

REDUCED-ORDER MODELS OF TRANSPORT PHENOMENA

A DISSERTATION  
SUBMITTED TO THE DEPARTMENT OF  
ENERGY RESOURCES ENGINEERING  
AND THE COMMITTEE ON GRADUATE STUDIES  
OF STANFORD UNIVERSITY  
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR THE DEGREE OF  
DOCTOR OF PHILOSOPHY

Hannah Lu

May 2022

© Copyright by Hannah Lu 2022  
All Rights Reserved

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

---

(Daniel M. Tartakovsky) Principal Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

---

(Hamdi Tchelepi)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

---

(Louis Durlofsky)

Approved for the Stanford University Committee on Graduate Studies



# Abstract

Quantitative models of transport phenomena play a significant role in understanding and optimizing many energy-, environment-, and medicine-related processes. Despite remarkable advances in algorithm development and computer architecture, fine-resolution/high-fidelity simulations remain a challenging and often unfeasible task due to the nonlinear nature of coupled transport phenomena, the complexity and heterogeneity of ambient environments, and the concomitant lack of sufficient data needed to parameterize the models. The computational demands can be prohibitive, especially in optimization, control and uncertainty quantification problems, where thousands of simulations need to be run.

Reduced-order models (ROMs) have been developed to obtain “cheap” yet accurate surrogates of high-fidelity models. The goal is to alleviate the expensive computational costs, while simultaneously capturing the underlying dynamic features. This dissertation addresses several challenges in construction of conventional ROMs for flow and transport problems, and introduces a physics-aware dynamic mode decomposition (DMD) framework to ameliorate the shortcomings of conventional ROMs. This framework supplements DMD, a data-driven tool that uses best linear approximations to construct efficient ROMs for complex systems, with physics-aware ingredients.

The first part of this dissertation presents a study on prediction accuracy of DMD. While multiple numerical experiments demonstrated the power and efficiency of DMD in representing data (i.e., in the interpolation mode), applications of DMD as a predictive tool (i.e., in the extrapolation mode) are scarce. This is due, in part, to the lack of rigorous error estimators for DMD-based predictions. We derive a theoretical error estimator for DMD extrapolation of numerical solutions, which allows one to

monitor and control the errors associated with DMD-based ROMs approximating the physics-based models.

In the second part of this dissertation, we demonstrate the shortcoming of conventional DMD methods (formulated within the Eulerian framework) for transport problems, in which sharp fronts play a dominant role in the dynamical systems. We propose a Lagrangian-based DMD method to overcome this so-called *translational issues*. This Lagrangian framework is valid only for smooth solutions, before a shock forms. After the shock formation, characteristic lines cross each other and the projection from the high-fidelity model (HFM) space to the low-fidelity model (LFM) space severely distorts the moving grid, resulting in numerical instabilities.

In the third part of this dissertation, we address this grid distortion issue in ROMs of conservation laws with shock features. Then, we propose a shock-preserving DMD method based on a nonlinear hodograph transformation that relies on the conservation law at hand to recover a low-rank structure and overcome the numerical instability.

Finally, we propose an extended dynamic mode decomposition (xDMD) approach to cope with the potentially unknown sources/sinks in inhomogeneous partial differential equations (PDEs). Our xDMD incorporates two new features, residual learning and bias identification, which are inspired by similar ideas in deep neural networks. Our theoretical error analysis demonstrates the guaranteed higher-order accuracy of xDMD relative to standard DMD.

Our approaches exemplify the spirit of physics-aware DMD since they account for the evolution of characteristic lines, the information about rarefactions/shocks and the awareness of potential model inhomogeneity. The resulting ROMs are capable of capturing key features of the underlying dynamics with higher-order accuracy than conventional DMD. They do so at a small fraction of the computational time of the iteration-based methods (e.g., proper orthogonal decomposition), which explains its rapid adoption by engineers in a plethora of applications.

# Acknowledgements

First and foremost, I would like to express my sincere gratitude to my advisor, Prof. Daniel M. Tartakovsky, who offered me everything that one could hope to find in an extraordinary advisor. I am eternally grateful for his inspiration, encouragement and guidance throughout my academic and professional pursuits. The life-changing opportunity of doing a PhD with Daniel is one of the most memorable things in my life. Over the years, he has provided me with the freedom to follow my passions, explore research ideas and collaborate with others. This dissertation would not have been possible without the unconditional support and inspirational mentoring from Daniel. I want to thank him for being such a role model for me - I hope to follow in his footsteps and fulfill my potential as a researcher in the future.

Besides my advisor, I would like to thank the rest of my committee, Prof. Hamdi Tchelepi, Louis Durlofsky, Charbel Farhat and Eric Darve for their valuable inputs on this work and for their guidance throughout my PhD and beyond. I am fortunate to benefit from their inspiring classes, fruitful discussions and insightful conversations.

My gratitude goes to all my collaborators and those who offered tremendous amount of help in my research. It has been truly rewarding and inspiring to work with them. I would like to thank the entire Stanford ERE faculty and staff for their dedication to a professional work environment as well as a supportive community. In addition, I want to acknowledge Stanford EDGE fellowship for the travel funding and Lawrence Berkeley National Laboratory for the research funding during my PhD study.

Furthermore, I would like to thank my officemates, labmates and my friends at Stanford, too many to list, for the joyful group dinners, for the girls' nights out and

team building, for the in-depth conversations about dreams, love and life, and for the companies along this unforgettable journey. I want to thank my lifelong friend Stephanie Wan, who has been like a sister to me, for always being there for me, no matter what.

Last, and most importantly, I want to thank my mother Yu Liu and my father Dr. Manping Lu, who give me unconditional love, undoubted faith and endless support to pursue my dreams. This dissertation is dedicated to the memory of my grandfather Chuangui Liu, who named me after a Chinese epic poem: “人生自古谁无死，留取丹心照汗青” . I hope he is proud of me when he looks down from heaven.

# Contents

<b>Abstract</b>	<b>v</b>
<b>Acknowledgements</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Literature Review . . . . .	3
1.1.1 POD-based ROMs . . . . .	5
1.1.2 DMD-based ROMs . . . . .	7
1.2 Challenges in Transport Phenomena . . . . .	15
1.3 Dissertation Outline . . . . .	17
<b>2 Prediction Accuracy of DMD</b>	<b>20</b>
2.1 Analysis of Predictive Accuracy . . . . .	22
2.1.1 Preliminaries . . . . .	22
2.1.2 Main Results . . . . .	23
2.1.3 Application to Nonlinear Parabolic Problems . . . . .	27
2.2 Numerical Tests of Predictive Accuracy . . . . .	29
2.2.1 Linear Diffusion (Test 1) . . . . .	30
2.2.2 Reaction-Diffusion Equation (Test 2) . . . . .	35
2.2.3 Nonlinear Reaction-Diffusion Equation (Test 3) . . . . .	41
2.2.4 Nonlinear Schrödinger Equation (Test 4) . . . . .	44
2.3 Summary . . . . .	46

<b>3</b>	<b>Lagrangian DMD for Advection-Dominated Phenomena</b>	<b>48</b>
3.1	Challenge Posed by Translational Problems . . . . .	51
3.2	Lagrangian Reduced-Order Models . . . . .	54
3.2.1	Lagrangian POD . . . . .	55
3.2.2	Lagrangian DMD . . . . .	57
3.3	Numerical Tests of Lagrangian DMD . . . . .	57
3.3.1	Linear Advection Equation . . . . .	58
3.3.2	One-Dimensional Advection-Diffusion Equation . . . . .	60
3.3.3	Two-Dimensional Advection-Diffusion Equation . . . . .	61
3.3.4	Inviscid Burgers' Equation . . . . .	63
3.3.5	Viscous Burgers' Equation . . . . .	65
3.3.6	Computational Costs . . . . .	65
3.4	Summary . . . . .	67
<b>4</b>	<b>Hodograph DMD for Hyperbolic Problems with Shocks</b>	<b>69</b>
4.1	ROM Failure for Problems with Shocks . . . . .	71
4.2	Hodograph Transformation . . . . .	74
4.2.1	Solution Before Shock Formation . . . . .	76
4.2.2	Solution After Shock Formation . . . . .	77
4.2.3	Summary of Hodograph Solution . . . . .	78
4.3	Physics-Aware DMD for Conservation Laws with Shocks . . . . .	79
4.4	Numerical Tests of Hodograph DMD . . . . .	80
4.4.1	Riemann Problem for Burgers' Equation with Shock . . . . .	81
4.4.2	Riemann Problem for Burgers' Equation with Rarefaction Wave	83
4.4.3	Smooth Solution of Riemann Problem for Burgers' Equation with Non-monotonic Initial Data . . . . .	85
4.4.4	Riemann Problem for Burgers' Equation with Rarefaction and Shock . . . . .	86
4.4.5	Riemann Problem for Buckley-Leverett's Equation . . . . .	88
4.4.6	Riemann Problem for Euler Equations . . . . .	90
4.5	Summary . . . . .	95

<b>5</b>	<b>Extended DMD for Inhomogeneous Problems</b>	<b>97</b>
5.1	Problem Formulation and Extended DMD . . . . .	99
5.1.1	Standard DMD . . . . .	100
5.1.2	Generalized DMD . . . . .	101
5.1.3	Residual DMD . . . . .	102
5.1.4	Extended DMD . . . . .	102
5.2	Relative Performance of Different DMD Formulations . . . . .	103
5.3	Numerical Tests of Extended DMD . . . . .	106
5.3.1	Inhomogeneous PDEs . . . . .	108
5.3.2	Inhomogeneous Boundary Conditions and Data Errors . . . . .	112
5.3.3	Coupled Nonlinear PDEs . . . . .	114
5.3.4	Generalizability to New Inputs . . . . .	119
5.4	Summary . . . . .	124
<b>6</b>	<b>Conclusion and Future Work</b>	<b>126</b>
<b>A</b>	<b>Level-set DMD for Hyperbolic Conservation Laws</b>	<b>129</b>
<b>B</b>	<b>Scalar Conservation Laws with Convex Fluxes</b>	<b>131</b>
B.1	Solution Before Shock Formation . . . . .	131
B.2	Solution After Shock Formation . . . . .	132
B.3	Summary of Hodograph Solution . . . . .	133
<b>C</b>	<b>Additional Numerical Tests of Extended DMD</b>	<b>135</b>
C.1	Boundary Conditions and Noisy Data . . . . .	135
C.2	Two-dimensional Viscous Burgers' Equation . . . . .	138
C.3	One-dimensional Advection-Diffusion Equation . . . . .	139
C.4	Two-dimensional Advection-Diffusion Equation . . . . .	141
C.5	Transport in Heterogeneous Media: Generalizability to New Inputs . . . . .	147
	<b>Bibliography</b>	<b>149</b>

# List of Tables

3.1	Computational time (sec) of the HFMs and the corresponding Lagrangian DMD- and POD-based ROMs. . . . .	67
5.1	Computational time (sec) and relative error for the reference solution and the DMD and xDMD models. . . . .	119

# List of Figures

1.1	Schematic illustrating the Koopman operator [1]. . . . .	13
2.1	Test 1a. Reference solution (left) and its DMD approximation with $M = 200$ snapshots (right). . . . .	30
2.2	Test 1a. Top: Local truncation error for DMD with $M = 100, 200$ and 300 snapshots; Middle: global error for DMD and our error bound estimator with $M = 100$ ; Bottom: global error for DMD and our error bound estimator with $M = 200$ snapshots. The global error is negligible for $M = 300$ (not shown). The rank threshold is set to $\varepsilon = 10^{-8}$ . . . . .	31
2.3	Test 1a. Top: Local truncation error for DMD with $M = 100, 200$ and 300 snapshots; Bottom: global error for DMD and our error bound estimator with $M = 100$ snapshots. The global error is negligible for $M = 200$ and 300 (not shown). The rank threshold is set to $\varepsilon = 10^{-12}$ . . . . .	32
2.4	Test 1b. Reference solution (left) and its DMD approximation with $M = 200$ snapshots (right). . . . .	33
2.5	Test 1b. Top: Local truncation error for DMD with $M = 100, 200$ and 300 snapshots; Middle: global error for DMD and our error bound estimator with $M = 100$ ; Bottom: global error for DMD and our error bound estimator with $M = 200$ snapshots. The global error is negligible for $M = 300$ (not shown). The rank threshold is set to $\varepsilon = 10^{-8}$ . . . . .	34

2.6	Test 2a. Fully resolved solution $u(x, t)$ of the reaction-diffusion problem (2.36) in the diffusion-dominated regime, and its approximations obtained from $M = 200$ snapshots with DMD (with two sets of observables $\mathbf{g}$ ) and POD-DEIM. . . . .	36
2.7	Test 2a. Local truncation error; Comparison of POD and DMD errors of the solution; Global error (errors of the observables) for DMD prediction with observables $\mathbf{g}_1$ and $\mathbf{g}_2$ . All ROMs are constructed using $M = 200$ snapshots. . . . .	37
2.8	Test 2b. Fully resolved solution $u(x, t)$ of the reaction-diffusion problem (2.36) in the reaction-dominated regime, and its approximations obtained from $M = 200$ snapshots with DMD (with two sets of observables $\mathbf{g}$ ) and POD-DEIM. . . . .	38
2.9	Test 2b. Local truncation error; Comparison of POD and DMD errors of the solution; Global error (errors of the observables) for DMD prediction with observables $\mathbf{g}_1$ and $\mathbf{g}_2$ using $M = 200$ snapshots. . . .	39
2.10	Test 2b. Computational times of the fully resolved solution, POD-DEIM, and DMD with the observables $\mathbf{g}_2(\mathbf{u})$ (left table); Comparison of POD and DMD in terms of computational time and accuracy (right figure). . . . .	40
2.11	Test 3. Fully resolved solution $u(x, t)$ of the nonlinear reaction-diffusion problem (2.37) and its approximations obtained from $M = 200$ snapshots with DMD (with two sets of observables $\mathbf{g}$ ) and POD-DEIM. . .	42
2.12	Test 3. Comparison of POD and DMD errors; Local truncation error and global error for DMD prediction with $\mathbf{g}_1$ and $\mathbf{g}_2$ using $M = 200$ snapshots. . . . .	43
2.13	Test 4. Resolved solution, DMD solutions and POD solution using $M = 20$ snapshots; Comparison of POD and DMD errors. . . . .	44
2.14	Test 4. Comparison of POD and DMD errors; Local truncation error and global error for DMD prediction with $\mathbf{g}_1$ and $\mathbf{g}_2$ using $M = 20$ snapshots. . . . .	45

3.1	Solution profiles $u(x, \cdot)$ , for several times $t$ , in the diffusion-dominated regime. These profiles are computed with DMD (left) and POD (right), and compared with the reference solution. . . . .	52
3.2	Solution profiles $u(x, \cdot)$ , for several times $t$ , in the advection-dominated regime. These profiles are computed with DMD (left column) and POD (right column) using the SVD rank of $r = 20$ (top row) and $r = 30$ (bottom row), and compared with the reference solution. . . . .	53
3.3	Three of the dominant DMD modes (left column) and POD modes (right column) extracted from the first 250 snapshots. . . . .	54
3.4	Solutions of the linear advection equation, $u(x, t)$ , alternatively obtained with the numerical method (3.2) and the ROMs constructed via Lagrangian DMD (left) and Lagrangian POD (right). . . . .	59
3.5	Errors of the Lagrangian DMD- and POD-based ROMs for the linear advection equation: error in reconstructing the state variable $u(x, t)$ (left) and its observables $\mathbf{g}(\mathbf{u})$ (right). The error bound for $\mathbf{g}(\mathbf{u})$ is derived in [2]. . . . .	59
3.6	Solutions of the linear advection-diffusion equation, $u(x, t)$ , alternatively obtained with the numerical method (3.2) and the ROMs constructed via Lagrangian DMD (left) and Lagrangian POD (right). . . . .	60
3.7	Errors of the Lagrangian DMD- and POD-based ROMs for the linear advection equation: error in reconstructing the state variable $u(x, t)$ (left) and its observables $\mathbf{g}(\mathbf{u})$ (right). The error bound for $\mathbf{g}(\mathbf{u})$ is derived in [2]. . . . .	61
3.8	Solutions of the two-dimensional linear advection-diffusion equation, $u(x, y, t)$ , obtained alternatively with the numerical method (3.2) and the ROMs constructed via Lagrangian DMD (left bottom) and Lagrangian POD (right bottom). . . . .	62
3.9	Errors of the Lagrangian DMD- and POD-based ROMs for the two-dimensional advection-diffusion equation: error in reconstructing the state variable $u(x, y, t)$ (left) and its observables $\mathbf{g}(\mathbf{u})$ (right). The error bound for $\mathbf{g}(\mathbf{u})$ is derived in [2]. . . . .	63

3.10	Solutions of the inviscid Burgers' equation, $u(x, t)$ , alternatively obtained with the numerical method (3.2) and the ROMs constructed via Lagrangian DMD (top) and Lagrangian POD (bottom). . . . .	64
3.11	Errors of the Lagrangian DMD- and POD-based ROMs for the inviscid Burgers' equation: error in reconstructing the state variable $u(x, t)$ (left) and its observables $\mathbf{g}(\mathbf{u})$ (right). The error bound for $\mathbf{g}(\mathbf{u})$ is derived in [2]. . . . .	65
3.12	Solutions of the viscous Burgers' equation, $u(x, t)$ , alternatively obtained with the numerical method (3.2) and the ROMs constructed via Lagrangian DMD (top) and Lagrangian POD (bottom). . . . .	66
3.13	Errors of the Lagrangian DMD- and POD-based ROMs for the viscous Burgers' equation: error in reconstructing the state variable $u(x, t)$ (left) and its observables $\mathbf{g}(\mathbf{u})$ (right). The error bound for $\mathbf{g}(\mathbf{u})$ is derived in [2]. . . . .	67
4.1	Solutions of the inviscid Burgers equation with a shock. <b>(a)</b> Top: the full solution obtained with the Lagrangian numerical scheme (4.4) leading to the overshoot. <b>(b)</b> Bottom: the Lagrangian DMD solution trained on a few snapshots of the faulty full solution. The reference solution is obtained with (3.2). . . . .	73
4.2	Solutions of the inviscid Burgers equation with a shock. <b>(a)</b> Top: the full solution obtained with the appropriate Lagrangian numerical scheme (4.5). <b>(b)</b> Bottom: the Lagrangian DMD solution trained on a few snapshots of the accurate full solution. The reference solution is obtained with (3.2). . . . .	75
4.3	Physics-aware DMD solution of the inviscid Burgers equation with a shock. The reference solution is given by analytic solution (4.16). 1st order upwind scheme by (3.2) is also plotted here in solid line. . . . .	83

