bounds,

$$\mathbf{p}_\ell = [\,0.96,\, 4.0,\, 8.0\,]^T, \ \ \mathbf{p}_u = [\,1.44,\, 6.0,\, 12.0\,]^T.$$

This dynamical system has an analytic solution given by

$$\mathbf{u}(t) = e^{\boldsymbol{A}t}\mathbf{p}, \ \ \boldsymbol{A} = \begin{bmatrix} -0.1 & -2 & 0 \\ 2 & -0.1 & 0 \\ 0 & 0 & -0.3 \end{bmatrix}. \tag{2.25}$$

At nominal initial conditions, $\mathbf{p}_{nominal} = [\,1.2,\, 5.0,\, 10.0\,]^T$, the trajectory of the linear harmonic oscillator in 3 dimensions is given by Figure 2.1.

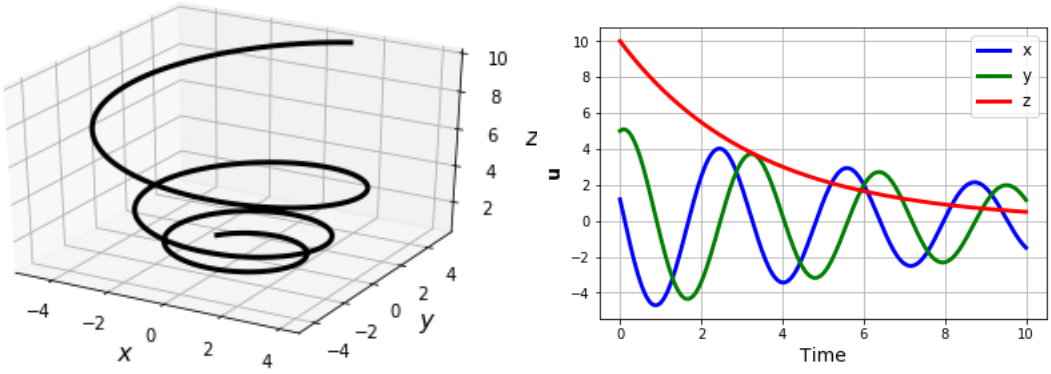

Figure 2.1: The trajectory of the x, y, and z components of the linear harmonic oscillator given by (2.24) with nominal initial conditions.

Consider the y component as the quantity of interest. Then for $\phi = [\,0,\ 1,\ 0\,]^T$, let

$$f(t, \mathbf{p}) = \phi^T \mathbf{u} = \phi^T e^{\boldsymbol{A}t} \mathbf{p}. \tag{2.26}$$

Note that for this system with the initial conditions as parameters, the governing system of the dynamic active subspace is given by (2.12). For the sake of demonstrating the methodology for this simple example we ignore this until it comes time to evaluate our approximation.

The gradient of this function with respect to $\mathbf{p}$ can be taken exactly from (2.26) for all time,

$$\nabla f(t, \mathbf{p}) = e^{\boldsymbol{A}^T t} \phi. \tag{2.27}$$

We will use this evaluation of the gradient to approximate $\boldsymbol{C}$ with Algorithm 1 and Monte Carlo sampling. Although we would normally conduct a convergence study on the number of samples M for our Monte Carlo approximation, it is not necessary in this case because we know that the gradients (2.27) are independent of our samples $\mathbf{p}_i$. To have enough points to construct the shadow plot in Figure 2.2, we let $N = 50$. We assume that $\mathbf{p}$ is uniformly distributed with upper and lower bounds 20% above and below the nominal values (see Table 2.1).

Parameter	Lower Bound	Nominal Value	Upper Bound
x_0	0.96	1.2	1.44
y_0	4.0	5.0	6.0
z_0	8.0	10.0	12.0

Table 2.1: The initial conditions $\mathbf{p}$ in (2.26) have a uniform joint density with lower and upper bounds 20% below and above nominal values.

To examine and explain the different aspects of the active subspace, we first compute $\boldsymbol{C}$ at time $t = 5$.

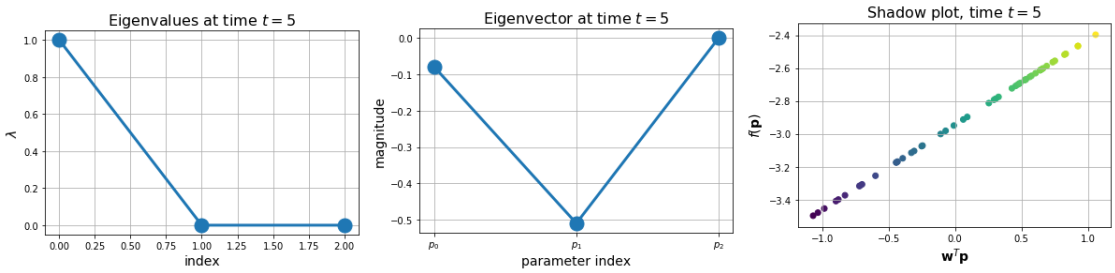

Figure 2.2: The eigenvalues (left) and first eigenvector (middle) of $\boldsymbol{C}$ at time $t = 5$. Note that there is only one nonzero eigenvalue. The shadow plot (right) at this time shows the value of the linear harmonic oscillator y coordinate as a function of a linear combination of possible initial conditions, $\mathbf{p}$.

The rightmost plot of Figure 2.2 shows the *shadow plot* for the active subspace at time $t = 5$. The shadow plot reveals the low-dimensional structure in $f(t = 5, \mathbf{p})$ by viewing the function evaluations in the direction defined by $\mathbf{w}$.