4.4	Physics-aware DMD solution of the inviscid Burgers equation with a rarefaction wave. The reference solution is given by analytic solution (4.20). 1st order upwind scheme by (3.2) is also plotted here in solid line. . . . .	84
4.5	Physics-aware DMD solution of the inviscid Burgers equation subject to nonmonotonic initial condition. The reference solution is computed with (3.2). . . . .	86
4.6	Physics-aware DMD solution of the inviscid Burgers equation with a rarefaction wave and shock. The reference solution is computed with (3.2). . . . .	88
4.7	Physics-aware DMD solution of the Buckley-Leverett equation, which has a nonmonotonic flux function. The reference solution is computed with (3.2). . . . .	89
4.8	The analytical solution to the Sod shock tube problem (4.28): (a) density, (b) pressure, (c) velocity, (d) energy; all evaluated at time $t = 0.25$ . . . . .	91
4.9	ROMs constructed by (a) top: the standard DMD and (b) bottom: the physics-aware DMD. The former uses snapshots of $\rho(x, t)$ in the $(x, \rho)$ plane, while the latter relies on snapshots of $x(\rho, t)$ in the $(\rho, x)$ plane. . . . .	92
4.10	Prediction errors of the ROMs constructed by the physics-aware DMD with different parameters. The top row shows the model-order reduction error $e_x$ defined in (4.30); the total DMD error $e_\rho$ , defined in (4.31), is shown in the bottom row. . . . .	94
5.1	Reference solution of (5.31) and its DMD (5.6) and xDMD (5.16) approximations (left column); the log relative error of the DMD (5.6), gDMD (5.10), rDMD (5.12), and xDMD (5.16) models (right column). . . . .	109
5.2	Reference solution of (5.31) and its DMD (5.6) and xDMD (5.16) approximations (left); the log relative error of the DMD (5.6), gDMD (5.10), rDMD (5.12), and xDMD (5.16) models (right) for long time extrapolation. . . . .	110

5.3	Estimation of the source term $S(x) = x$ and $e^x$ in (5.31) by gDMD and xDMD: the eyeball measure (left) and the log relative error (right).	111
5.4	Multi-connected simulation domain $\mathcal{D}$ (left) and the mesh used in the finite-element solution of (5.32).	112
5.5	The reference solution of (5.31), $u(x, y, t)$ , at times $t = 2000$ , $t = 5000$ and $t = 10000$ .	113
5.6	Dependence of the log relative error of the DMD and xDMD models on time in the representation, extrapolation and interpolation modes. These errors are reported for noiseless data (top row) and data corrupted by addition of zero-mean white noise whose strengths at any $(x, t)$ is 0.1% of the nominal value of $u(x, t)$ at that point (bottom row).	114
5.7	Velocity magnitude $U = \sqrt{u^2 + v^2}$ of incompressible flow with the Reynolds number $\text{Re} = 60$ around an impermeable circle predicted by solving numerically the two-dimensional Navier-Stokes equations (5.33) (top row) and by using the DMD and xDMD models in the representation mode. The representation errors (5.30) for these two approximations are displayed in the second and third rows, respectively.	116
5.8	Velocity magnitude $U = \sqrt{u^2 + v^2}$ of incompressible flow with the Reynolds number $\text{Re} = 60$ around an impermeable circle predicted by solving numerically the two-dimensional Navier-Stokes equations (5.33) (top row) and by using the DMD and xDMD models in the extrapolation mode. The extrapolation errors (5.30) for these two approximations are displayed in the second and third rows, respectively.	118
5.9	Temporal evolution of log relative error of the DMD and xDMD models in the representation and extrapolation modes.	119
5.10	Spatial distribution of hydraulic conductivity $K_c(\mathbf{x})$ used in our simulations.	120
5.11	Representative snapshots of solute concentration, $u(\mathbf{x}, t)$ , in the training dataset with initial condition (5.36).	121
5.12	Representative snapshots of solute concentration, $u(\mathbf{x}, t)$ , in the test case with initial condition (5.38).	122

5.13	Solute concentration predicted with the DMD and xDMD models for the initial condition not seen during training. Also shown are the absolute errors of DMD and xDMD. . . . .	123
5.14	Solute concentration predicted with the DMD and xDMD models using 0.1% noisy data for the initial condition not seen during training. Also shown are the absolute errors of DMD and xDMD. . . . .	124
A.1	Solutions of the inviscid Burgers equation, $u(x, t)$ , alternatively obtained with the numerical method (3.2) and the ROM constructed via the level-set DMD. . . . .	130
C.1	Solution accuracy in Cases 1 (left column), 2 (middle column) and 3 (right column) in the representation (top row), extrapolation (middle row) and interpolation (bottom row) regimes. . . . .	137
C.2	Temporal snapshots of the solution, $u(\mathbf{x}, t) = v(\mathbf{x}, t)$ , to the 2D viscous Burgers' equation. . . . .	138
C.3	Temporal evolution of the log relative errors of the DMD and xDMD solutions, $\mathbf{u}(\mathbf{x}, t) = (u, v)^\top$ , to the viscous Burgers' equation in the interpolation regime. . . . .	139
C.4	DMD and xDMD solutions to (C.6) on the training data. Left: the DMD and xDMD solution profiles compared with the reference solution at different times; Right: temporal evolution of the log relative error. . . . .	140
C.5	DMD and xDMD solutions to (C.6) on the test data. Left: the DMD and xDMD solution profiles compared with the reference solution at different times; Right: temporal evolution of the log relative error. . . . .	141
C.6	Test 1: The ground truth at the end of simulation time (top row) and its DMD and xDMD approximations (middle row), accompanied by the corresponding absolute error maps (bottom row). . . . .	144
C.7	Test 2: The ground truth at the end of simulation time (top row) and its DMD and xDMD approximations (middle row), accompanied by the corresponding absolute error maps (bottom row). . . . .	145

C.8	Test 3: The ground truth at the end of simulation time (top row) and its DMD and xDMD approximations (middle row), accompanied by the corresponding absolute error maps (bottom row). . . . .	146
C.9	Section 4.4: The ground truth at the end of simulation time (top row) and its DMD and xDMD approximations (middle row), accompanied by the corresponding absolute error maps (bottom row). . . . .	148

# Chapter 1

## Introduction

Predictive capabilities are critical for sustainable utilization of subsurface environment, improving the access to energy resources, and developing cost-effective and environmentally safe operations. Examples of such predictive modeling include oil and gas production [3, 4], geological CO<sub>2</sub> sequestration [5–7], geothermal reservoirs [8], nuclear waste disposal [9, 10], and subsurface contamination and remediation [11]. A recent focus has been to incorporate multi-scale complex systems with models informed by the ongoing deluge of scientific data. The developments in data-driven modeling have been driven by the urgent needs within the scientific and engineering communities. They aim to approximate the solution of the high-fidelity model with significant speedup in applications that include, but are not limited to, surrogate modeling for future prediction, digital twins for real-time visualization, and fast uncertainty quantification for model predictions.

Model reduction techniques can significantly reduce the (prohibitively) high computational cost of physics-based simulations, while capturing key features of the underlying dynamics. Such techniques have been used extensively in subsurface applications [12–18] and can be grouped in two general classes. The first is physics-based ROMs, which seek to map a high-dimensional model onto a meaningful representation of reduced dimensionality; in this context, dimensionality refers to the number of degrees of freedom in a discretized numerical model. A prime example of this class is proper orthogonal decomposition (POD) [19–21], which is grounded in singular value

decomposition (SVD). It obtains a ROM by projecting the dynamics of the full model onto the hyperplane using the basis extracted from the SVD analysis. The computational saving stems from replacing the high-dimensional full nonlinear system with its lower-dimensional counterpart for future prediction. Such ROMs are physics-based in the sense that they inherit the dynamic operator from the projection.

The second class of ROMs are surrogates. Instead of reducing a model’s dimensionality, these methods aim to reduce its complexity by learning the dynamics of the state variables or quantities of interest directly from the full model’s output and/or observational data. These data-informed and equation-free ROMs are built by using such machine learning techniques as DMD [22, 23], Gaussian process regression (GPR) [24, 25], random forest (RF) [26, 27], and neural networks (NN) [28, 29].

In this dissertation, we contribute to this second class by introducing a physics-aware DMD framework, which combines the popular data-driven tool DMD with physics-aware ingredients. It addresses the following challenging questions in conventional ROMs:

- How accurate is the DMD-based ROM in prediction (i.e., in extrapolation regime)?
- How to tackle the *translational issue* for SVD-based methods?
- How to preserve physical characteristics and honor conservations laws in the constructed ROM?
- How to handle the nonlinearity and inhomogeneity from the HFM and enhance the generalizability of the LFM?

We further discuss the above issues in details and propose the corresponding solutions to ameliorate the difficulties in Chapter 2-5 respectively. The physics-aware DMD framework summarized in this dissertation originates from four published papers [2, 30–32]. It has been rapidly adopted by engineers in fast predictions for geopotential fields [33], real-time control for robotic systems [34] and for wind farms [35], modeling for pulsatile blood flow [36].

## 1.1 Literature Review

We review POD-based ROMs and DMD-based ROMs as representative approaches in the two general classes of ROMs respectively. SVD is widely utilized to extract a low-dimensional structure from the data and thus serving as a fundamental tool to construct ROMs from data generated with HFMs. While both POD and DMD are based on SVD, they provide two independent approaches to constructing ROMs. POD-based ROMs are built by deploying a projection (e.g., Galerkin projection) to map a HFM onto a much smaller subspace. The efficiency and accuracy of POD in nonlinear setting are increased by combining it with either the empirical interpolation method (EIM) [37] or discrete empirical interpolation method (DEIM) [38]. DMD is used to discover a spatiotemporal coherent structure in the HMF-generated data. DMD's connection to the Koopman operator theory of nonlinear dynamic systems [39] is of theoretical interest [40–43], while its *equation-free* spirit facilitates its use in conjunction with machine learning techniques [44, 45].

First, we provide a problem setup. Consider a real-valued state variable  $u(\mathbf{x}, t)$ , whose dynamics is described by a boundary-value problem

$$\begin{cases} \frac{\partial u}{\partial t} = f(u, \mathbf{x}), & (\mathbf{x}, t) \in \mathcal{D} \times \mathbb{R}^+; \\ \mathcal{B}(u) = b(\mathbf{x}), & (\mathbf{x}, t) \in \partial\mathcal{D} \times \mathbb{R}^+; \\ u(\mathbf{x}, 0) = u_0(\mathbf{x}), & \mathbf{x} \in \mathcal{D}. \end{cases} \quad (1.1)$$

Here,  $t$  denotes time;  $\mathbf{x}$  is the spatial coordinate;  $\mathcal{D} \subset \mathbb{R}^d$  is the simulation domain bounded by the surface  $\partial\mathcal{D}$ ;  $f$  is a (linear or nonlinear) operator that involves spatial derivatives and sources/sinks functions;  $\mathcal{B}$  is the boundary differential operator describing Dirichlet, Neumann, and/or Robin boundary conditions;  $b(\mathbf{x})$  represents boundary functions and  $u_0(\mathbf{x})$  is the initial state.

The simulation domain is discretized with a mesh consisting of  $N_{\mathbf{x}}$  elements ( $N_{\mathbf{x}} \gg 1$ ). A suitable numerical approximation of (1.1) yields a system of (coupled,

linear/nonlinear, high-dimensional) ODEs,

$$\begin{cases} \frac{d\mathbf{u}}{dt} = \mathbf{f}(\mathbf{u}; \mathbf{s}), & \mathbf{u}, \mathbf{s} \in \mathbb{R}^{N_{\mathbf{x}}}, \\ \mathbf{u}(0) = \mathbf{u}^0, & \mathbf{u}^0 \in \mathbb{R}^{N_{\mathbf{x}}}, \end{cases} \quad (1.2)$$

where  $\mathbf{s}$  comes from potential sources/sinks in  $f(u, \mathbf{x})$  and boundary  $b(\mathbf{x})$ . Let  $\mathcal{F}_{\Delta t} : \mathbb{R}^{N_{\mathbf{x}}} \rightarrow \mathbb{R}^{N_{\mathbf{x}}}$  denote a flow map, which relates the discretized system state  $\mathbf{u}$  at time  $t = t_0$  to that at time  $t = t_0 + \Delta t$ , where  $\Delta t$  is a (sufficiently small) time increment. There exists a flow map  $\mathcal{F}$ , depending only on the time difference  $t - t_0$ , which represents the solution to (1.2) as  $\mathbf{u}(t; \mathbf{u}^0, t_0, \mathbf{s}) = \mathcal{F}_{t-t_0}(\mathbf{u}^0; \mathbf{s})$ . Denote  $\mathbf{u}^n = \mathbf{u}(n\Delta t)$ , then the discrete-time flow map  $\mathcal{F}_{\Delta t}$  gives the evolution of (1.2):

$$\mathbf{u}^{n+1} = \mathcal{F}_{\Delta t}(\mathbf{u}^n; \mathbf{s}). \quad (1.3)$$

Our goal is to learn the dynamic system  $\mathbf{f}$ , or, more precisely, its reduced-order surrogate, using  $M$  temporal snapshots of the numerical solutions. Let the numerical solution  $\mathbf{u}^n \approx \mathbf{u}(t_n)$  with  $n = 1, 2, \dots$ , where the time lag between the input and output states,  $\Delta t$ , is assumed to be independent of  $n$  for the sake of convenience. In practice,  $\mathbf{u}^{n+1}$  is solved by an iterative numerical scheme

$$\mathbf{R}(\mathbf{u}^{n+1}) = \mathbf{u}^{n+1} - \mathbf{u}^n + \Delta t \mathbf{F}(\mathbf{u}^{n+1}, \mathbf{u}^n) = 0, \quad (1.4)$$

where  $\mathbf{F}$  represents appropriate discrete approximations of the derivatives and  $\mathbf{R}$  is the vectorized residual of the scheme. Certain CFL condition needs to be satisfied to ensure the stability of the scheme depending on the functional forms of  $f$  in (1.1). Simulation results obtained with the above method constitute our HFM. The discrepancy between the ROM and PDE (1.1) stems from two sources: the error between the ROM and the HFM (1.4) (numerical solution of certain order) and the error between the HFM (1.4) and the PDE (1.1) (or its equivalent representation via flow map (1.3)). Since we focus on the accuracy of the ROM compared to the corresponding HFM, we used stable numerical scheme (specified in corresponding numerical tests) as the HFM reference. One could improve the accuracy of the HFM by using a higher-order

numerical scheme or obtaining an analytical solution.

A reduced-order, low-fidelity model (ROM) is constructed from a data set comprising a sequence of solution snapshots collected from the HFM (1.4). Let  $\mathbf{X}$  denote the data matrix, consisting of  $M$  snapshots of  $\mathbf{u}$ ,

$$\mathbf{X} = \begin{bmatrix} | & | & \dots & | \\ \mathbf{u}^0 & \mathbf{u}^1 & \dots & \mathbf{u}^{M-1} \\ | & | & & | \end{bmatrix}, \quad \mathbf{X} \in \mathbb{R}^{N_{\mathbf{x}} \times M}. \quad (1.5)$$

Two alternative strategies for building a ROM from these data, both grounded in SVD, are described below.

### 1.1.1 POD-based ROMs

POD was first proposed in [46] to identify the coherent structure in dynamical systems. It is closely related to the principle component analysis (PCA) and the Karhuen-Loève transform. POD-based ROMs have been widely applied in linear subsurface flow problems [47] and nonlinear reservoir simulation problems [6, 37, 48].

The conventional POD method generates a ROM by using a low-dimensional basis extracted from the data  $\mathbf{X}$  in (1.5) to project the dynamics  $\mathbf{u}(t)$  onto a lower-dimensional hyperplane. If the data matrix  $\mathbf{X} \in \mathbb{R}^{N_{\mathbf{x}} \times M}$  has rank  $N_{\text{data}} \leq \min\{N_{\mathbf{x}}, M\}$ , then the POD modes are constructed by using a reduced SVD,

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*, \quad (1.6)$$

where  $\mathbf{U} \in \mathbb{C}^{N_{\mathbf{x}} \times N_{\text{data}}}$  is the matrix of  $N_{\text{data}}$  orthonormal columns of length  $N_{\mathbf{x}}$ ;  $\mathbf{\Sigma} = \mathbb{R}^{N_{\text{data}} \times N_{\text{data}}}$  is the diagonal matrix with real diagonal elements  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{N_{\text{data}}} > 0$ ;  $\mathbf{V} \in \mathbb{C}^{M \times N_{\text{data}}}$  is the matrix of  $N_{\text{data}}$  orthonormal columns of length  $M$ ; and the superscript  $*$  denotes its conjugate transpose. A ROM is constructed by choosing a rank  $r$  ( $r \ll N_{\text{data}}$ ), which satisfies a energy criteria, e.g.,

$$r = \min_k \left\{ \frac{\sigma_k}{\sum_{k=1}^{N_{\text{data}}} \sigma_k} < \varepsilon \right\}, \quad (1.7)$$

where  $\varepsilon$  is a user-supplied small number that determines the accuracy of the low-rank representation relative to the original  $\mathbf{X}$ . The rank truncation relies on the energy criteria to find the best low-rank representation of  $\mathbf{X}$  (see [49] for an in-depth analysis). The size of the snapshot matrix depends on the temporal correlation between snapshots, the dynamic of the governing equations and many other factors. Generally, a data matrix with enough snapshots is capable of capturing the dominant features of the underlying physical process. In (1.7), the rank truncation is only data-dependent, i.e., it depends only on  $\mathbf{X}$ . Next, the matrix  $\mathbf{U} \in \mathbb{C}^{N_x \times N_{\text{data}}}$  is replaced with a matrix  $\Phi \in \mathbb{C}^{N_x \times r}$  comprising  $r$  orthogonal columns of length  $N_x$ ,

$$\Phi = \begin{bmatrix} | & | & \cdots & | \\ \phi_1 & \phi_2 & \cdots & \phi_r \\ | & | & \cdots & | \end{bmatrix}. \quad (1.8)$$

The orthonormal vectors  $\{\phi_1, \dots, \phi_r\}$  form a POD basis. Finally, a ROM (low-fidelity solution) is constructed by the Galerkin projection of  $\mathbf{u}$  onto the low-dimensional space spanned by the POD basis (also known as trial basis),

$$\mathbf{u}_{\text{POD}}^n = \sum_{i=1}^r \hat{u}_i^n \phi_i = \Phi \hat{\mathbf{u}}^n, \quad \hat{\mathbf{u}}^n = [\hat{u}_1^n, \dots, \hat{u}_r^n]^\top. \quad (1.9)$$

Substituting (1.9) into (1.4) and projecting onto the low-dimensional space, yields an equations for the vector of coefficients  $\hat{\mathbf{u}}^n$ :

$$\Phi^\top \mathbf{R}(\Phi \hat{\mathbf{u}}^n) = 0. \quad (1.10)$$

Alternatively, one can use Petrov-Galerkin projection [50] and least-squares Petrov-Galerkin projection [51] by enforcing the residual to be orthogonal to a different low-dimensional test basis  $\tilde{\Phi}$ , i.e.,

$$\tilde{\Phi}^\top \mathbf{R}(\Phi \hat{\mathbf{u}}^n) = 0. \quad (1.11)$$

Time evolution still needs to be computed but only in a small subspace of the

original high-dimensional space. For linear problems, (1.10) or (1.11) can be solved efficiently in a subspace of dimension  $r$ . However, solving (1.10) or (1.11) in a nonlinear form scales with the dimension  $N_{\mathbf{x}}$  of the HFM. There are two general classes of nonlinear function approximations to eliminate the computational bottleneck for nonlinear model reduction. The first is to linearize the nonlinear function at multiple locations in the state space. A prime example of such linearization approaches is trajectory piecewise linear (TPWL) method. Applications of POD-TPWL-based ROMs in subsurface simulations include [3–5, 52, 53]. The second class is hyper-reduction approaches, such as empirical interpolation method (EIM) [38], discrete empirical interpolation method (DEIM) [37] and Gauss-Newton with approximated tensor (GNAT) [54]. The hyper-reduction approaches evaluate only a small subset of the entries in the nonlinear function  $f$  and reconstruct all other entries by interpolation strategies. POD-DEIM and POD-GNAT have been applied for model reduction in many reservoir simulation settings [55–57]. Although the aforementioned two classes of nonlinear model reduction techniques can achieve significant speedup relative to the full-order simulation, the accuracy of the resulting ROMs cannot be determined a priori. To the best of our knowledge, error estimates of POD-based nonlinear model reduction approaches are lacking unless the fully resolved solution is available or special assumptions are satisfied on the nonlinear terms.

### 1.1.2 DMD-based ROMs

DMD [58] has recently become a popular tool of data-driven regression. It belongs to the family of SVD and has its origins in representation of complex fluid flows in terms of their spatial modes and temporal frequencies [22]. This strategy for representation of spatiotemporal coherent structures has since been used for data diagnostics and related applications including video processing [23], interpretation of neural activity measurements [59], financial trading [60], and forecast of infectious disease spreading [61]. DMD with control has been developed to extract the input-output characteristics of dynamic systems with external control [62]. It has also been deployed to learn models of high-dimensional complex systems from data [63–65], in

the spirit of *equation-free* simulations [66].

DMD is connected to interpretation of nonlinear dynamical systems via the Koopman operator theory [41, 43]. The latter provides a bridge between finite-dimensional nonlinear dynamics and infinite-dimensional linear systems by observable functions [39]. Theoretical studies of the DMD approximation to eigenvalues and eigenfunctions of the infinite-dimensional Koopman operator show that the performance of this finite eigen-approximation depends crucially on the choice of observable functions, requiring expert prior knowledge of the underlying dynamics [43, 65]. Machine learning techniques have been used to select the observable by identifying relevant terms in the dynamics from data [44, 45, 67]. [65] employs regression from a dictionary of observables that spans a subspace of the space of scalar observables.

Numerical implementations of DMD are also undergoing modifications and enhancements. Under various assumptions on the data, many variants of the standard DMD algorithm were introduced to compute the eigenvalues and DMD modes in more accurate and efficient ways [68, 69]. Sparsity-promoting DMD and compressed DMD combine DMD with sparsity techniques and modern theory of compressed sensing [70, 71]. Inspired by the applications of DMD in video processing, multi-resolution DMD or mrDMD provides a means for recursive computation of DMD of separate spatiotemporal features at different scales in the data [72]. The mrDMD approach preserves the translational and rotational invariances, which remains the Achilles heel of many SVD-based methods [73].

In solving large linear systems, the DMD method aims to approximate the eigenvalues and eigenfunctions of  $\mathbf{f}$  in (1.2) and provides an alternative to POD. A major advantage of DMD over POD is its *equation-free* nature, which allows future-state predictions without any computation of further time evolution. For a nonlinear dynamic  $\mathbf{f}$  in (1.2), DMD seeks a finite-dimensional approximation of the infinite-dimensional Koopman operator of the nonlinear dynamics. With carefully chosen observables, a ROM can be constructed in the observable space with sufficient accuracy. We briefly review the standard DMD algorithm on state space and the related Koopman operator theory as follows.

Given the data matrix  $\mathbf{X}$  in (1.5), the DMD procedure constructs the proxy,

approximate locally linear dynamical system

$$\begin{cases} \frac{d\mathbf{u}}{dt} = \mathcal{K}\mathbf{u}, \\ \mathbf{u}(0) = \mathbf{u}^0. \end{cases} \quad (1.12)$$

The analytical solution to (1.12) is well-known [74]:

$$\begin{aligned} \mathbf{u}(t) &= \sum_{i=1}^{N_x} \phi_i \exp(\omega_i t) \hat{u}_i^0 = \Phi \exp(\Omega t) \hat{\mathbf{u}}^0, \\ \Phi &= \begin{bmatrix} | & | & & | \\ \phi_1 & \phi_2 & \cdots & \phi_r \\ | & | & & | \end{bmatrix}, \quad \Omega = \text{diag}(\omega_i), \\ \hat{\mathbf{u}}^0 &= [\hat{u}_1^0, \dots, \hat{u}_{N_x}^0]^\top = \Phi^\dagger \mathbf{u}^0, \end{aligned} \quad (1.13)$$

where  $\phi_i$  and  $\omega_i$  are the eigenvectors and eigenvalues of the matrix  $\mathcal{K}$ , and the coefficients  $\hat{u}_i^0$  are the coordinates of  $\mathbf{u}(0)$  in the eigenvector basis,  $^\dagger$  represents the Moore-Penrose pseudo-inverse.

Given continuous dynamics in (1.12), the analogous discrete-time system sampled every  $\Delta t$  in time can be described as follows:

$$\mathbf{u}^{n+1} = \mathbf{K}\mathbf{u}^n, \quad (1.14)$$

where  $\mathbf{K} = \exp(\mathcal{K}\Delta t)$ . The solution to this system (1.14) may be expressed in terms of the eigenvalues  $\lambda_i$  and eigenvectors  $\phi_i$  of the discrete-time map  $\mathbf{K}$ :

$$\mathbf{u}^n = \sum_{i=1}^{N_x} \phi_i (\lambda_i)^n \hat{u}_i^0 = \Phi (\Lambda)^n \hat{\mathbf{u}}^0, \quad \Lambda = \text{diag}(\lambda_i). \quad (1.15)$$

The DMD algorithm produces a low-rank eigen-decomposition (1.15) of the matrix  $\mathbf{K}$  that optimally fits the measured trajectory  $\mathbf{u}^k$  for  $k = 0, 1, \dots, M-1$  in a least-square

sense, i.e.,

$$\mathbf{K} = \underset{\hat{\mathbf{K}} \in \mathbb{R}^{N_{\mathbf{x}} \times N_{\mathbf{x}}}}{\operatorname{argmin}} \frac{1}{M-1} \sum_{k=0}^{M-2} \|\mathbf{u}^{k+1} - \hat{\mathbf{K}}\mathbf{u}^k\|_2^2, \quad (1.16)$$

where  $\|\cdot\|_2$  is the  $L_2$  norm defined as

$$\|\mathbf{u}\|_2 = \sqrt{\sum_{i=1}^{N_{\mathbf{x}}} u_i^2}, \quad \mathbf{u} = [u_1, \dots, u_{N_{\mathbf{x}}}]^\top. \quad (1.17)$$

The optimality of the approximation holds only over the sampling window where  $\mathbf{K}$  is constructed, and the approximation solution can be used to not only make future state predictions but also decompose the dynamics into various time scales, since the  $\lambda_i$  are prescribed.

The most modern definition of the DMD method was provided in [75]:

**Definition 1.1** (Dynamic mode decomposition). *Suppose we have a dynamical system (1.2) and two sets of data created from the  $M$  snapshots data matrix  $\mathbf{X}$  in (1.5):*

$$\mathbf{X}_1 = \begin{bmatrix} | & | & \dots & | \\ \mathbf{u}^0 & \mathbf{u}^1 & \dots & \mathbf{u}^{M-2} \\ | & | & & | \end{bmatrix} \quad \text{and} \quad \mathbf{X}_2 = \begin{bmatrix} | & | & \dots & | \\ \mathbf{u}^1 & \mathbf{u}^2 & \dots & \mathbf{u}^{M-1} \\ | & | & & | \end{bmatrix}, \quad (1.18)$$

where each  $\mathbf{u}^k, k = 0, 1, \dots, M-1$  is driven by (1.3). DMD computes the leading eigendecomposition of the best-fit linear operator  $\mathbf{K}$  relating the data  $\mathbf{X}_2 \approx \mathbf{K}\mathbf{X}_1$ :

$$\mathbf{K} = \mathbf{X}_2 \mathbf{X}_1^\dagger, \quad (1.19)$$

where  $\dagger$  is the Moore-Penrose pseudo-inverse. The DMD modes, also called dynamic modes, are the eigenvectors of  $\mathbf{K}$ , and each DMD mode corresponds to a particular eigenvalue of  $\mathbf{K}$ .

**Remark 1.1.** *The obtained  $\mathbf{K}$  is a solution to (1.16) and equivalently a minimizer to*

$\|\mathbf{X}_2 - \mathbf{K}\mathbf{X}_1\|_F$ , where  $\|\cdot\|_F$  is the Frobenius norm, given by

$$\|\mathbf{X}\|_F = \sqrt{\sum_i \sum_j X_{i,j}^2}. \quad (1.20)$$

In practice, when the state dimension  $N_x$  is large, the matrix  $\mathbf{K}$  may be intractable to analyze directly. Instead, DMD circumvents the eigen-decomposition of  $\mathbf{K}$  by considering a rank-reduced representation in terms of a POD-projected matrix  $\mathbf{K}_r$ . The algorithm proceeds as follows [58]:

---

**Algorithm 1:** Standard DMD algorithm

---

1. Apply SVD  $\mathbf{X}_1 \approx \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*$ , where  $\mathbf{U} \in \mathbb{C}^{N_x \times r}$  is a unitary matrix,  $\mathbf{\Sigma} \in \mathbb{C}^{r \times r}$  is a diagonal matrix with components  $\sigma_i \geq 0$  that are called singular values of  $\mathbf{X}_1$ ,  $\mathbf{V}^*$  is the conjugate transpose of unitary matrix  $\mathbf{V} \in \mathbb{C}^{r \times (M-1)}$ , and  $r$  is the truncated rank chosen by certain criteria (e.g., (1.7)).
  2. Compute  $\mathbf{K}_r = \mathbf{U}^*\mathbf{X}_2\mathbf{V}\mathbf{\Sigma}^{-1}$  as a low-rank ( $r \times r$ ) approximation of  $\mathbf{K}$ .
  3. Compute eigendecomposition of  $\mathbf{K}_r$ :  $\mathbf{K}_r\mathbf{W} = \mathbf{W}\mathbf{\Lambda}$ , where  $\mathbf{\Lambda} = \text{diag}(\lambda_i)$  are eigenvalues and columns of  $\mathbf{W}$  are the corresponding eigenvectors.
  4. Eigenvalues of  $\mathbf{K}$  can be approximated by  $\mathbf{\Lambda}$  with corresponding eigenvectors in the columns of  $\mathbf{\Phi} = \mathbf{U}\mathbf{W}$ .
- 

**Remark 1.2.** The modes  $\mathbf{\Phi} = \mathbf{U}\mathbf{W}$  are referred to as projected DMD modes [22]. Alternative formula is given by  $\mathbf{\Phi} = \mathbf{X}_2\mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{W}$  in [75], which is called exact DMD modes. These two will tend to converge if  $\mathbf{X}_1$  and  $\mathbf{X}_2$  have the same column spaces. The projected DMD formulation  $\mathbf{\Phi} = \mathbf{U}\mathbf{W}$  should be used unless a zero eigenvalue is associated with  $\mathbf{K}$  [75].

The low-rank matrix  $\mathbf{K}_r$  defines a low-dimensional linear ROM of the dynamical system on POD coordinates:

$$\begin{cases} \hat{\mathbf{u}}^{n+1} = \mathbf{K}_r \hat{\mathbf{u}}^n, \\ \hat{\mathbf{u}}^0 = \mathbf{\Phi}^\dagger \mathbf{u}^0. \end{cases} \quad (1.21)$$

The high-dimensional state  $\mathbf{u}^n$  can be reconstructed by

$$\mathbf{u}^n = \mathbf{\Phi} \hat{\mathbf{u}}^n. \quad (1.22)$$

With the low-rank approximations of both the eigenvalues and the eigenvectors in hand, the projected future solution can be constructed for all time in the future. By first rewriting for convenience  $\omega_i = \ln(\lambda_i)/\Delta t$ , the approximate solution at all future times is given by

$$\mathbf{u}_{\text{DMD}}(t) \approx \sum_{i=1}^r \phi_i \exp(\omega_i t) \hat{u}_i^0 = \Phi \exp(\Omega t) \hat{\mathbf{u}}^0, \quad \Omega = \text{diag}(\omega_i), \quad (1.23)$$

or in discrete-form for  $n$ -th time step ( $n \geq M$ )

$$\mathbf{u}_{\text{DMD}}^n = \Phi \Lambda^n \hat{\mathbf{u}}^0, \quad n \geq M, \quad (1.24)$$

Notice that no more iteration is needed in the prediction. The solution at any future time is approximated in (1.23) or (1.24) using only information encapsulated in the first  $M$  snapshots.

If the underlying dynamics  $\mathbf{f}$  in (1.2) is linear, then the DMD method recovers the leading eigenvalues and eigenvectors normally computed using standard solution methods for linear differential equations. For nonlinear problems, the DMD method approximates the modes of the so-called *Koopman operator* [41, 43, 58] defined as follows:

**Definition 1.2** (Koopman operator). *Consider a nonlinear dynamic system (1.2), where  $\mathbf{u} \in \mathcal{M}$  is a state on a smooth  $N_{\mathbf{x}}$ -dimensional manifold  $\mathcal{M}$ . The Koopman operator  $\mathcal{K}$  is an infinite-dimensional linear operator that acts on all observable functions  $g : \mathcal{M} \rightarrow \mathbb{C}$  so that*

$$\mathcal{K}g(\mathbf{u}) = g(\mathbf{f}(\mathbf{u})). \quad (1.25)$$

*For discrete dynamic system (1.3), the discrete-time Koopman operator  $\mathcal{K}_{\Delta t}$  is*

$$\mathcal{K}_{\Delta t}g(\mathbf{u}^n) = g(\mathcal{F}_{\Delta t}(\mathbf{u}^n)) = g(\mathbf{u}^{n+1}). \quad (1.26)$$

The Koopman operator transforms the finite-dimensional nonlinear problem (1.3) in the state space into the infinite-dimensional linear problem (1.26) in the observable space. The observable function  $g$  provides a bridge between the two equivalent

representations of the same fundamental behavior. Figure 1.1 shows the schematic illustration of the Koopman operator. Since  $\mathcal{K}$  is an infinite-dimensional linear operator, it is equipped with infinite eigenvalues  $\{\lambda_i\}_{i=1}^{\infty}$  and eigenfunctions  $\{\phi_i\}_{i=1}^{\infty}$ . In practice, one has to make a finite approximation of the eigenvalues and eigenfunctions. The following assumption is essential to both a finite-dimensional approximation and the choice of observables.