Given the methodology from Section 2.3.1, we seek to form data matrices using snapshots of the eigenvector $\mathbf{w}$ at n points in time. We compute the first eigenvector from $t_0 = 0.001$ to $t_F = 10.0$ at 8000 points in time by independently computing the active subspace at each time step. The time-series of this eigenvector will be used as the "truth" to

which we will compare all future approximations. Figure 2.3 shows an interesting aspect of the computation. Using `numpy`'s `eig` function, the eigenvectors of $\boldsymbol{C}$ are normalized at each time step. Using snapshots from the left of Figure 2.3 will be problematic if aiming to use a linear approximation method (DMD) or a derivative-based method (SINDy) in recovering the dynamics. Following analysis from Section 2.2 we observe that multiplying the computed time series by the square root of the first computed eigenvalue of $\boldsymbol{C}$ yields the same dynamics given by the dynamical system (2.12). Thus in the following experiments we use this unnormalized form of the eigenvector to construct the data matrices for DMD and SINDy.

Important to note is that the sign of an eigenvector is not unique (an eigenvector is still an eigenvector if multiplied by negative one). Thus when computing eigenvectors for multiple points in time, we must normalize the sign of the eigenvector at each time step. We do so by checking,

$$||\mathbf{w}(t_i) + \mathbf{w}(t_{i+1})||_2 > ||\mathbf{w}(t_i) - \mathbf{w}(t_{i+1})||_2. \tag{2.28}$$

If (2.28) is true, $\mathbf{w}(t_{i+1})$ is updated such that $\mathbf{w}(t_{i+1}) = -\mathbf{w}(t_{i+1})$. This ensures that any discontinuity observed in the time-dependent eigenvector is not due to a difference in signs.

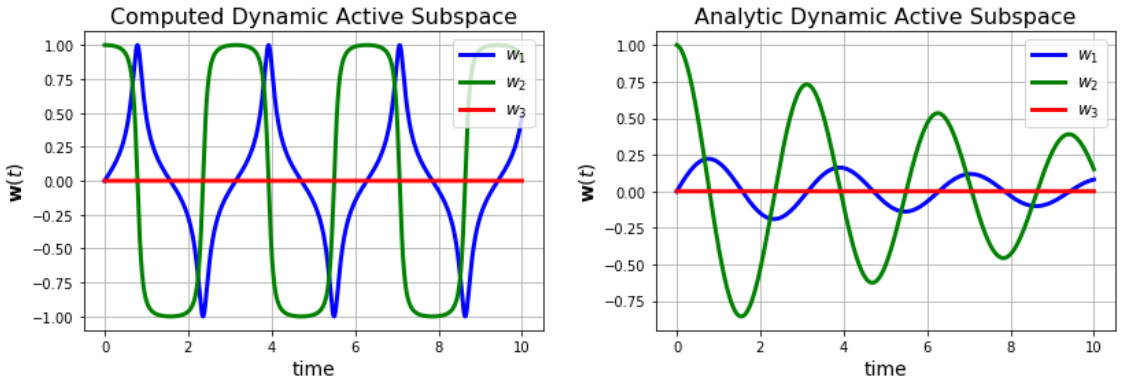

Figure 2.3: *(Left)* the first eigenvector of $\boldsymbol{C}$ computed for the function (2.26) *(Right)* the first eigenvector after being multiplied by $\sqrt{\lambda_0(t)}$.

2.3.2.1 Numerical Experiments

The methods of DMD and SINDy depend on data matrices formed with snapshots of the dynamical system in time. We test the accuracy of these methods in recovering the dynamic active subspace as a function of (i) the *proportion of time* the snapshots represent, i.e. snapshots from the first two out of ten seconds of a system would represent 0.2 of the data, and (ii) the *number of samples* in the data matrices.

In the following experiments we construct the data matrices $\boldsymbol{X_{DMD}}$ and $\boldsymbol{X_{SINDy}}$ with snapshots from proportions of the data from 0.1 to 1.0. For each proportion we also construct the data matrices with 10, 20, 30, 40, 50, 75, 100, 200, 500, 1000, 2000, and 5000 snapshots. In the implementation of SINDy we use `scikit-learn`'s `Lasso` function with $\alpha = 10^{-5}$. In order to construct a data matrix $\boldsymbol{X_{SINDy}}$ with N snapshots at times $t_0, t_1, \cdots, t_N$, we compute an additional N eigenvectors at $t_0 + h, t_1 + h, \cdots, t_N + h$ for $h = 10^{-6}$ with which to approximate $\dot{\boldsymbol{X}}$. The library matrix $\boldsymbol{\Theta}$ is constructed with monomials of up to degree 5, trigonometric combinations of up to degree 3, and exponential combinations of up to degree 3. In Figure 2.4, the error for each experiment is computed as the maximum absolute error between the approximated dynamic active subspace, and the "true" active subspace computed at all 8000 time steps.

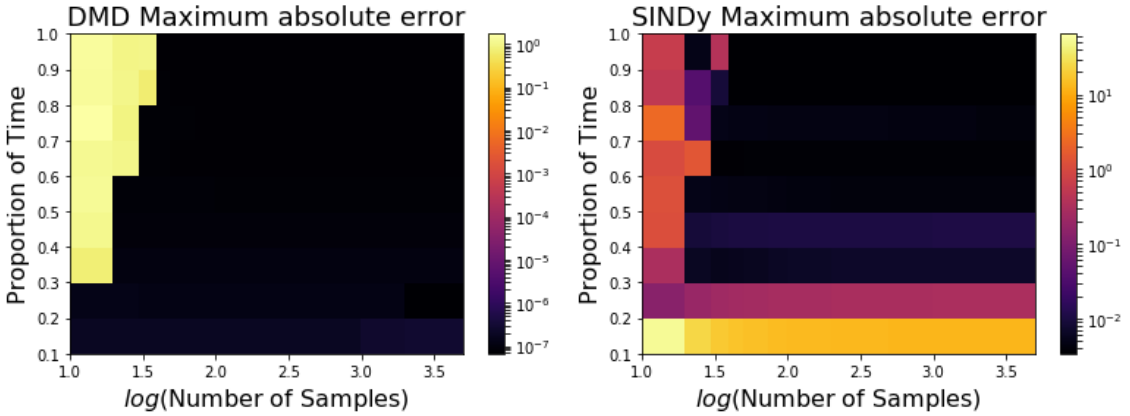

Figure 2.4: The maximum absolute error of the dynamic active subspace approximated by *(left)* DMD and *(right)* SINDy. The data matrices for each algorithm have been created with N samples from p proportion of the total time.