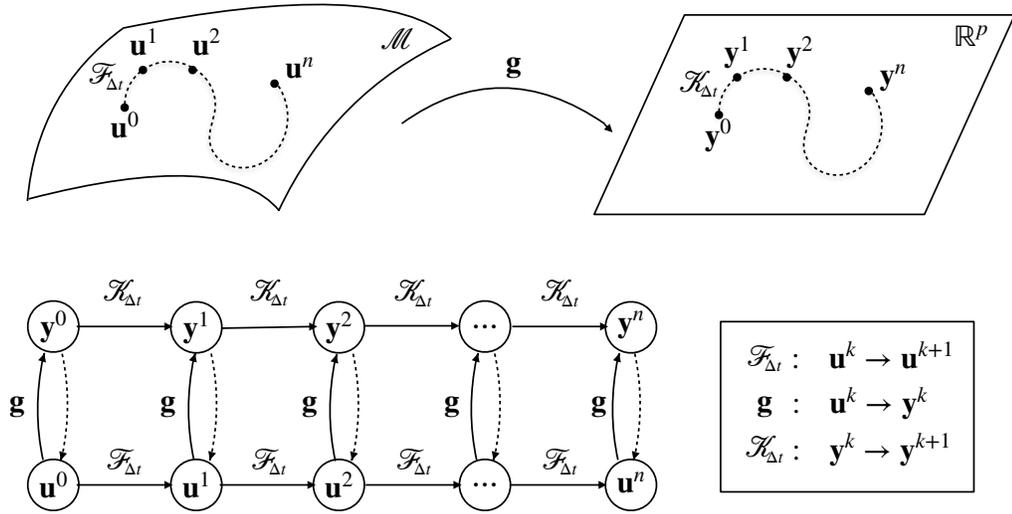


Figure 1.1: Schematic illustrating the Koopman operator [1].

**Assumption 1.1.** Let  $\mathbf{y}$  denote a  $p \times 1$  vector of observables,

$$\mathbf{y}^k = \mathbf{g}(\mathbf{u}^k) = [g_1(\mathbf{u}^k), \dots, g_p(\mathbf{u}^k)]^\top, \quad (1.27)$$

where  $g_j : \mathcal{M} \rightarrow \mathbb{C}$  is an observable function, with  $j = 1, \dots, p$ . If the chosen observable  $\mathbf{g}$  is restricted to an invariant subspace spanned by eigenfunctions of the Koopman operator  $\mathcal{K}_{\Delta t}$ , then it induces a linear operator  $\mathbf{K}$  that is finite-dimensional and advances these eigenobservable functions on this subspace [1].

Based on Assumption 1.1, the DMD algorithm in observable space is implemented in the following Algorithm 2 :

---

**Algorithm 2:** DMD algorithm on observable space [58]

---

0. Create the data matrices of observables

$$\mathbf{Y}_1 = \begin{bmatrix} | & | & & | \\ \mathbf{y}^0 & \mathbf{y}^1 & \dots & \mathbf{y}^{M-2} \\ | & | & & | \end{bmatrix} \quad \text{and} \quad \mathbf{Y}_2 = \begin{bmatrix} | & | & & | \\ \mathbf{y}^1 & \mathbf{y}^2 & \dots & \mathbf{y}^{M-1} \\ | & | & & | \end{bmatrix} \quad (1.28)$$

where each column is given by  $\mathbf{y}^k = \mathbf{g}(\mathbf{u}^k)$ .

1. Apply SVD  $\mathbf{Y}_1 \approx \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*$  with  $\mathbf{U} \in \mathbb{C}^{p \times r}$ ,  $\mathbf{\Sigma} \in \mathbb{C}^{r \times r}$ ,  $\mathbf{V} \in \mathbb{C}^{r \times (M-1)}$ , where  $r$  is the truncated rank chosen by certain criteria (e.g., (1.7)).
  2. Compute  $\mathbf{K}_r = \mathbf{U}^* \mathbf{Y}_2 \mathbf{V} \mathbf{\Sigma}^{-1}$  as a  $r \times r$  low-rank approximation for  $\mathbf{K}$ .
  3. Compute eigendecomposition of  $\mathbf{K}_r$ :  $\mathbf{K}_r \mathbf{W} = \mathbf{W} \mathbf{\Lambda}$ ,  $\mathbf{\Lambda} = \text{diag}(\lambda_i)$ .
  4. Reconstruct eigendecomposition of  $\mathbf{K}$ . Eigenvalues are  $\mathbf{\Lambda}$  and eigenvectors are  $\mathbf{\Phi} = \mathbf{U} \mathbf{W}$ .
- 

The future state in the space of observables is then given by

$$\mathbf{y}_{\text{DMD}}(t) = \mathbf{\Phi} \exp(\mathbf{\Omega}t) \hat{\mathbf{y}}^0, \quad \mathbf{\Omega} = \text{diag}(\ln(\lambda_i)/\Delta t), \quad \hat{\mathbf{y}}^0 = \mathbf{\Phi}^\dagger \mathbf{y}^0. \quad (1.29)$$

or in discrete form

$$\mathbf{y}_{\text{DMD}}^n = \mathbf{\Phi} \mathbf{\Lambda}^n \hat{\mathbf{y}}^0 \quad \text{for } n \geq M. \quad (1.30)$$

The future prediction in the state space can be obtained by the inverse transform  $\mathbf{g}^{-1}$ :

$$\mathbf{u}_{\text{DMD}} = \mathbf{g}^{-1}(\mathbf{y}_{\text{DMD}}). \quad (1.31)$$

It is important to note that the Koopman operator does not rely on linearization of the dynamics but instead represents the flow map of the dynamical system on observable functions as an infinite-dimensional operator. The approximation of the Koopman operator is at the heart of the DMD methodology. However, the quality of any finite-dimensional approximation to the Koopman operator depends on the observable function  $\mathbf{y} = \mathbf{g}(\mathbf{u})$ . Connections between the DMD theory and the Koopman spectral analysis under specific conditions on the observables and collected data are established by a theorem in [75]. This theorem indicates that judicious selection of the observables is critical to the success of the Koopman method. In general, there

is no principled way to select observables without expert knowledge of a dynamical system. Machine learning techniques can be deployed to identify relevant terms in the dynamics from data, which guide selection of the observables [44, 45, 67]. The key contribution of this dissertation in this field includes constructing a physics-aware DMD framework, where the observable functions are carefully designed by knowledge of the underlying physics so that accurate low-dimensional approximations to the Koopman operator can be obtained.

## 1.2 Challenges in Transport Phenomena

Transport phenomena is used to describe processes in which mass, momentum, angular momentum and energy move about in matter. Thus it includes diffusional phenomena, fluid dynamics and heat transfer. The multi-scale and multi-physics nature is the key to understanding and managing the behavior of transport phenomena in natural and artificial heterogeneous porous materials. This behavior dominates multiple phenomena of practical significance in the planet, such as subsurface storage of radioactive waste or carbon dioxide, sustainable exploitation of groundwater resources, and design of novel materials for electrical storage or desalination membranes. Most, if not all, of these applications rely on highly nonlinear models that strive to capture processes that occur on multiplicity of scales both in space and time. These multi-scale models are not only computationally expensive, but also have to cope with unknown physics and unknown parameters. The increasing accessibility of data collected from environmental monitoring, with developments of data-driven modeling, allows us to discover details about these unknowns, establish causal relations and bridge the scales.

There have been arising research interests in replacing the expensive PDE-based models with data-driven surrogate models. In particular, the following challenges are critical for model reduction in transport phenomena:

- One major concern is the lack of universal prediction accuracy analysis and

quantitative error estimators for data-driven modeling. Existing accuracy studies of ROMs are confined to projection-based POD methods for linear problems [76]. The nonlinear nature of most transport phenomena calls for the needs of efficient model reduction techniques as well as associated rigorous error analysis in a broader context of ROMs. With a quantitative error bound, a ROM can provide a means for accelerating computationally expensive Monte Carlo and multi-scale simulations in uncertainty quantification, real-time control and optimal designs.

- A well-reported issue with conventional linear-subspace ROMs (including standard POD and DMD) is that they fail to model advection-dominated problems [77–79], e.g., wave-like phenomena, advection-dominated flows, moving interfaces and shock propagation. This so called *translational problems* typically exhibits a slow decay of the singular values [73, 80, 81] (i.e., the Kolmogorov  $n$ -width is high [82]) and results in high-dimensional features, which means that no low-dimensional space exists in which the HFM solutions can be approximated well. High-dimensional ROMs not only render inefficient from a computational perspective but also lead to numerical instabilities. Literature on addressing this issue can be roughly distinguished into three categories. The first primarily focused on developing special reduced basis [78] that are better suited for advection-dominated problems to obtain stable approximations [83–87]. The second category recovers the low-rank structure of advection-dominated problems by transforms of the HFMs, e.g., transport maps [88, 89]. A typical numerical algorithm of this category is shifted POD [81, 90], which recovers the shift due to the advection and applies POD after having reversed the shift. The third category exploits the locally (in temporal space) low-dimensional structure of advection-dominated problems and constructs dynamic low-rank approximation [91–97] that adapts low-rank basis updates over time. The aforementioned literature is mostly projection-based and relies on substantial additional knowledge about the problem, such as the particular advection phenomena governing

basis shifting. More recently, data-driven nonintrusive methods use various machine learning techniques in attempts to resolve the translational challenge for model reduction on linear subspaces [98–101] or nonlinear manifolds [102–109] of smaller dimension. However, computationally efficient hyper-reduction for highly-nonlinear transport and shock-preserving model reduction for hyperbolic conservation laws in general remain challenging problems.

- In many practical problems, the inhomogeneity arising from source terms and boundary conditions plays a key role in the underlying transport dynamics. Conventional data-driven methods, such as standard DMD, lump together the differential operators and the inhomogeneous sources/boundary conditions in the underlying dynamics. This issue was addressed by [110] in the context of neural networks and also discussed by [62] in the context of DMD-control problems. Without the awareness of the potential inhomogeneity, the resulting ROM may provide inaccurate predictions and ambiguous interpretations of the complex systems.

### 1.3 Dissertation Outline

The above challenges will be further addressed in the following chapters of this dissertation. In Chapter 2, we provide a theoretical error estimator for DMD extrapolation of numerical solutions to linear and nonlinear flow problems. This error analysis allows one to monitor and control the errors associated with DMD-based temporal extrapolation of numerical solutions to parabolic differential equations. We use several computational experiments to verify the robustness of our error estimators and to compare the predictive ability of DMD with that of POD. Our analysis demonstrates the importance of a proper selection of observables, as predicted by the Koopman operator theory. In all the tests considered, DMD outperformed POD in terms of efficiency due to its iteration-free feature. In some of these experiments, POD proved to be more accurate than DMD. This suggests that DMD is preferable for obtaining a fast prediction with slightly lower accuracy, while POD should be used if the accuracy

is paramount. The work in Chapter 2 was published in *SIAM Journal on Scientific Computing* [2] in 2020.

In Chapter 3, the shortcoming of conventional POD and DMD methods formulated within the Eulerian framework for advection-dominated problems is investigated. Motivated by the recent work on Lagrangian POD [111], we propose a physics-aware DMD method to construct a ROM within the Lagrangian framework. We choose the temporally evolving characteristic lines, a crucial physical quantity in advection-dominated systems, as a key observable of the underlying Koopman operator. Then, the DMD algorithm is used to identify, from sufficient data, a low-dimensional structure in the Lagrangian framework and thus to construct a physics-aware ROM by approximating the underlying Koopman operator. The Lagrangian DMD can be applied to general advection-diffusion nonlinear flows. Furthermore, DMD outperforms POD in terms of computational costs because it is iteration free. Using our error analysis in [2], one can also estimate the accuracy of the ROM. The work in Chapter 3 was published in *Journal of Computational Physics* [30] in 2020.

In Chapter 4, we address the numerical instability issue caused by grid distortion in the presence of strong shocks and/or sharp gradients in Lagrangian POD and Lagrangian DMD methods. We resolve this outstanding issue in construction of ROMs for PDEs with discontinuous solutions and shocks by developing a physics-aware DMD method based on hodograph transformation. The latter provides a map between the original nonlinear system and its linear counterpart, which coincides with the Koopman operator. This strategy is consistent with the spirit of physics-aware DMDs in that it retains information about shock dynamics. The work in Chapter 4 was published in *Journal of Machine Learning for Modeling and Computing* [31] in 2021.

In Chapter 5, an extended dynamic mode decomposition (xDMD) approach is proposed to cope with the potential unknown inhomogeneity in PDEs. Motivated by similar ideas in deep neural networks, we equip our xDMD with two new features. First, it has a bias term, which accounts for inhomogeneity of PDEs and/or boundary conditions. Second, instead of learning a flow map, xDMD learns the residual increment by subtracting the identity operator. We also provide a theoretical error

analysis to illustrate the improved accuracy of xDMD compared to standard DMD. The work in Chapter 5 was published in *Journal of Computational Physics* [32] in 2021.

In Chapter 6, we conclude this dissertation with a summary, an overview of ongoing work and suggestions for future research directions. Level-set DMD for hyperbolic conservation laws is presented in Appendix A. A detailed construction of Hodograph transform for scalar conservation laws with convex fluxes is reviewed in Appendix B. Additional simulation results with xDMD are presented in Appendix C.

## Chapter 2

# Prediction Accuracy of DMD

DMD and POD are two complementary SVD techniques that are widely used to construct ROMs in a variety of fields of science and engineering. They are widely used as surrogates to predict future states of a system. DMD relies on the linear dynamical model learned entirely from data. Eigen-decomposition of the learned linear operator allows one to make predictions of DMD-based ROMs in an *iteration-free* fashion. For POD, time integration is still needed to compute future states, but only in a low-dimensional surrogate model. Thus, the computational cost is reduced and future states are predicted using the ROM derived from projecting the dynamics of the full system onto the hyperplane that the POD extracts from data. While both POD and DMD are based on SVD, they provide two independent approaches to constructing ROMs. It is therefore worthwhile to compare their relative performance in terms of accuracy and efficiency. Advantages of hybridizing the two methods have been demonstrated in several numerical tests [112, 113].

The convergence of DMD predictions are reported in [114] from the numerical perspective and in [42, 115, 116] from the theoretic perspective. A goal of our analysis is to provide a theoretical error estimator for DMD extrapolation of numerical solutions to linear and nonlinear parabolic equations. We are aware of no other quantitative analysis of the accuracy of DMD predictions. This error analysis allows one to monitor and control the errors associated with DMD-based temporal extrapolation of numerical solutions to parabolic differential equations.

We consider the following group of parabolic PDEs belonging to the general boundary-value problem (1.1):

$$\begin{cases} \frac{\partial u}{\partial t} = f_d(u) + f_s(u), & (\mathbf{x}, t) \in \mathcal{D} \times \mathbb{R}^+; \\ \mathcal{B}(u) = 0, & (\mathbf{x}, t) \in \partial\mathcal{D} \times \mathbb{R}^+; \\ u(\mathbf{x}, 0) = u_0(\mathbf{x}), & \mathbf{x} \in \mathcal{D}. \end{cases} \quad (2.1)$$

where  $f_d$  is a linear or nonlinear differential operator representing the internal dynamics in  $d$ -dimensional space, and the linear or nonlinear source term  $f_s$  represents the external source/sink into the system. Discretization of the simulation domain  $\mathcal{D}$  into  $N_{\mathbf{x}}$  elements or nodes ( $N_{\mathbf{x}} \gg 1$ ) transforms the PDE (2.1) into either a high-dimensional linear dynamical system

$$\frac{d\mathbf{u}}{dt} = \mathcal{A}\mathbf{u} + \mathbf{f}_s, \quad (2.2)$$

or a high-dimensional nonlinear dynamical system

$$\frac{d\mathbf{u}}{dt} = \mathbf{f}_d(\mathbf{u}) + \mathbf{f}_s(\mathbf{u}), \quad (2.3)$$

where  $\mathbf{u} = [u(\mathbf{x}_1, t), \dots, u(\mathbf{x}_{N_{\mathbf{x}}}, t)]^\top$  is the spatial discretization of  $u(\mathbf{x}, t)$ ;  $\mathcal{A}$  and  $\mathbf{f}_d$  are linear and nonlinear differential operators on  $\mathbb{R}^{N_{\mathbf{x}}}$ , respectively; and  $\mathbf{f}_s$  represents the correspondingly discretized reaction term  $f_s$ .

Low-dimensional ROMs are often used to reduce the computational costs of solving the high-dimensional systems (2.2) and (2.3). For example, POD has been deployed to construct accurate and efficient ROMs for (2.2) [19, 20]. Time evolution of  $u(\mathbf{x}, t)$  needs to be computed but only in a small subspace of the original high-dimensional space. For nonlinear systems (2.3), construction of a right ROM using POD becomes more challenging and requires some modifications, such as empirical interpolation method (EIM) [38] and discrete empirical interpolation method (DEIM) [37], whose accuracy cannot be determined a priori. To the best of our knowledge, error estimates of POD-EIM/DEIM are lacking unless the fully resolved solution is available.

The DMD method aims to approximate the eigenvalues and eigenfunctions of  $\mathbf{A}$  in (2.2) and provides an alternative to POD in solving large linear systems. A major advantage of DMD over POD is its *equation-free* nature, which allows future-state predictions without any computation of further time evolution. For the nonlinear problems (2.3), DMD seeks a finite-dimensional approximation of the infinite-dimensional Koopman operator of the nonlinear dynamics. With carefully chosen observables, a ROM can be constructed in the observable space with sufficient accuracy. We have briefly reviewed DMD and the related Koopman operator theory in section 1.1.2 as a set up for the accuracy analysis in the following section 2.1.

## 2.1 Analysis of Predictive Accuracy

We start with linear parabolic equations. Temporal discretization of (2.2) with time step  $\Delta t$  yields

$$\mathbf{u}^{n+1} = \mathbf{A}\mathbf{u}^n + \Delta t \mathbf{f}_s^{n+1/2}, \quad n \geq 0, \quad (2.4)$$

where  $\mathbf{A}$  is an  $N_{\mathbf{x}} \times N_{\mathbf{x}}$  matrix and  $\mathbf{f}_s^{n+1/2}$  is, e.g., a linear interpolation of  $\mathbf{f}_s^n$  and  $\mathbf{f}_s^{n+1}$ . The fully resolved model (2.4) is advanced by  $M$  time steps and the resulting temporal snapshots of  $\mathbf{u}(t)$  is recorded in the data matrix (1.5). We use a resolved accurate solution of (2.4) under a certain CFL condition as a reference or yardstick against which to test the accuracy of the DMD prediction (1.24).

### 2.1.1 Preliminaries

We start with a brief summary of the key results relevant to our subsequent analysis.

**Lemma 2.1.** *For parabolic PDEs, denote the spectral radius of  $\mathbf{A}$  in (2.4) by  $\rho(\mathbf{A})$  and assume  $\rho(\mathbf{A}) < 1$ . Then a stable numerical method of (2.4) satisfies the maximum*

principle in the discrete setting, i.e.,

$$\begin{aligned}
\|\mathbf{u}^{n+1}\|_2^2 &< \|\mathbf{u}^n\|_2^2 + \Delta t \max\{\|\mathbf{f}_s^n\|_2^2, \|\mathbf{f}_s^{n+1}\|_2^2\} \\
&< \dots \\
&< \|\mathbf{u}^0\|_2^2 + \Delta t \sum_{k=0}^n \max\{\|\mathbf{f}_s^k\|_2^2, \|\mathbf{f}_s^{k+1}\|_2^2\}.
\end{aligned} \tag{2.5}$$

*Proof.*

$$\begin{aligned}
\|\mathbf{u}^{n+1}\|_2^2 &\leq \|\mathbf{A}\mathbf{u}^n\|_2^2 + \Delta t \|\mathbf{f}_s^{n+1/2}\|_2^2 \\
&\leq \|\mathbf{A}\|_2^2 \|\mathbf{u}^n\|_2^2 + \Delta t \|\mathbf{f}_s^{n+1/2}\|_2^2 \\
&\leq \rho(\mathbf{A})^2 \|\mathbf{u}^n\|_2^2 + \Delta t \max\{\|\mathbf{f}_s^n\|_2^2, \|\mathbf{f}_s^{n+1}\|_2^2\}
\end{aligned} \tag{2.6}$$

According to the assumption  $\rho(\mathbf{A}) < 1$ ,

$$\|\mathbf{u}^{n+1}\|_2^2 < \|\mathbf{u}^n\|_2^2 + \Delta t \max\{\|\mathbf{f}_s^n\|_2^2, \|\mathbf{f}_s^{n+1}\|_2^2\}, \tag{2.7}$$

and Lemma 2.1 holds.  $\square$

**Lemma 2.2.** *DMD on  $M$  temporal snapshots is designed such that  $\|\mathbf{u}^{M-1} - \mathbf{u}_{DMD}^{M-1}\|_2$  is minimized.*

*Proof.* See [22, 69].  $\square$

### 2.1.2 Main Results

We rewrite the DMD prediction (1.24) as

$$\begin{aligned}
\mathbf{u}_{DMD}^{n+1} &= \Phi \Lambda^{n+1} \hat{\mathbf{u}}^0 \\
&= \Phi \Lambda \Phi^{-1} \Phi \Lambda^n \hat{\mathbf{u}}^0 \\
&= \Phi \Lambda \Phi^{-1} \mathbf{u}_{DMD}^n \\
&= \mathbf{u}_{DMD}^n + (\Phi \Lambda \Phi^{-1} - \mathbf{I}_{N_x \times N_x}) \mathbf{u}_{DMD}^n \\
&= \mathbf{u}_{DMD}^n + \mathbf{B} \mathbf{u}_{DMD}^n.
\end{aligned} \tag{2.8}$$

Here  $\mathbf{B} = \Phi \Lambda \Phi^{-1} - \mathbf{I}_{N_x \times N_x}$ , where  $\mathbf{I}$  represents identity matrix,  $\Phi$  is an  $N_x \times r$  matrix and  $\Phi^{-1}$  is an  $r \times N_x$  matrix defined as  $\Phi^{-1} \Phi = \mathbf{I}_{r \times r}$ .

**Theorem 2.1.** *Define the local truncation error*

$$\boldsymbol{\tau}^n = \mathbf{u}^n - \mathbf{u}_{DMD}^n(\mathbf{u}^{n-1}). \quad (2.9)$$

Then, for any  $n \geq M - 1$ ,

$$\|\boldsymbol{\tau}^n\|_2 \leq \varepsilon_M, \quad (2.10)$$

where the constant  $\varepsilon_M$  depends only on the number of snapshots  $M$ .

*Proof.*

$$\begin{aligned} \|\boldsymbol{\tau}^n\|_2^2 &= \|\mathbf{u}^n - \mathbf{u}_{DMD}^n(\mathbf{u}^{n-1})\|_2^2 \\ &= \|\mathbf{A} \mathbf{u}^{n-1} - \Phi \Lambda \Phi^{-1} \mathbf{u}^{n-1}\|_2^2 \\ &= \|(\mathbf{A} - \Phi \Lambda \Phi^{-1}) \mathbf{u}^{n-1}\|_2^2 \\ &\leq \|\mathbf{A} - \Phi \Lambda \Phi^{-1}\|_2^2 \|\mathbf{u}^{n-1}\|_2^2. \end{aligned} \quad (2.11)$$

Note that

$$\begin{aligned} \mathbf{A} - \Phi \Lambda \Phi^{-1} &= \mathbf{A} - (\mathbf{U} \mathbf{W}) \Lambda (\mathbf{U} \mathbf{W})^{-1} \\ &= \mathbf{A} - \mathbf{U} (\mathbf{W} \Lambda \mathbf{W}^{-1}) \mathbf{U}^* \\ &= \mathbf{A} - \mathbf{U} \mathbf{K}_r \mathbf{U}^* \\ &= (\mathbf{A} - \mathbf{X}_2 \mathbf{X}_1^\dagger) + (\mathbf{X}_2 \mathbf{X}_1^\dagger - \mathbf{U} \mathbf{K}_r \mathbf{U}^*). \end{aligned} \quad (2.12)$$

The term  $\mathbf{X}_2 \mathbf{X}_1^\dagger - \mathbf{U} \mathbf{K}_r \mathbf{U}^*$  introduces an error depending only on the rank truncation in the SVD step of the DMD algorithm. The error is assumed to be subordinate, i.e.,

$$\|\mathbf{X}_2 \mathbf{X}_1^\dagger - \mathbf{U} \mathbf{K}_r \mathbf{U}^*\|_2^2 \leq \delta \|\mathbf{A} - \mathbf{X}_2 \mathbf{X}_1^\dagger\|_2^2 \quad (2.13)$$

for constant  $\delta$  s.t.  $0 < \delta \ll 1$ . Subsequently,

$$\|\mathbf{A} - \Phi \Lambda \Phi^{-1}\|_2^2 \leq (1 + \delta) \|\mathbf{A} - \mathbf{X}_2 \mathbf{X}_1^\dagger\|_2^2. \quad (2.14)$$

Since  $\mathbf{X}_2\mathbf{X}_1^\dagger$  is the best-fit linear operator to approximate  $\mathbf{A}$  obtained from available  $M$  snapshots,

$$\mathbf{X}_2\mathbf{X}_1^\dagger = \operatorname{argmin}_{\hat{\mathbf{K}} \in \mathbb{R}^{N_x \times N_x}} \sum_{k=0}^{M-2} \|\mathbf{u}^{k+1} - \hat{\mathbf{K}}\mathbf{u}^k\|_2^2. \quad (2.15)$$

On the other hand,

$$\mathbf{A} = \operatorname{argmin}_{\hat{\mathbf{K}} \in \mathbb{R}^{N_x \times N_x}} \sum_{k=0}^{\infty} \|\mathbf{u}^{k+1} - \hat{\mathbf{K}}\mathbf{u}^k\|_2^2 \quad (2.16)$$

is the linear operator that fits all data. Thus,

$$\begin{aligned} & \|\mathbf{A} - \Phi\Lambda\Phi^{-1}\|_2^2 \\ & \leq (1 + \delta) \|\mathbf{A} - \mathbf{X}_2\mathbf{X}_1^\dagger\|_2^2 \\ & = (1 + \delta) \left\| \operatorname{argmin}_{\hat{\mathbf{K}} \in \mathbb{R}^{N_x \times N_x}} \sum_{k=0}^{\infty} \|\mathbf{u}^{k+1} - \hat{\mathbf{K}}\mathbf{u}^k\|_2^2 - \operatorname{argmin}_{\hat{\mathbf{K}} \in \mathbb{R}^{N_x \times N_x}} \sum_{k=0}^{M-2} \|\mathbf{u}^{k+1} - \hat{\mathbf{K}}\mathbf{u}^k\|_2^2 \right\|_2^2 \\ & \leq \|c_M\|_2^2 \end{aligned} \quad (2.17)$$

where  $c_M$  is a term depending on the number of snapshots  $M$ , Theorem 2.1 holds with

$$\varepsilon_M = \left( \|c_M\|_2^2 (\|\mathbf{u}^0\|_2^2 + \Delta t \sum_{k=0}^{n-1} \max\{\|\mathbf{f}_s^k\|_2^2, \|\mathbf{f}_s^{k+1}\|_2^2\}) \right)^{1/2}. \quad (2.18)$$

□

**Remark 2.1.** *The value of  $c_M$  decreases to 0 as  $M$  increases and so does  $\varepsilon_M$ . In the limit of large number of snapshots,  $\Lambda$  and  $\Phi$  become the exact eigenvalues and eigenvectors of  $\mathbf{A}$ . Then*

$$\begin{aligned} \|\mathbf{A} - \Phi\Lambda\Phi^{-1}\|_2 &= \sup_{z \in \mathbb{R}^{N_x} \setminus \{0\}} \frac{\|\mathbf{A}z - \Phi\Lambda\Phi^{-1}z\|_2}{\|z\|_2} \\ &= \sup_{w \in \mathbb{R}^{N_x} \setminus \{0\}} \frac{\|\mathbf{A}\Phi w - \Phi\Lambda w\|_2}{\|\Phi w\|_2} \\ &= \sup_{w \in \mathbb{R}^{N_x} \setminus \{0\}} \frac{\|\Lambda\Phi w - \Phi\Lambda w\|_2}{\|\Phi w\|_2} \\ &= 0. \end{aligned} \quad (2.19)$$

In other words, the more snapshots are obtained, the more accurate the approximation of  $\mathbf{A}$  becomes. Thus, the local truncation error caused by replacing  $\mathbf{A}$  with  $\mathbf{B}$  can be minimized. A convergence proof of the eigenvalue and eigenfunction approximation of  $\mathbf{A}$  by DMD and convergence from  $\mathbf{A} \rightarrow \mathcal{K}_{\Delta t}$  can be found in [115].

**Remark 2.2.** For fixed  $M$ , the local truncation error can be improved by refining the Ritz pairs in the DMD algorithm. See [69].

**Theorem 2.2.** Define the global truncation error

$$\mathbf{e}^n = \mathbf{u}^n - \mathbf{u}_{\text{DMD}}^n. \quad (2.20)$$

Then, for  $n \geq M - 1$ ,

$$\|\mathbf{e}^n\|_2 < \|\Phi\|_2 \|\Phi^{-1}\|_2 [\|\mathbf{e}^{M-1}\|_2 + (n - M + 1)\varepsilon_M]. \quad (2.21)$$

*Proof.* Subtracting

$$\mathbf{u}_{\text{DMD}}^n = \mathbf{u}_{\text{DMD}}^{n-1} + \mathbf{B}\mathbf{u}_{\text{DMD}}^{n-1} \quad (2.22)$$

from

$$\mathbf{u}^n = \mathbf{u}^{n-1} + \tau^n + \mathbf{B}\mathbf{u}^{n-1}, \quad (2.23)$$

one gets

$$\begin{aligned} \mathbf{e}^n &= \mathbf{u}^n - \mathbf{u}_{\text{DMD}}^n \\ &= \mathbf{e}^{n-1} + \tau^n + \mathbf{B}\mathbf{e}^{n-1} \\ &= \tau^n + \Phi\Lambda\Phi^{-1}\mathbf{e}^{n-1} \\ &= \tau^n + \Phi\Lambda\Phi^{-1}(\tau^{n-1} + \Phi\Lambda\Phi^{-1}\mathbf{e}^{n-2}) \\ &= \tau^n + \Phi\Lambda\Phi^{-1}\tau^{n-1} + \Phi\Lambda^2\Phi^{-1}\mathbf{e}^{n-2} \\ &= \dots \\ &= \Phi\Lambda^{n-M+1}\Phi^{-1}\mathbf{e}^{M-1} + \sum_{k=0}^{n-M} \Phi\Lambda^k\Phi^{-1}\tau^{n-k}. \end{aligned} \quad (2.24)$$

Then

$$\begin{aligned}
& \left\| \Phi \Lambda^{n-M+1} \Phi^{-1} \mathbf{e}^{M-1} + \sum_{k=0}^{n-M} \Phi \Lambda^k \Phi^{-1} \tau^{n-k} \right\|_2 \\
& \leq \left\| \Phi \Lambda^{n-M+1} \Phi^{-1} \right\|_2 \left\| \mathbf{e}^{M-1} \right\|_2 + (n-M+1) \varepsilon_M \max_{0 \leq k \leq n-M} \left\| \Phi \Lambda^k \Phi^{-1} \right\|_2 \\
& \leq \left\| \Phi \Lambda^{n-M+1} \right\|_2 \left\| \Phi^{-1} \right\|_2 \left\| \mathbf{e}^{M-1} \right\|_2 + (n-M+1) \varepsilon_M \max_{0 \leq k \leq n-M} \left\| \Phi \Lambda^k \right\|_2 \left\| \Phi^{-1} \right\|_2 \\
& \leq \left\| \Phi^{-1} \right\|_2 \left( \left\| \Lambda^{n-M+1} \right\|_2 \left\| \Phi \right\|_2 \left\| \mathbf{e}^{M-1} \right\|_2 + (n-M+1) \varepsilon_M \max_{0 \leq k \leq n-M} \left\| \Lambda^k \right\|_2 \left\| \Phi \right\|_2 \right) \\
& = \left\| \Phi \right\|_2 \left\| \Phi^{-1} \right\|_2 \left( \rho(\Lambda^{n-M+1}) \left\| \mathbf{e}^{M-1} \right\|_2 + (n-M+1) \varepsilon_M \max_{0 \leq k \leq n-M} \rho(\Lambda^k) \right) \\
& \leq \left\| \Phi \right\|_2 \left\| \Phi^{-1} \right\|_2 \left[ \rho(\Lambda) \left\| \mathbf{e}^{M-1} \right\|_2 + (n-M+1) \varepsilon_M \rho(\Lambda) \right] \\
& < \left\| \Phi \right\|_2 \left\| \Phi^{-1} \right\|_2 \left[ \left\| \mathbf{e}^{M-1} \right\|_2 + (n-M+1) \varepsilon_M \right].
\end{aligned} \tag{2.25}$$

According to Lemma 2.2,  $\|\mathbf{e}^{M-1}\|$  is fixed and minimal. Hence, if accuracy of the local truncation error is of  $\mathcal{O}((\Delta t)^q)$ , then the global truncation error is of  $\mathcal{O}((\Delta t)^{q-1})$ .  $\square$

Theorem 2.2 provides quantitative error bounds of the DMD method with explicit error dependence. In practice, one can determine  $c_M$  by  $\|\mathbf{A} - \Phi \Lambda \Phi^{-1}\|_2$  and determine  $\varepsilon_M$  from (2.18) subsequently. In complex simulations, one would not expect the DMD prediction from a local data set to capture the global dynamics accurately. Instead, one can use the error bounds to set up a threshold for DMD prediction limits and combine a resolved algorithm with fast DMD prediction. This would considerably speed up the simulations.

### 2.1.3 Application to Nonlinear Parabolic Problems

Consider a general nonlinear reaction-diffusion equation in  $d$  spatial dimensions

$$\begin{cases} \partial_t u = \nabla \cdot [k\psi(u)\nabla u] + f_s(u), & \mathbf{x} \in \mathcal{D} \subset \mathbb{R}^d, \quad t > 0 \\ u(\mathbf{x}, 0) = u_0(\mathbf{x}), & \mathbf{x} \in \mathcal{D}, \end{cases} \tag{2.26}$$

with non-negative functions  $k = k(\mathbf{x})$  and  $\psi = \psi(u)$  whose product is diffusion coefficient  $D(\mathbf{x}, u) = k(\mathbf{x})\psi(u)$ . Spatial discretization of (2.26) leads to the corresponding high-dimensional nonlinear ODE (2.3). Its DMD treatment relies on the one's ability to identify informative observables and requires the prior knowledge of the structure of governing equations such as (2.26). Examples in section 2.2.2 to section 2.2.4 illustrate the critical role of observable selection in the DMD method.