One of the reasons DMD and SINDy poorly approximate the dynamic active subspace when given few snapshots of data is due to the non-uniqueness of eigenvectors discussed above. For a small time step between computed eigenvectors (i.e., there are many snapshots), the normalization of signs by checking (2.28) ensures that the nearby eigenvectors have the same sign. However, when the time step between computed eigenvectors is large, the trajectory may have evolved significantly enough that (2.28) will not properly indicate what the sign "should" be. In such cases, the DMD and SINDy data matrices will be given snapshots that do not match the "true" state of the eigenvector.

2.3.3 Enzyme Kinetics

In this section we extend the methodology presented in Section 2.3.1 to a nonlinear dynamical system. The simple enzyme kinetics system given by (2.29) describes the chemical reaction of the catalysis of enzymes [26], and the reaction is written as,

$$S + E \underset{\mathrm{k}^-}{\overset{\mathrm{k}^+}{\rightleftharpoons}} ES \xrightarrow{\mathrm{k}^c} P + E. \tag{2.29}$$

The reaction (2.29) is described by the system of nonlinear ordinary differential equations in (2.33).

$$\frac{d[S]}{dt} = -k^+[E][S] + k^-[ES] \tag{2.30}$$

$$\frac{d[E]}{dt} = -k^+[E][S] + (k^- + k^c)[ES] \tag{2.31}$$

$$\frac{d[P]}{dt} = k^c[ES] \tag{2.32}$$

$$\frac{d[ES]}{dt} = k^+[E][S] - (k^- + k^c)[ES] \tag{2.33}$$

Let us consider the concentration of [ES] in time our quantity of interest,

$$f(t, \mathbf{p}) = [ES](t, \mathbf{p}) \tag{2.34}$$

We assume that the parameters, $\mathbf{p} = [k^+, k^-, k^c]^T$ have uniform joint density with lower and upper bounds 50% below and above the nominal values (see Table 2.2).

Parameter	Lower Bound	Nominal Value	Upper Bound
k^+	1.0	2.0	3.0
k^-	0.1	0.2	0.3
k^c	0.05	0.1	0.15

Table 2.2: The parameters $\mathbf{p}$ have uniform joint density with upper and lower bounds 50% above and below the nominal values.

Approximation of C Given that our quantity of interest only has 3 parameters, we approximate the integral in (1.1) with the tensor product Gauss-Legendre quadrature scheme. We conduct a numerical experiment to choose the number of quadrature points per dimension, M, in approximating (1.1). We choose to use 7 quadrature points per dimension for a total number of $7^3 = 343$ sample points (see Figure 2.5).

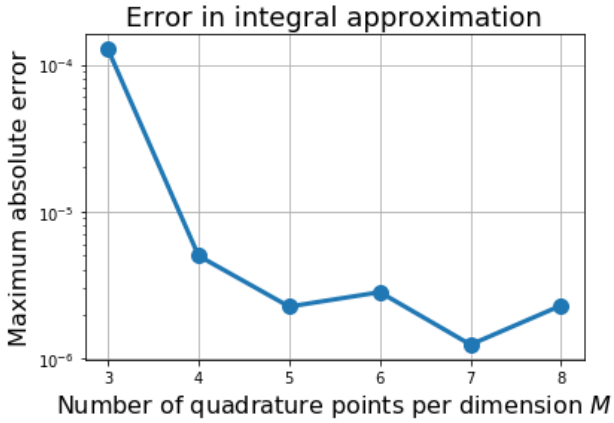

Figure 2.5: The convergence of the approximation to the integral in (1.1) for the number of nodes per dimension in the tensor product Gauss-Legendre quadrature scheme.

Gradients We augment the system of ODEs in (2.33) to include the Gronwall Gradients of each state with respect to each parameter according to (1.11).