For (2.26), expert knowledge suggests the existence of a function  $\eta(u)$  such that  $\eta'(u) = \psi(u)$ , which can be constructed via the Kirchhoff transform (e.g., [117, 118]). Then, by chain rule, (2.26) is rewritten as

$$\begin{cases} \partial_t u - \nabla \cdot [k \nabla \eta(u)] = f_s(u) \\ u(x, 0) = u_0(x), \end{cases} \quad (2.27)$$

so that the nonlinear diffusion in  $u$  becomes linear in  $\eta$ . Spatial discretization of (2.27) leads to

$$\frac{d\mathbf{u}}{dt} = \mathcal{A}\boldsymbol{\eta}(\mathbf{u}) + \mathbf{f}_s(\mathbf{u}) \quad (2.28)$$

where  $\mathcal{A}$  is the same linear operator in (2.2). Motivated by the nonlinear observable choice for the nonlinear Schrödinger equation in [58], and by the accurate and robust performance of DMD on linear diffusion reported below, we choose the observable

$$\mathbf{g} = [g_1(\mathbf{u}), \dots, g_p(\mathbf{u})], \quad \text{s.t. } \mathbf{u}, \boldsymbol{\eta}(\mathbf{u}), \mathbf{f}_s(\mathbf{u}) \in \text{span}\{g_1(\mathbf{u}), \dots, g_p(\mathbf{u})\}. \quad (2.29)$$

The reference solution of (2.26) is obtained by discretizing (2.28) in time,

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t \boldsymbol{\eta}^* + \Delta t \mathbf{f}_s^*, \quad (2.30)$$

where the superscript  $*$  denotes linear interpolation between time  $t^{n+1}$  and  $t^n$ . For the observables in (2.29), we have

$$\begin{aligned} \mathbf{u}^{n+1}, \boldsymbol{\eta}^{n+1}, \mathbf{f}_s^{n+1} &\in \text{span}\{g_1(\mathbf{u}^{n+1}), \dots, g_p(\mathbf{u}^{n+1})\}, \\ \mathbf{u}^n, \boldsymbol{\eta}^n, \mathbf{f}_s^n &\in \text{span}\{g_1(\mathbf{u}^n), \dots, g_p(\mathbf{u}^n)\}. \end{aligned} \quad (2.31)$$

Thus, Algorithm 2 induces a linear operator denoted by  $\mathbf{K}$  such that

$$\mathbf{y}^{n+1} = \mathbf{K}\mathbf{y}^n, \quad (2.32)$$

where  $\mathbf{y}^n = \mathbf{g}(\mathbf{u}^n)$  defined in (1.27). Treating (2.32) as reference solution, against which we compare the DMD prediction (1.30), one gets exactly the same formulae as (2.4) and (1.24) but in observable space:

$$\begin{aligned} \mathbf{y}^{n+1} &= \mathbf{K}\mathbf{y}^n, \\ \mathbf{y}_{\text{DMD}}^{n+1} &= \mathbf{\Phi}\mathbf{\Lambda}^{n+1}\hat{\mathbf{y}}^0. \end{aligned} \quad (2.33)$$

So the error analysis in section 2.1 carries on in terms of  $\mathbf{y}$ .

## 2.2 Numerical Tests of Predictive Accuracy

We test the robustness of our error estimates and the DMD performance in the extrapolation regime on several test problems arranged in order of difficulty.

In our resolved simulations, we use finite difference in space and forward Euler in time with CFL condition  $\Delta t \sim \mathcal{O}((\Delta x)^2)$ . Although there are many relatively efficient implicit/semi-implicit solvers, the computational difficulty of solving high-dimensional systems iteratively remains essentially the same. We would regard them the same order of computational time and simply take the fully explicit discretization as the resolved solutions. In the following tests,  $N_x = 500$  spatial mesh is created in  $x$  and  $N_t = 500$  solutions are uniformly selected from a specified time interval. Thus, the reference solution is built on this  $500 \times 500$  mesh. We also compare the relative performance of DMD and POD(-DEIM) in terms of both their computational time and error with respect to the reference solution.

### 2.2.1 Linear Diffusion (Test 1)

We start with a linear diffusion equation,

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, \quad x \in [0, 1], \quad t \in [0, T] \quad (2.34a)$$

subject to several sets of initial and boundary conditions

$$u(x, 0) = u_0, \quad u(0, t) = u_L, \quad u(1, t) = 1. \quad (2.34b)$$

Discretization of the spatial domain  $[0, 1]$  with a fine mesh of size  $\Delta x \ll 1$  gives rise to the equivalent high-dimensional ODE (2.2), where  $\mathbf{u} = [u(x_1, t), \dots, u(x_{N_x}, t)]^\top$  is the spatial discretization of  $u(x, t)$  with  $N_x \gg 1$  and  $\mathcal{A}$  is a linear operator representing the diffusion.

#### 2.2.1.1 Relaxation to Equilibrium (Test 1a)

Consider (2.34) with  $T = 0.2$ ,  $u_0 = 0$ , and  $u_L = 0$ . Figure 2.1 demonstrates visual agreement between the true solution  $u(x, t)$  and its counterpart predicted by DMD with  $M = 200$  temporal snapshots, the two solutions converge to the same stationary state.

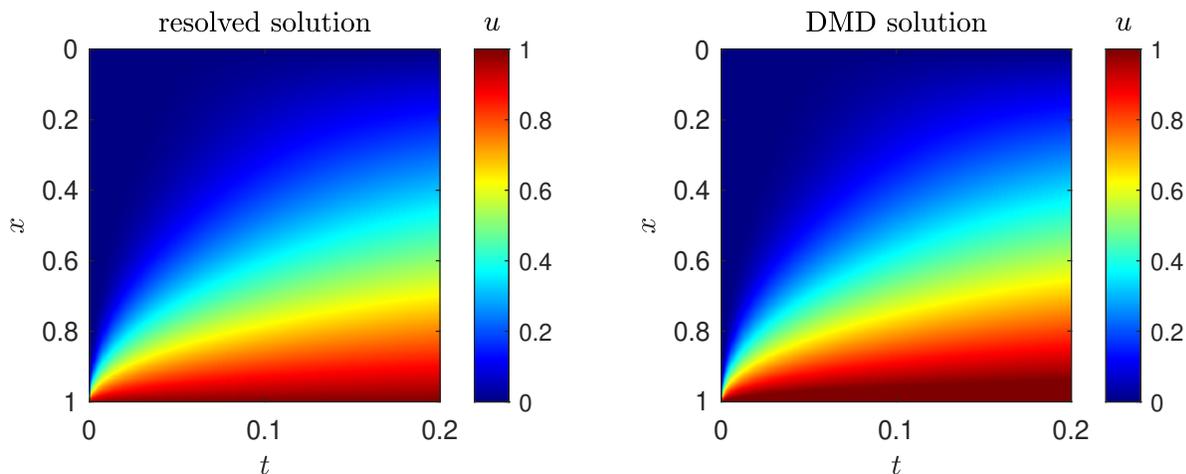


Figure 2.1: Test 1a. Reference solution (left) and its DMD approximation with  $M = 200$  snapshots (right).

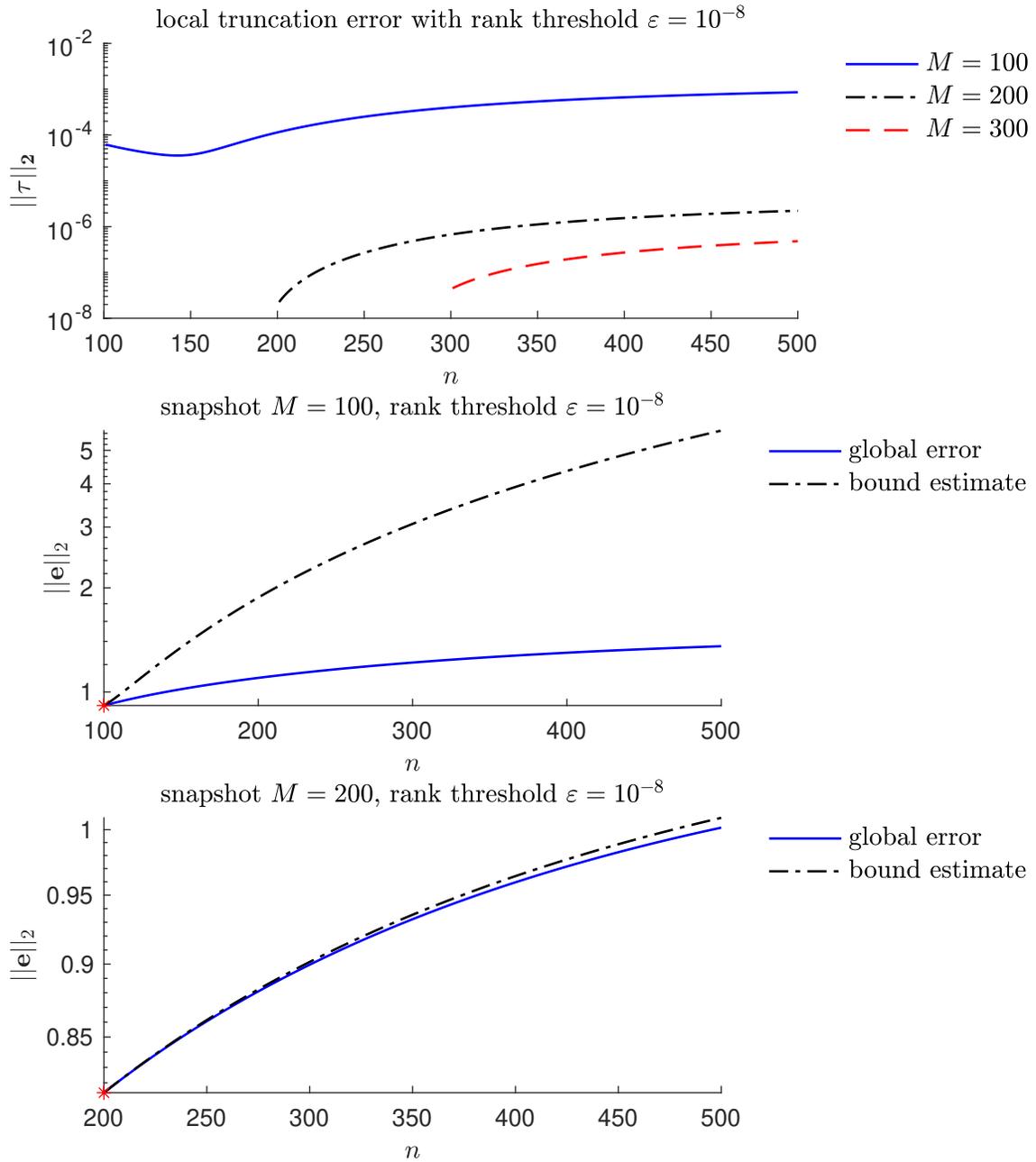


Figure 2.2: Test 1a. Top: Local truncation error for DMD with  $M = 100$ , 200 and 300 snapshots; Middle: global error for DMD and our error bound estimator with  $M = 100$ ; Bottom: global error for DMD and our error bound estimator with  $M = 200$  snapshots. The global error is negligible for  $M = 300$  (not shown). The rank threshold is set to  $\varepsilon = 10^{-8}$ .

Figure 2.2 exhibits the local truncation error  $\tau$  (2.9) and the global truncation error  $\mathbf{e}$  (2.20) of the DMD with  $M = 100, 200$  and  $300$  snapshots of the reference solution. The rank in step 1 of Algorithm 1 is truncated by the criteria of

$$r = \max\{i : \sigma_i > \epsilon\sigma_1\}, \quad (2.35)$$

where  $\sigma_i$  are the diagonal elements of  $\Sigma$  in SVD. The figure shows that the local truncation error decreases with the number of snapshots, resulting in a more accurate prediction. This is consistent with the intuition that DMD can better capture the dynamics by learning from richer/larger data sets.

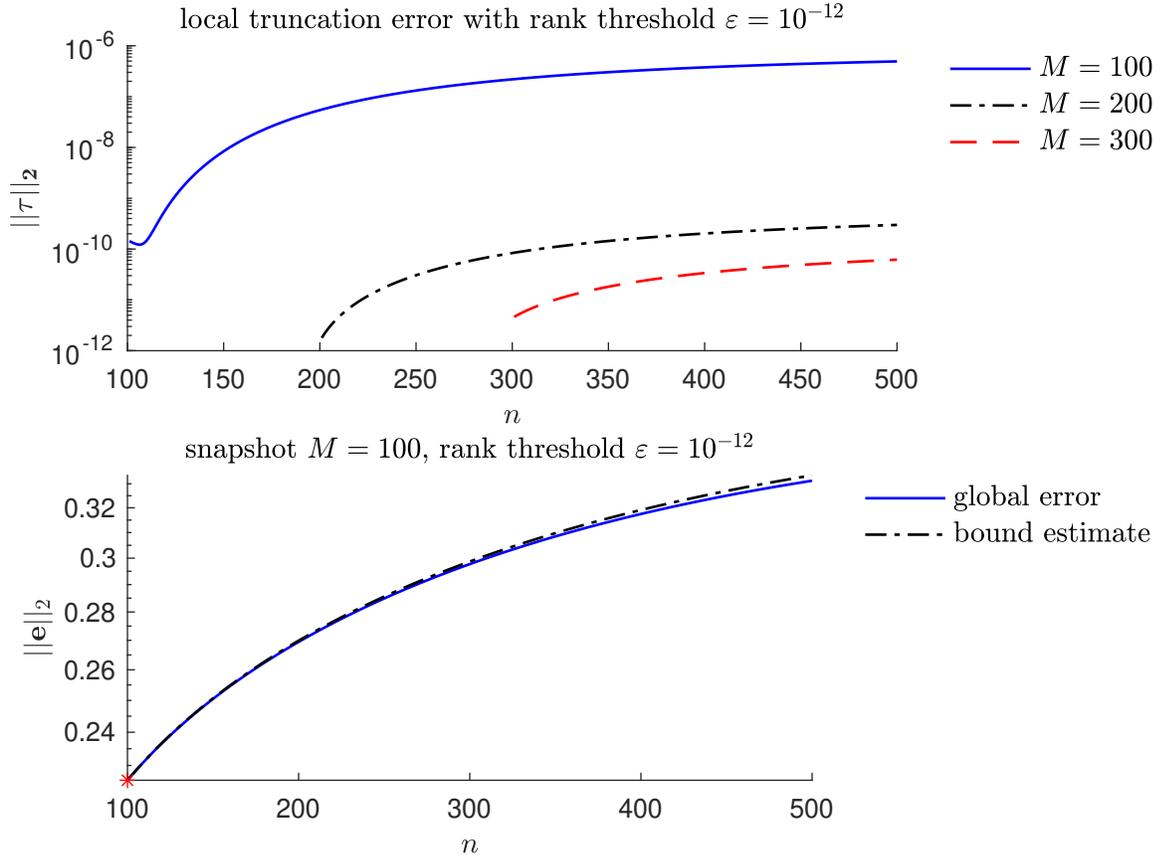


Figure 2.3: Test 1a. Top: Local truncation error for DMD with  $M = 100, 200$  and  $300$  snapshots; Bottom: global error for DMD and our error bound estimator with  $M = 100$  snapshots. The global error is negligible for  $M = 200$  and  $300$  (not shown). The rank threshold is set to  $\epsilon = 10^{-12}$ .

If a more stringent condition on the rank truncation is imposed, i.e., a relatively higher-order surrogate model is established, further reduction in both local and global errors is observed (Figure 2.3). The good performance of DMD in Test 1a is not surprising: the monotonic (exponential) decay of the solution to the linear diffusion equation is captured by a relatively few temporal snapshots. The next example provides a more challenging test by introducing temporal fluctuations at the boundary  $x = 0$ .

### 2.2.1.2 Periodic Boundary Fluctuations (Test 1b)

Consider (2.34) with  $T = \pi/2$ ,  $u_0 = 1$ , and  $u_L = 1.01 + 0.01 \sin(-\pi/2 + 10t)$ . Figure 2.4 demonstrates that  $M = 200$  snapshots is sufficient for DMD to match the reference solution. The corresponding local and global truncation errors are plotted in Figure 2.5. Since the solution  $u(x, t)$  to (2.34) with the parameter values used in Test 1b has a period of  $\pi/5$ ,  $M = 100$  snapshots are not enough to cover the whole period. Consequently, DMD fails to capture the system dynamic and to predict the future states accurately. However, once the full period of the solution is covered by snapshots data, i.e., when  $M = 200$  or 300 snapshots are used, DMD is accurate even for long-time prediction. The error bound in Theorem 2.2 does a good job bounding the computed error.

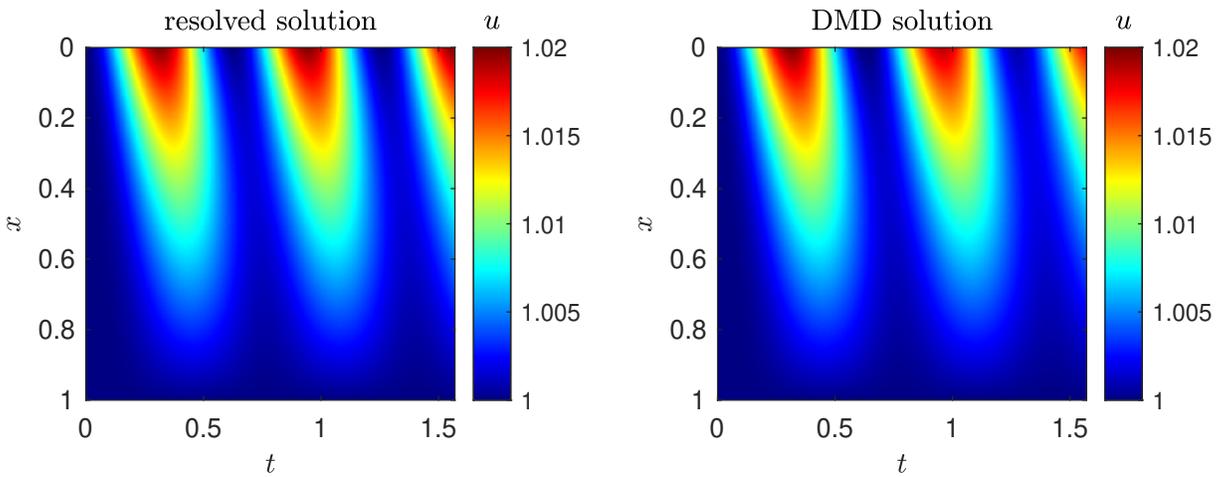


Figure 2.4: Test 1b. Reference solution (left) and its DMD approximation with  $M = 200$  snapshots (right).

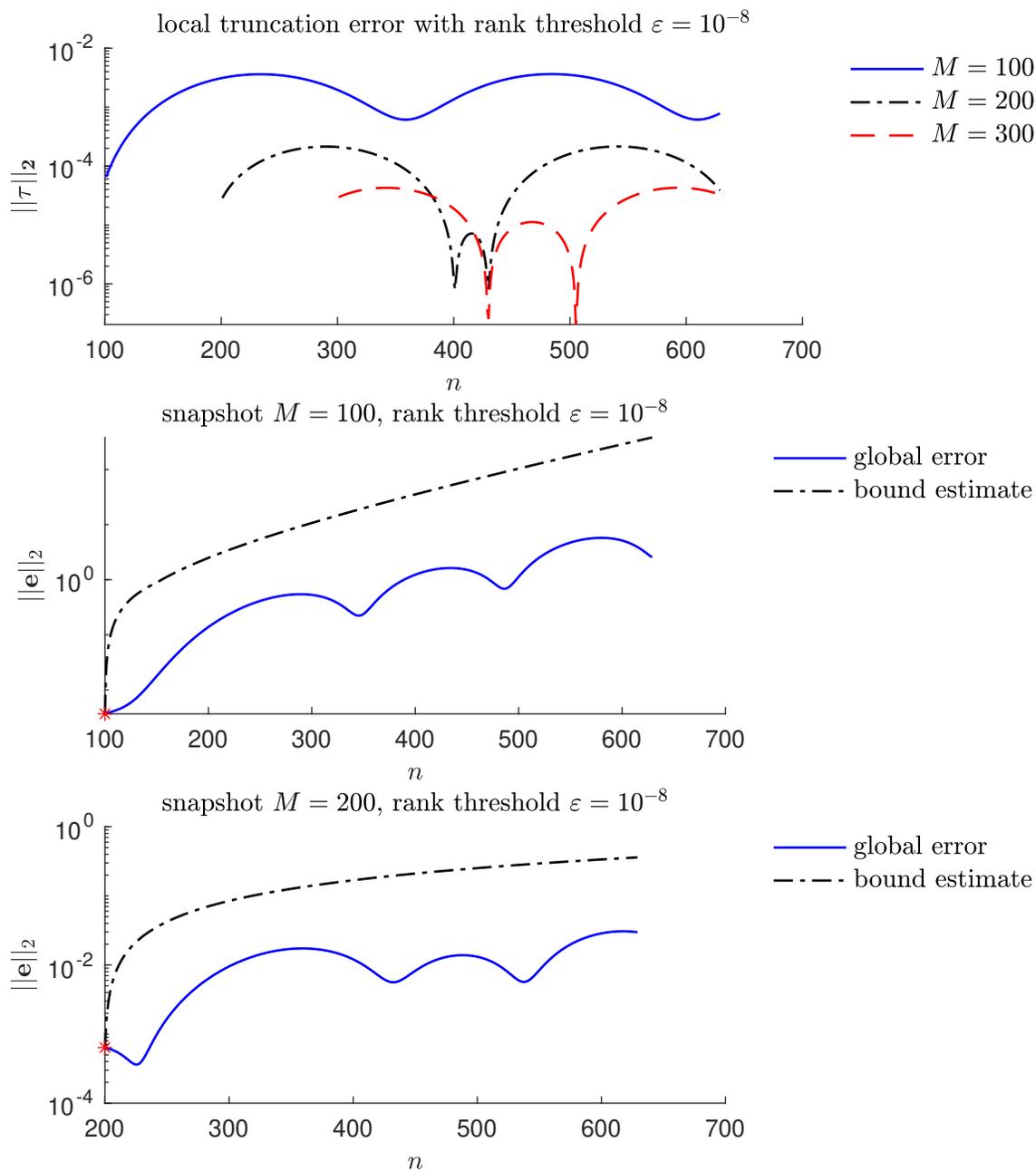


Figure 2.5: Test 1b. Top: Local truncation error for DMD with  $M = 100$ , 200 and 300 snapshots; Middle: global error for DMD and our error bound estimator with  $M = 100$ ; Bottom: global error for DMD and our error bound estimator with  $M = 200$  snapshots. The global error is negligible for  $M = 300$  (not shown). The rank threshold is set to  $\varepsilon = 10^{-8}$ .

Although not shown here, the reliance on a more restricted rank truncation, i.e., setting the rank threshold to  $\varepsilon = 10^{-12}$ , improves DMD’s accuracy by at least an order of magnitude for the parameter values considered.

## 2.2.2 Reaction-Diffusion Equation (Test 2)

Consider a reaction-diffusion equation

$$\frac{\partial u}{\partial t} = \theta \frac{\partial^2 u}{\partial x^2} - \mu(u - u^3), \quad x \in [0, 1], \quad t \in [0, 2] \quad (2.36a)$$

with constant coefficients  $\theta, \mu \in \mathbb{R}^+$ . It is subject to initial and boundary conditions

$$u(x, 0) = 0.5 + 0.5 \sin(\pi x), \quad u(0, t) = 0, \quad u(1, t) = 0. \quad (2.36b)$$

### 2.2.2.1 Diffusion-Dominated Regime (Test 2a)

To achieve this regime ( $\theta \gg \mu$ ), we set  $\theta = 0.1$  and  $\mu = 0.01$ . Figure 2.6 exhibits the fully resolved solution with its approximations provided by DMD with different observables,  $\mathbf{g}_1(\mathbf{u}) = \mathbf{u}$  and  $\mathbf{g}_2(\mathbf{u}) = [\mathbf{u}; \mathbf{u}^3]$ , and by POD-DEIM. In Figure 2.6, the choice of observables does not appreciably affect DMD’s performance due to the dominating linear diffusion, though one can still observe higher order accuracy of  $\mathbf{g}_2$  than  $\mathbf{g}_1$  in the logarithm solution error plot Figure 2.7.

The corresponding prediction errors are also reported in Figure 2.7. With the same rank truncation criteria, POD is more accurate than DMD, especially in the absence of “right” observables. However, DMD is much faster than POD. We report the computational costs comparison in section 2.2.2.3.

### 2.2.2.2 Reaction-Dominated Regime (Test 2b)

To explore this regime ( $\mu \gg \theta$ ), we set  $\theta = 0.1$  and  $\mu = 1$ . Now the choice of observables has significant (visual) impact on the predictive accuracy (Figure 2.8). The Koopman operator theory helps explain this observation. Since the nonlinear source term dominates the dynamics, only the consistent observables can capture the

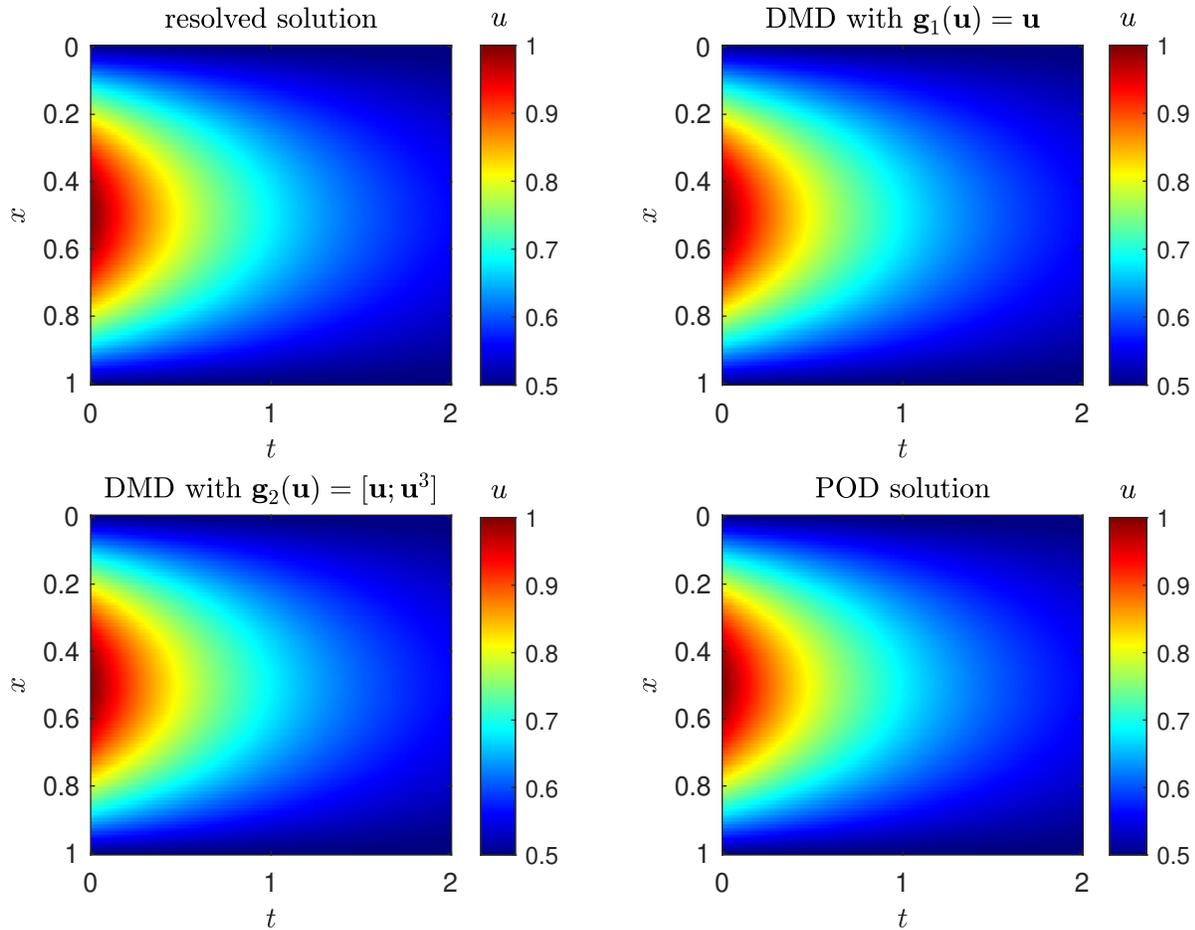


Figure 2.6: Test 2a. Fully resolved solution  $u(x, t)$  of the reaction-diffusion problem (2.36) in the diffusion-dominated regime, and its approximations obtained from  $M = 200$  snapshots with DMD (with two sets of observables  $\mathbf{g}$ ) and POD-DEIM.

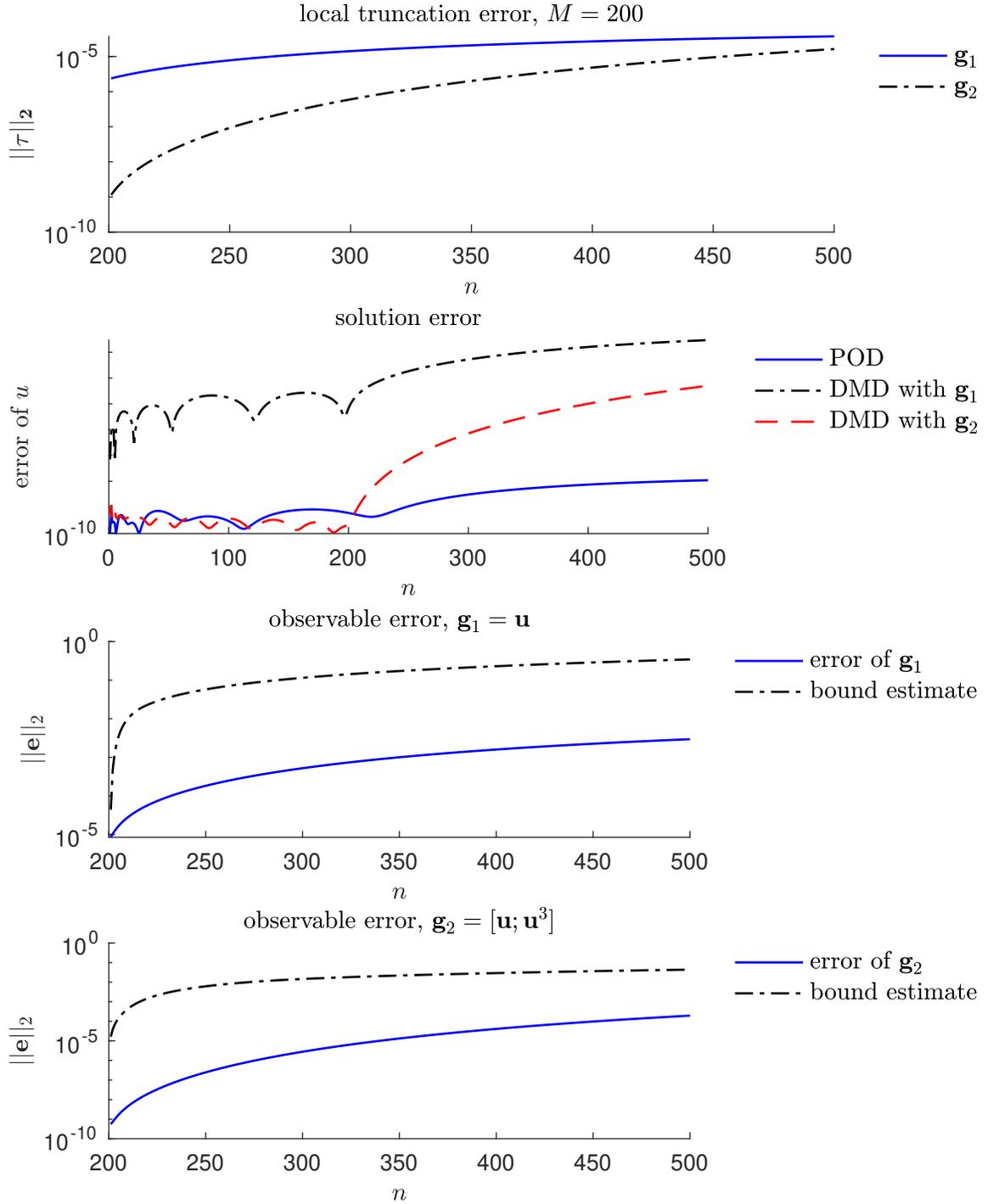


Figure 2.7: Test 2a. Local truncation error; Comparison of POD and DMD errors of the solution; Global error (errors of the observables) for DMD prediction with observables  $\mathbf{g}_1$  and  $\mathbf{g}_2$ . All ROMs are constructed using  $M = 200$  snapshots.

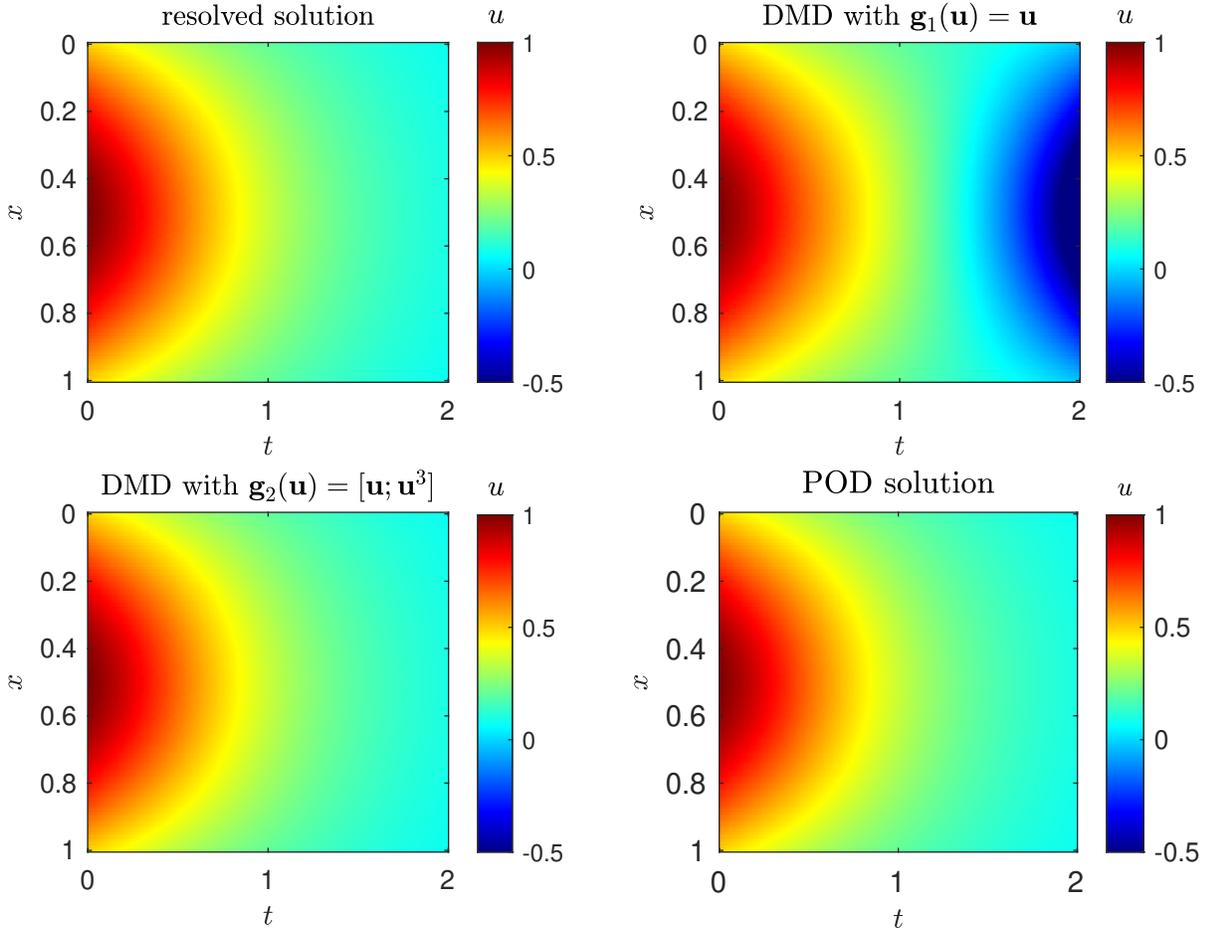


Figure 2.8: Test 2b. Fully resolved solution  $u(x, t)$  of the reaction-diffusion problem (2.36) in the reaction-dominated regime, and its approximations obtained from  $M = 200$  snapshots with DMD (with two sets of observables  $\mathbf{g}$ ) and POD-DEIM.

eigenvalues and eigenfunctions of the Koopman operator.