$$\frac{d}{dt}\begin{pmatrix} [S] \\ [E] \\ [P] \\ [ES] \\ \frac{\partial [S]}{\partial k^+} \\ \frac{\partial [E]}{\partial k^+} \\ \frac{\partial [P]}{\partial k^+} \\ \frac{\partial [ES]}{\partial k^+} \\ \frac{\partial [S]}{\partial k^-} \\ \frac{\partial [E]}{\partial k^-} \\ \frac{\partial [P]}{\partial k^-} \\ \frac{\partial [ES]}{\partial k^-} \\ \frac{\partial [S]}{\partial k^c} \\ \frac{\partial [E]}{\partial k^c} \\ \frac{\partial [P]}{\partial k^c} \\ \frac{\partial [ES]}{\partial k^c} \end{pmatrix} = \begin{pmatrix} -k^+[E][S] + k^-[ES] \\ -k^+[E][S] + (k^- + k^c)[ES] \\ k^+[ES] \\ k^+[E][S] - (k^- + k^c)[ES] \\ -k^+[E]\frac{\partial [S]}{\partial k^+} - k^+[S]\frac{\partial [E]}{\partial k^+} + k^-\frac{\partial [ES]}{\partial k^+} - [E][S] \\ -k^+[E]\frac{\partial [S]}{\partial k^+} - k^+[S]\frac{\partial [E]}{\partial k^+} + (k^- + k^c)\frac{\partial [ES]}{\partial k^+} - [E][S] \\ k^c\frac{\partial [ES]}{\partial k^+} \\ k^+[E]\frac{\partial [S]}{\partial k^+} + k^+[S]\frac{\partial [E]}{\partial k^+} - (k^- + k^c)\frac{\partial [ES]}{\partial k^+} + [E][S] \\ -k^+[E]\frac{\partial [S]}{\partial k^-} - k^+[S]\frac{\partial [E]}{\partial k^-} + k^-\frac{\partial [ES]}{\partial k^-} + [ES] \\ -k^+[E]\frac{\partial [S]}{\partial k^-} - k^+[S]\frac{\partial [E]}{\partial k^-} + (k^- + k^c)\frac{\partial [ES]}{\partial k^-} + [ES] \\ k^c\frac{\partial [ES]}{\partial k^-} \\ k^+[E]\frac{\partial [S]}{\partial k^-} + k^+[S]\frac{\partial [E]}{\partial k^-} - (k^- + k^c)\frac{\partial [ES]}{\partial k^-} - [ES] \\ -k^+[E]\frac{\partial [S]}{\partial k^c} - k^+[S]\frac{\partial [E]}{\partial k^c} + k^-\frac{\partial [ES]}{\partial k^c} \\ -k^+[E]\frac{\partial [S]}{\partial k^c} - k^+[S]\frac{\partial [E]}{\partial k^c} + (k^- + k^c)\frac{\partial [ES]}{\partial k^c} + [ES] \\ k^c\frac{\partial [ES]}{\partial k^c} + [ES] \\ k^+[E]\frac{\partial [S]}{\partial k^c} + k^+[S]\frac{\partial [E]}{\partial k^c} - (k^- + k^c)\frac{\partial [ES]}{\partial k^c} - [ES] \end{pmatrix} \tag{2.35}$$

We compare the solutions for the partial derivative of f with respect to the parameters numerically integrated from (2.35) with a second-order finite difference method for decreasing h as well as over time for a fixed $h = 10^{-3}$ (Figure 2.6). We compare the wall clock time for each method (Figure 2.7).

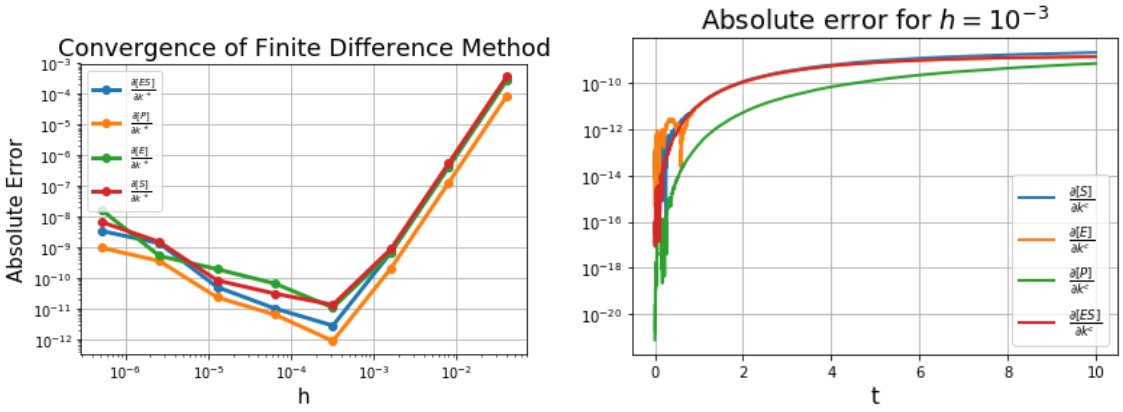

Figure 2.6: *(Left)* the average absolute error over time between Gronwall Gradients and the second-order finite difference approximation for the gradient of f with respect to parameters. *(Right)* the absolute error over time between Gronwall Gradients and the second-order finite difference approximation for the gradient of f with respect to parameters for $h = 10^{-3}$.

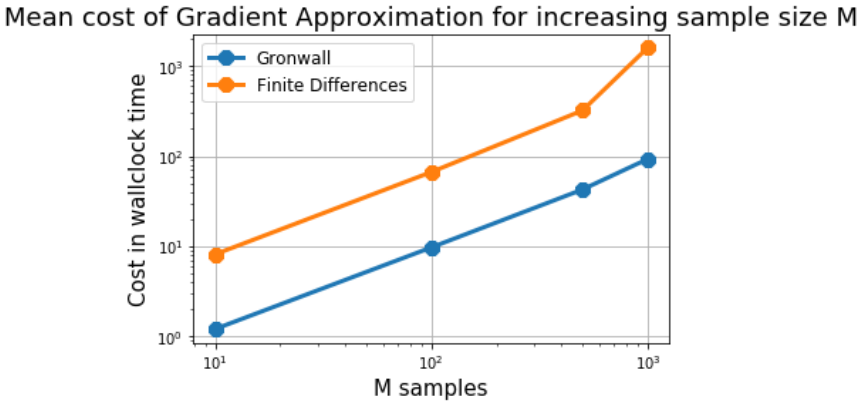

Figure 2.7: The wallclock time in computing M second-order finite difference approximations to the gradient versus integrating the augmented system in (2.35), averaged over 15 runs.

Given that, in this case, Gronwall Gradients are less expensive and comparably accurate to a second order finite difference method, we integrate the augmented system (2.35) to obtain $f(t, \mathbf{p})$ and $\nabla f(t, \mathbf{p})$.

We compute the first eigenvector of $\boldsymbol{C}$ from $t_0 = 0.0$ to $t_F = 35.0$ at 8000 points in

time by independently computing the active subspace at each time step. The time-series of this eigenvector (with signs normalized according to (2.28)) will be used as the "truth" to which we will compare all future approximations. For the following experiments we begin our approximations from $t_0 = 0.001$ because the initial condition of ∇f for Gronwall Gradients is $\nabla f = \mathbf{0}$.

2.3.3.1 Numerical Experiments

We again test the accuracy of DMD and SINDy in recovering the dynamic active subspace as a function of (i) the *proportion of time* the snapshots represent, and (ii) the *number of samples* in the data matrices.

In the following experiments we construct the data matrices $\boldsymbol{X_{DMD}}$ and $\boldsymbol{X_{SINDy}}$ with snapshots from proportions of the data from 0.1 to 1.0. For each proportion we construct the data matrices from 10, 20, 30, 40, 50, 75, 100, 200, 500, 1000, 2000, and 5000 snapshots. To determine the influence of α in the accuracy of the SINDy approximation of the dynamic active subspace, we test the absolute relative error and wall clock time for an approximation with 30 snapshots from 60% of time for varying the α parameter in `scikit-learn`'s `lasso` function.

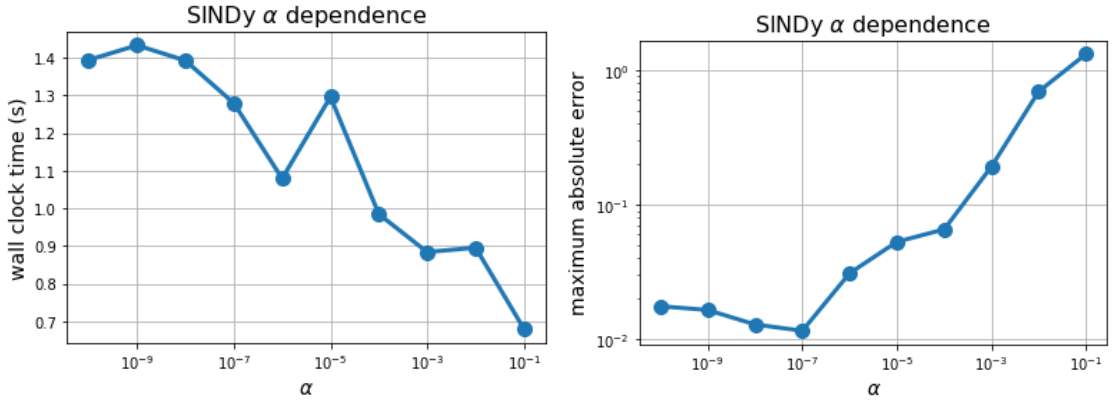

Figure 2.8: The influence of varying α on the *(left)* cost and *(right)* absolute error of the approximation of the dynamic active subspace with SINDy.

We determine that for these experiments $\alpha = 10^{-7}$ provides the most accurate approximation with not significantly more cost (see Figure 2.8). Thus in the implementation of SINDy we use `scikit-learn`'s `lasso` function with $\alpha = 10^{-7}$.

In order to construct a data matrix $\boldsymbol{X_{SINDy}}$ with N snapshots at times $t_0, t_1, \cdots, t_N$, we compute an additional N eigenvectors at $t_0 + h, t_1 + h, \cdots, t_N + h$ for $h = 10^{-6}$ with which to approximate $\dot{\boldsymbol{X}}$. In Figure 2.9, the error for each experiment is computed as the maximum absolute error between the approximated dynamic active subspace, and the "true" active subspace computed at all 8000 time steps. The library matrix $\boldsymbol{\Theta}$ is constructed with monomials of up to degree 5, trigonometric combinations of up to degree 3, and exponential combinations of up to degree 3.

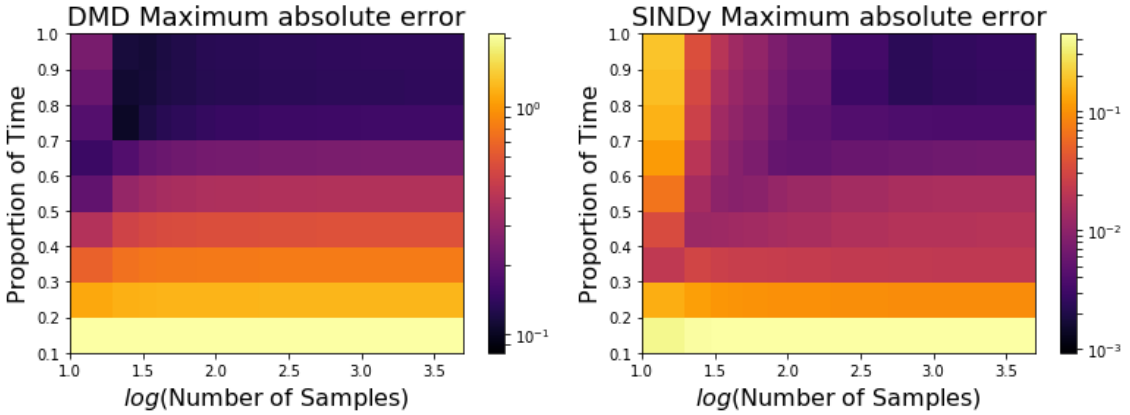

Figure 2.9: The maximum absolute error of the dynamic active subspace approximated by *(left)* DMD and *(right)* SINDy. The data matrices for each algorithm have been created with N samples from p proportion of the total time.

To understand the error in these approximations, we approximate the dynamic active subspace with 10 samples from the first 50% of time (Figure 2.10). Recall that although there are 10 samples in the $\boldsymbol{X_{SINDy}}$ data matrix, there were 20 active subspace computations.

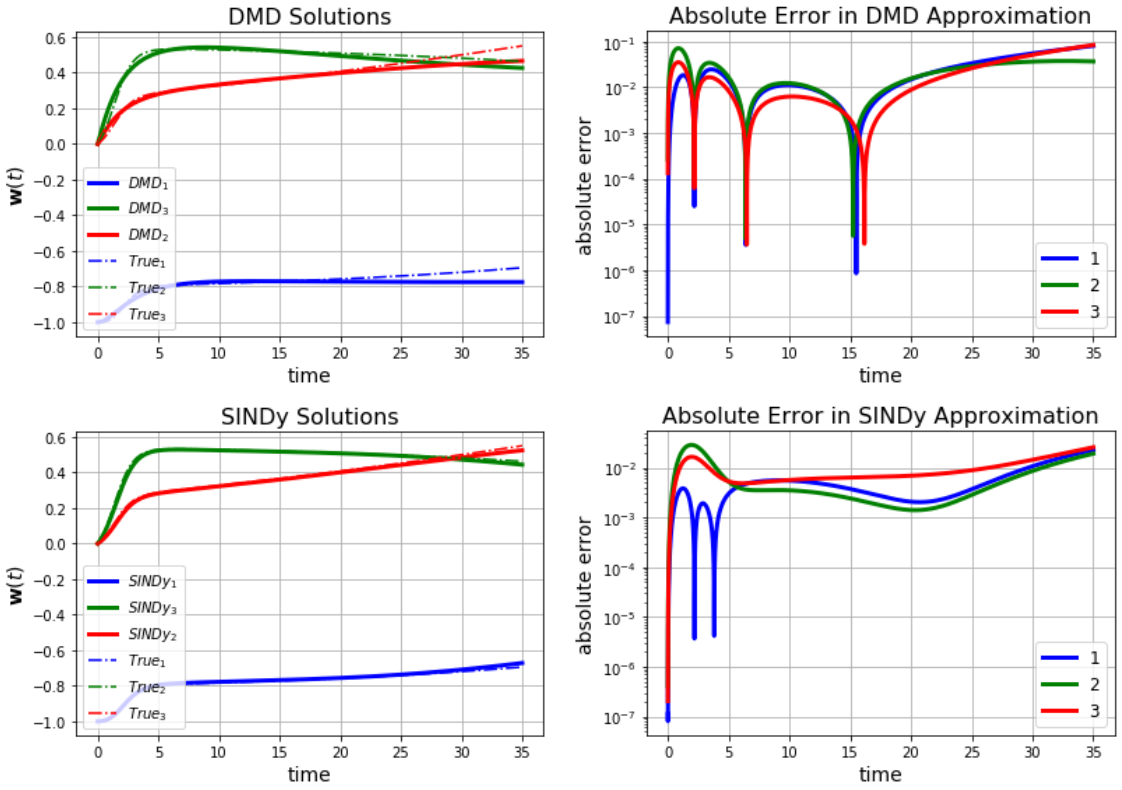

Figure 2.10: The *(left)* approximated dynamic active subspace and its *(right)* absolute error using *(top)* DMD and *(bottom)* SINDy. Solutions have been approximated with 10 samples from the first 50% of time.

2.4 DISCUSSION

Our methodology has shown to be effective in recovering the dynamic active subspace for (i) the linear case for which we also have the analytical form, and (ii) the nonlinear enzyme kinetics system.

In the linear case, Figure 2.4 shows that the dynamic active subspace can be recovered with DMD to within 10^{-8} from only ten samples from the first 10% of time. Conversely we see that SINDy is only ever able to recover the dynamic active subspace to within 10^{-2}. Given that it is necessary to compute the active subspace at twice as many points in time for the SINDy approximation, it is clear that, for this case, DMD recovers the dynamic active subspace for less cost and with more accuracy than SINDy.

In the nonlinear enzyme kinetics system Figure 2.9 shows that the dynamic active subspace can be recovered with SINDy to within 10^{-3} with 40 samples (80 including those used to approximate the derivative) from 60% of the total time. Conversely, we see that the approximation given by DMD is bounded below by 10^{-1}. This example given in Figure 2.10 allows us to visualize what the error in Figure 2.9 *means* in the context of approximating the dynamic active subspace. We see that, qualitatively, both the DMD and SINDy approximations recover the dynamic active subspace fairly well. However, we see that the approximation found with SINDy overall captures the dynamic active subspace better than DMD.

DMD so accurately recovered the dynamic active subspace for the linear case due to the fact that the true dynamic active subspace is indeed a linear dynamical system. Furthermore, DMD accurately recovered the dynamic active subspace for the snapshots that were in some sense *post-processed* when multiplied by the square root of the time-dependent first eigenvalue. Without this scalar multiplier, the snapshots in DMD would not be from a linear dynamical system. We only knew that this was, in some sense, the *correct* scalar multiplier because of the analysis in Section 2.2. In all other cases, this insight into what the dynamic active subspace *should* be or look like does not exist.

SINDy, however, has accurately recovered the dynamic active subspace for the nonlinear case. In this case, it was not necessary to have any prior knowledge of the dynamic active subspace in order for the methodology to recover the dynamics both qualitatively and quantitatively well from very little data. In future applications of this work to analyze and understand other systems, it is recommended that one use the SINDy approach to estimate the dynamic active subspace. In these applications it has shown to provide approximations to the one-dimensional dynamic active subspace from little computation and to within 10^{-3}. To fully assess the usefulness of this methodology on other systems, however, we must either find a mathematical guarantee of its ability to approximate the dynamics, or test it on bigger and more complex systems than those tested here.

2.5 CONCLUSIONS AND FUTURE WORK

In this thesis we have presented a novel approach for uncertainty quantification and sensitivity analysis in time-dependent systems. In the above sections we have discussed (i) the ideas and algorithms in finding the active subspace of a function, (ii) the use of Gronwall Gradients in approximating the gradient of a time-dependent function, and (iii) the methods of dynamic mode decomposition and sparse identification for nonlinear dynamical systems in recovering and approximating dynamical systems from data.

We have proven that, in the case where the QoI is a function of a linear dynamical system dependent on its initial conditions, or of an inhomogeneous linear dynamical system dependent on the inhomogeneous part of the system, there exists a dynamical system describing the dynamic active subspace.

For other time-dependent QoI, we have developed a methodology for approximating the dynamic active subspace given snapshots of the active subspace. We have approximated the dynamic active subspace using this methodology for (i) a linear harmonic oscillator and (ii) the enzyme kinetics system. We have quantitatively and qualitatively compared the recovered dynamic active subspace obtained with DMD and SINDy. We quantitatively assessed the tradeoff between the proportion of time represented in the data snapshots, and the number of snapshots, used in DMD and SINDy, and showed that for the linear harmonic oscillator, DMD was able to recover the dynamics with more accuracy and for less cost. Conversely, in the enzyme kinetics example, SINDy was able to recover the dynamics with more accuracy and for less cost.

The insight given from a time-dependent sensitivity metric allows for an in-depth understanding of the underlying system. Furthermore, the parameters in biological systems are often approximated with uncertainty resulting from measurement or observation error or parameter identifiability problems. Our methodology provides a means by which to identify the linear combination of parameters that will change the function the most on average, and

how the linear combination changes in time.

There is great potential for future work in line with this research. By properly ensuring orthogonality between states, the methodology presented in this work could be extended to recover the k-dimensional dynamic active subspace given by the first k eigenvectors. It is possible that the analytical form for the dynamical system of the dynamic active subspace exists for other time-dependent systems; further research in the overlap with differential geometry, classical dynamical systems analysis, and the study of manifolds could yield promising results in the study of dynamic active subspaces.

The concept of using data-driven recovery techniques (such as DMD and SINDy) to approximate a time-dependent sensitivity metric given snapshots of the sensitivity metric is, to the best of our knowledge, novel. Furthermore, the methodology presented in this work can be extended to develop time-dependent insight for other global sensitivity metrics. This intellectual and technical contribution of the novel ideas and methodology developed in this work lend to the advancement of uncertainty quantification and sensitivity analysis for time-dependent functions, with applications ranging from epidemiological studies, to chemical kinetics, to biological systems.

BIBLIOGRAPHY

[1] N. T. J. Bailey and J. Duppenthaler, Sensitivity analysis in the modelling of infectious disease dynamics, Journal of Mathematical Biology, 10 (1980), pp. 113–131.

[2] S. M. Blower and H. Dowlatabadi, Sensitivity and uncertainty analysis of complex models of disease transmission: An hiv model, as an example, International Statistical Review / Revue Internationale de Statistique, 62 (1994), pp. 229–243.

[3] N. Borisov, E. Aksamitiene, A. Kiyatkin, S. Legewie, J. Berkhout, T. Maiwald, N. Kaimachnikov, J. Timmer, J. Hoek, and B. Kholodenko, Systems-level interactions between insulin–egf networks amplify mitogenic signaling, Molecular Systems Biology, 5 (2009), pp. 1–15.

[4] S. L. Brunton, J. L. Proctor, and J. N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, Proceedings of the National Academy of Science, 113 (2016), pp. 3932–3937.

[5] E. Candes and J. Romberg, l1-magic: A collection of MATLAB routines for solving the convex optimization programs central to compressive sampling, www. acm. caltech. edu/l1magic, (2006).

[6] E. J. Candes and T. Tao, Decoding by linear programming, IEEE Transactions on Information Theory, 51 (2005), pp. 4203–4215.

[7] H. Caswell, Sensitivity analysis of transient population dynamics, Ecology Letters, 10 (2007), pp. 1–15.

[8] R. Chartrand, Numerical differentiation of noisy, nonsmooth data, International Scholarly Research Network Applied Mathematics, 2011 (2011), pp. 1–11.

[9] S. S. Chen, D. L. Donoho, and M. A. Saunders, Atomic decomposition by basis pursuit, SIAM Journal on Scientific Computing, (1999), pp. 33–61.

[10] N. Chitnis, J. M. Cushing, and J. M. Hyman, Bifurcation analysis of a mathematical model for malaria transmission, SIAM Journal on Applied Mathematics, 67 (2006), pp. 24–45.

[11] P. G. Constantine, Active Subspaces: Emerging Ideas for Dimension Reduction in Parameter Studies (SIAM Spotlights), SIAM-Society for Industrial and Applied Mathematics, 2015.

[12] P. G. Constantine, Z. del Rosario, and G. Iaccarino, Many physical laws are ridge functions, arXiv:1605.07974 [math.NA], (2016), pp. 1–20.

[13] P. G. Constantine and P. Diaz, Global sensitivity metrics from active subspaces, Reliability Engineering & System Safety, 162 (2017), pp. 1–13.

[14] P. G. Constantine and A. Doostan, Time-dependent global sensitivity analysis with active subspaces for a lithium ion battery model, Statistical Analysis and Data Mining, 10 (2017), pp. 243–262.

[15] R. Cooke, Experts in Uncertainty: Opinion and Subjective Probability in Science, Oxford University Press, 1991.

[16] J. Crutchfield and B. McNamara, Equations of motion from a data series, Complex Systems, 1 (1987), pp. 417–452.

[17] R. Cukier, C. Fortuin, K. Shuler, A. G. Petschek, and J. Schailby, Study of the sensitivity of the coupled reaction systems to uncertainties in rate coefficients: I. theory, Journal of Chemical Physics, 59 (1973), pp. 3873–3878.

[18] G. Dieter, Engineering design: a materials and processing approach, McGraw-Hill, 2 ed., 1991.

[19] M. Elad and M. Aharon, Image denoising via sparse and redundant representations over learned dictionaries, IEEE Transactions on Image Processing, (2006), pp. 3736–3745.

[20] H. Enderling and M. Chaplain, Mathematical modeling of tumor growth and treatment, Current Pharmaceutical Design, 20 (2014), pp. 1–7.

[21] T. Gronwall, Note on the derivatives with respect to a parameter of the solutions of a system of differential equations, Annals of Mathematics, 20 (1919), pp. 292–296.

[22] T. J. Hastie, R. J. Tibshirani, and J. H. Friedman, The elements of statistical learning, Springer Series in Statistics, New York, NY, 2 ed., 2009.

[23] B. Ioos and P. Lemaître, A review on global sensitivity analysis methods, 2015.

[24] I. Jolliffe, Principal Component Analysis, Springer Series in Statistics, New York, 1986.

[25] D. Jones, M. Plank, and B. Sleeman, Differential Equations and Mathematical Biology, Chapman and Hall/CRC, 2 ed., 2009.

[26] J. Keener and J. Sneyd, Mathematical Physiology (Interdisciplinary Applied Mathematics), Springer, 1998.

[27] J. Kim and H. Park, Fast active-set-type algorithms for l1- regularized linear regression, Proceedings of the International Conference on Artificial Intelligence and Statistics, 13 (2010), pp. 397–404.

[28] A. Kiparissides, S. S. Kucherenko, A. Mantalaris, and E. N. Pistikopoulos, Global sensitivity analysis challenges in biological systems modeling, Industrial and Engineering Chemistry Research, 48 (2009), pp. 7168–7180.

[29] P. Krishnaiah, Analysis of Variance, Elsevier: New York, 1981.

[30] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor, Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems, SIAM-Society for Industrial and Applied Mathematics, 2016.

[31] V. Kuznetsov, I. Makalkin, M. Taylor, and A. Perelson, Nonlinear dynamics of immunogenic tumors: Parameter estimation and global bifurcation analysis, Bulletin of Mathematical Biology, 56 (1994), pp. 295–321.

[32] R. LeVeque, Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems (Classics in Applied Mathematics), SIAM, Society for Industrial and Applied Mathematics, 2007.

[33] T. Loudon and S. Pankavich, Mathematical analysis and dynamic active subspaces for a long term model of HIV, Mathematical Biosciences & Engineering, 14 (2016), pp. 709–733.

[34] S. Marino, I. B. Hogue, C. J. Ray, and D. E. Kirschner, A methodology for performing global uncertainty and sensitivity analysis in systems biology, Journal of Theoretical Biology, 254 (2008), pp. 178–196.

[35] M. McKay, R. Beckman, and W. Conover, A comparison of three methods for selecting values of input variables in the analysis of output from a computer code, Technometrics, 21 (1979), pp. 239–245.

[36] M. Morris, Factorial sampling plans for preliminary computational experiments, Technometrics, 33 (1991), pp. 161–174.

[37] J. Newman and W. Tiedemann, Porous-electrode theory with battery applications, American Institute of Chemical Engineers Journal, 21 (1975), pp. 25–41.

[38] S. Pankavich and D. Shutt, An in-host model of HIV incorporating latent infection and viral mutation, arXiv:1508.07616 [q-bio.PE], (2015).

[39] K. Pearson, Notes on regression and inheritance in the case of two parents, Proceedings of the Royal Society of London, 58 (1895), pp. 240–242.

[40] R. Sachs, L. Hlatky, and P. Hahnfeldt, Simple ode models of tumor growth and anti-angiogenic or radiation treatment, Mathematical and Computer Modeling, 33 (2001), pp. 1297–1305.

[41] A. Saltelli, D. Gatelli, F. Campolongo, J. Cariboni, M. Ratto, M. Saisana, and T. Andres, Global Sensitivity Analysis: The Primer, John Wiley & Sons, 2008.

[42] H. Schaeffer, S. Osher, R. Caflisch, and C. Hauck, Sparse dynamics for partial differential equations, Proceedings of the National Academy of Sciences, 110 (2013), pp. 6634–6639.

[43] M. Schilling, T. Maiwald, S. Hengl, D. Winter, C. Kreutz, W. Kolch, W. Lehmann, J. Timmer, and U. Klingmüller, Theoretical and experimental analysis links isoform-specific erk signalling to cell fate decisions, Molecular Systems Biology, 5 (2009), pp. 1–18.

[44] P. Schmid, Dynamic mode decomposition of numerical and experimental data, Journal of Fluid Mechanics, 656 (2010), pp. 5–28.

[45] M. Schmidt and H. Lipson, Distilling free-form natural laws from experimental data, Science, 324 (2009), pp. 81–85.

[46] J. Schmitz, N. V. Riel, K. Nicolay, P. Hilbers, and J. Jeneson, Silencing of glycolysis in muscle: experimental observation and numerical analysis, Experimental Physiology, 95 (2010), pp. 380–397.

[47] S. Tavener, M. Mikucki, S. G. Field, and M. F. Antolin, Transient sensitivity analysis for nonlinear population models, Methods in Ecology and Evolution, 2 (2011), pp. 560–575.

[48] R. Tibshirani, Regression shrinkage and selection via the lasso, Journal of the Royal Statistical Society, (1996), pp. 267–288.

[49] G. Tran and R. Ward, Exact recovery of chaotic systems from highly corrupted data, arXiv:1607.01067 [math.DS], (2016).

[50] J. Tu, C. Rowley, D. Luchtenburg, S. Brunton, and J. Kutz, On dynamic mode decomposition: Theory and applications, Journal of Computational Dynamics, 1 (2014), pp. 391–421.

[51] T. Turányi, Sensitivity analysis of complex kinetic systems. tools and applications, Journal of Mathematical Chemistry, 5 (1990), pp. 203–248.

[52] H. Voss, P. Kolodner, M. Abel, and J. Kurths, Amplitude equations from spatiotemporal binary-fluid convection data, Physical review letters, 83 (1999), pp. 3422–3425.