Errors of DMD prediction relying on the observables  $\mathbf{g}_1(\mathbf{u}) = \mathbf{u}$  and  $\mathbf{g}_2(\mathbf{u}) = [\mathbf{u}; \mathbf{u}^3]$  are shown in Figure 2.9. Our error estimation Theorem 2.2 indicates the failure of the DMD prediction based on the observable  $\mathbf{g}_1(\mathbf{u}) = \mathbf{u}$  and provides robust error bound for the DMD prediction based on the observable  $\mathbf{g}_2(\mathbf{u}) = [\mathbf{u}; \mathbf{u}^3]$ . For the same rank truncation criteria, the errors of POD and DMD with using  $\mathbf{g}_2(\mathbf{u})$  are comparable, while that of DMD with  $\mathbf{g}_1(\mathbf{u})$  is orders of magnitude higher.

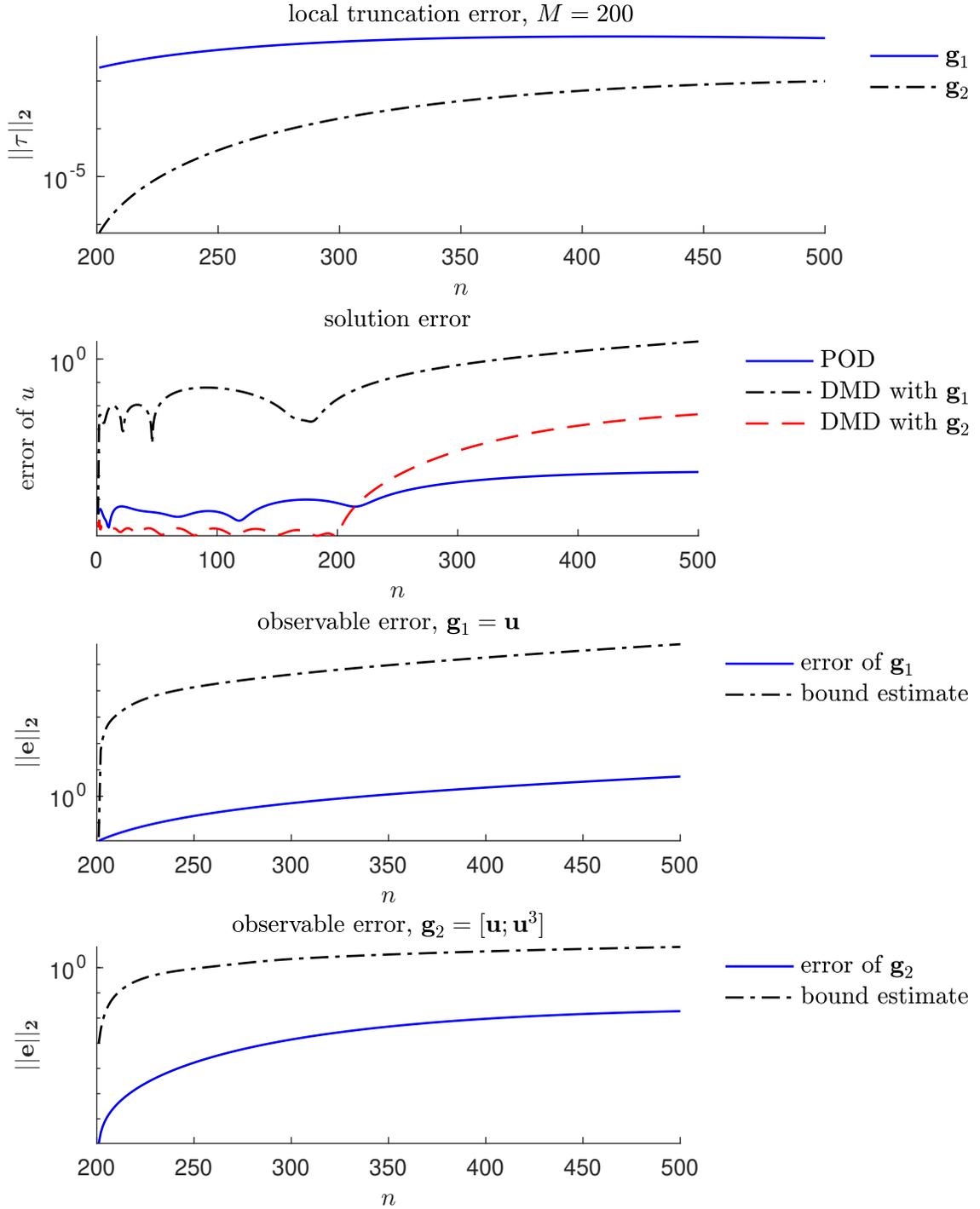


Figure 2.9: Test 2b. Local truncation error; Comparison of POD and DMD errors of the solution; Global error (errors of the observables) for DMD prediction with observables  $\mathbf{g}_1$  and  $\mathbf{g}_2$  using  $M = 200$  snapshots.

### 2.2.2.3 Comparison of POD and DMD

Comparison of the computation time and accuracy of DMD and POD-DEIM is presented in Figure 2.10 for Test 2b. The computational time comparison is made for the same rank truncation criteria. Note that the rank of the reduced-order model is different for DMD and POD because of the different dimension of the input data matrix. The ROM derived by DMD is in observable space and the ROM derived by POD is in state space.

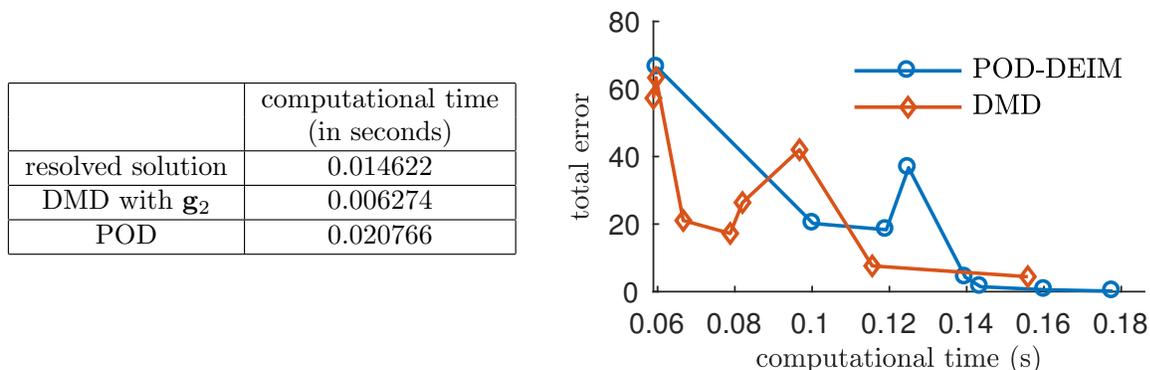


Figure 2.10: Test 2b. Computational times of the fully resolved solution, POD-DEIM, and DMD with the observables  $\mathbf{g}_2(\mathbf{u})$  (left table); Comparison of POD and DMD in terms of computational time and accuracy (right figure).

Figure 2.10 demonstrates that DMD prediction is computationally efficient due to its iteration-free feature. POD, on the other hand, is computationally more expensive than the fully resolved solver because the computational cost saved by ROM in the prediction process does not compensate for the cost of establishing the ROM by SVD and DEIM. This would not be the case for higher dimensional problems and longer prediction times. However, being non-iterative, DMD would outperform POD on such problems as well.

Both the accuracy and computational time depend on the rank of the ROM. The table in Figure 2.10 reveals that POD has advantage in accuracy and DMD has advantage in efficiency. Thus, if one wants a fast prediction with slightly lower accuracy, then DMD is a better choice and vice versa.

### 2.2.3 Nonlinear Reaction-Diffusion Equation (Test 3)

Consider a reaction-diffusion equation with the state-dependent diffusion coefficient,

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \left( u \frac{\partial u}{\partial x} \right) - (u - u^3), \quad 0 < x < 1, \quad t > 0. \quad (2.37a)$$

It is subject to the initial and boundary conditions

$$u(x, 0) = 0.5 + 0.5 \sin(\pi x), \quad u(0, t) = 0, \quad u(1, t) = 0. \quad (2.37b)$$

As discussed earlier, the Koopman operator theory suggests that only the physical-informed observables can capture the dynamical systems. To identify the relevant observables, we use the Kirchhoff transformation to recast (2.37a) as

$$\frac{\partial u}{\partial t} = \frac{\partial^2 \phi}{\partial x^2} - (u - u^3), \quad \phi = u^2/2. \quad (2.38)$$

This form suggests a set of observables  $\mathbf{g}_2 = [\mathbf{u}; \mathbf{u}^2; \mathbf{u}^3]$ .

Figure 2.11 and Figure 2.12 provide the visual and quantitative comparison between the fully resolved solution  $u(x, t)$  and its POD and DMD approximations. The performance of these approximators on this highly nonlinear problem is qualitatively similar to its weakly nonlinear counterpart analyzed in section 2.2.2. For the inadequate choice of observables,  $\mathbf{g}_1 = \mathbf{u}$ , our error bound diverges from the true error because of the fast decay of both the reference and wrong solutions. Nevertheless, the error bound still serves as a good discriminator between the accurate or inaccurate predictions. For the proper choice of observables,  $\mathbf{g}_2 = [\mathbf{u}; \mathbf{u}^2; \mathbf{u}^3]$ , our error bound remains accurate.

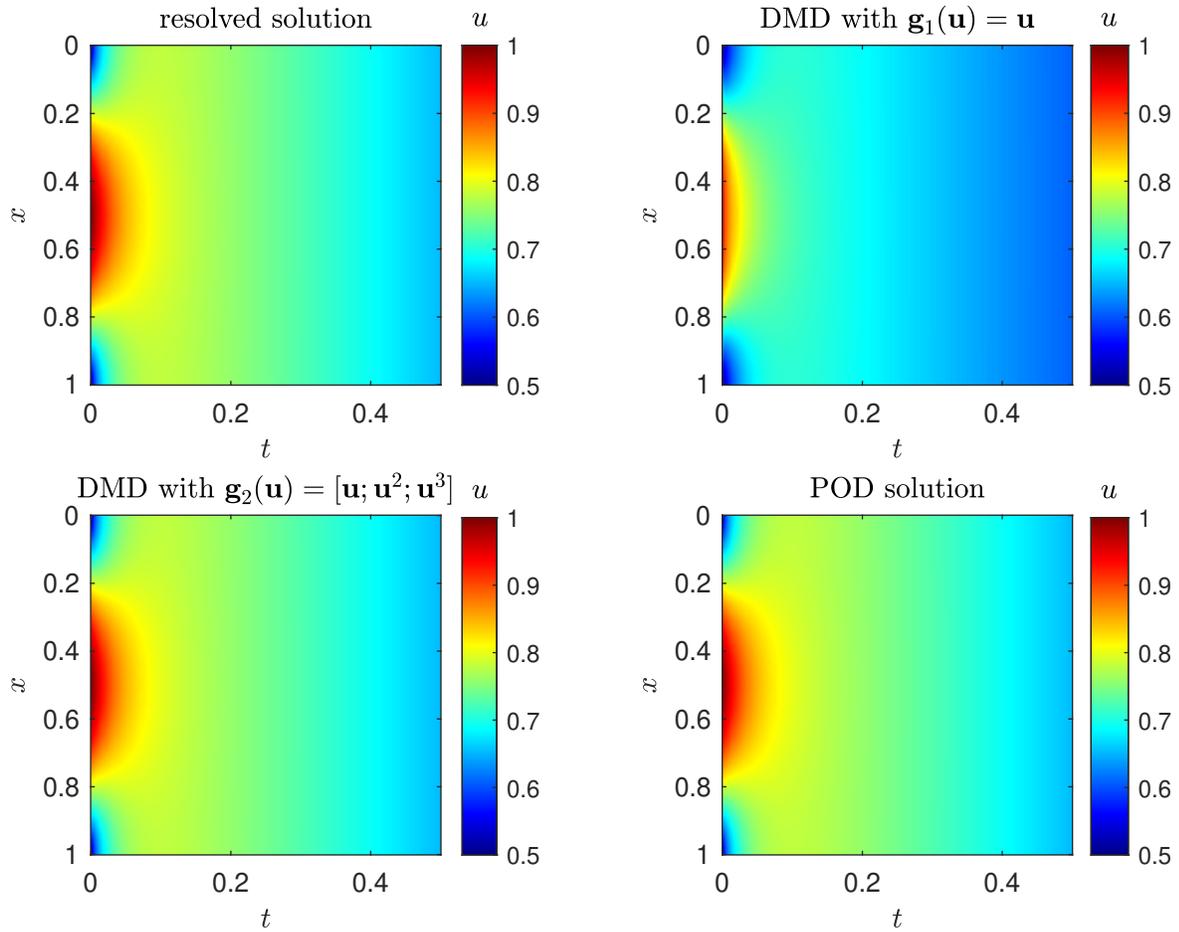


Figure 2.11: Test 3. Fully resolved solution  $u(x, t)$  of the nonlinear reaction-diffusion problem (2.37) and its approximations obtained from  $M = 200$  snapshots with DMD (with two sets of observables  $\mathbf{g}$ ) and POD-DEIM.

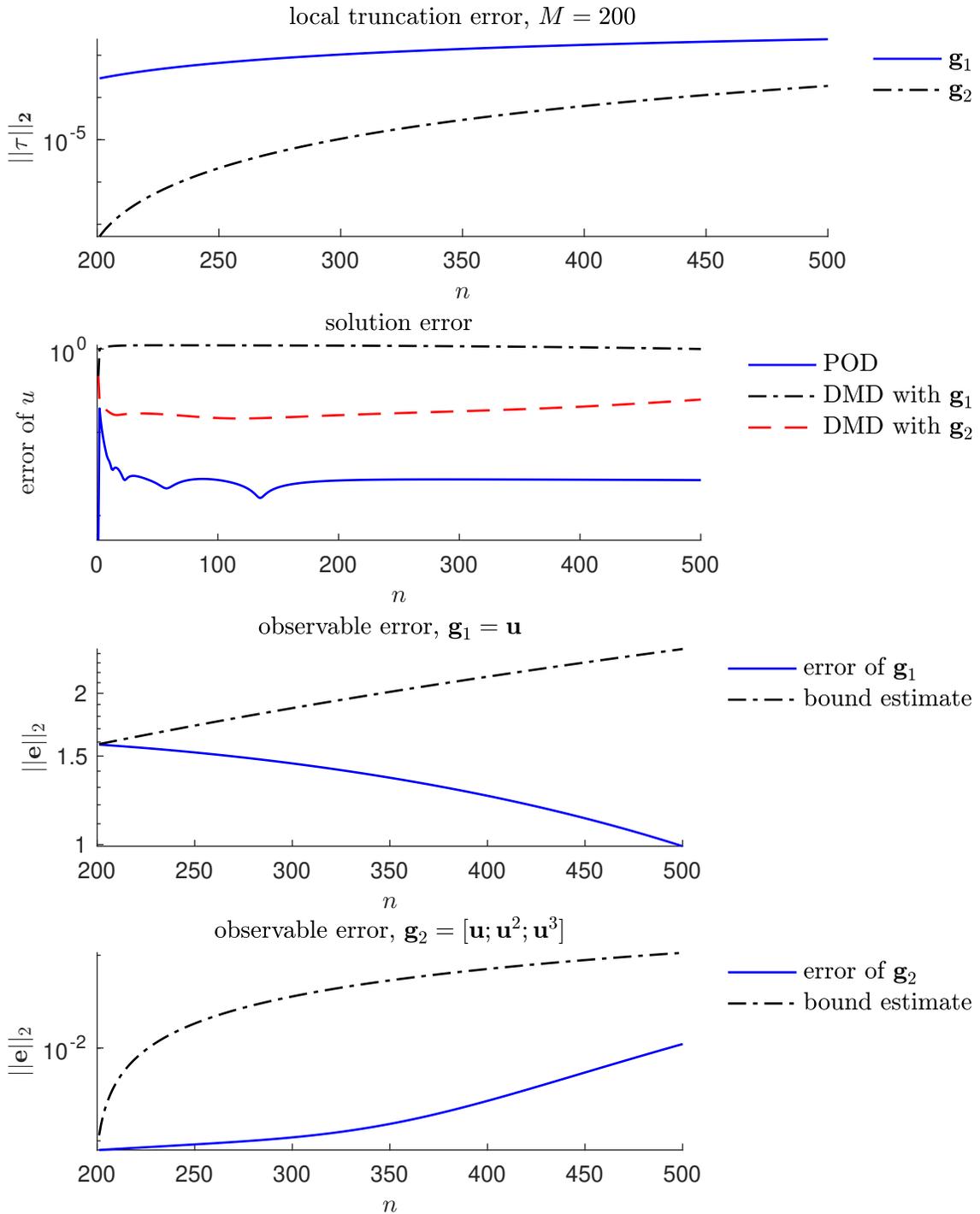


Figure 2.12: Test 3. Comparison of POD and DMD errors; Local truncation error and global error for DMD prediction with  $\mathbf{g}_1$  and  $\mathbf{g}_2$  using  $M = 200$  snapshots.

### 2.2.4 Nonlinear Schrödinger Equation (Test 4)

Finally, we consider the nonlinear Schrödinger equation,

$$\begin{cases} i\frac{\partial u}{\partial t} + \frac{1}{2}\frac{\partial^2 u}{\partial x^2} + |u|^2 u = 0, \\ u(x, 0) = 2\text{sech}(x). \end{cases} \quad (2.39)$$

It belongs to the the general class of nonlinear parabolic PDEs (2.1) and satisfies all of the assumptions underlying our error estimator. The reference solution is obtained by using Fast Fourier Transform in space and Runge-Kutta in time evolution.

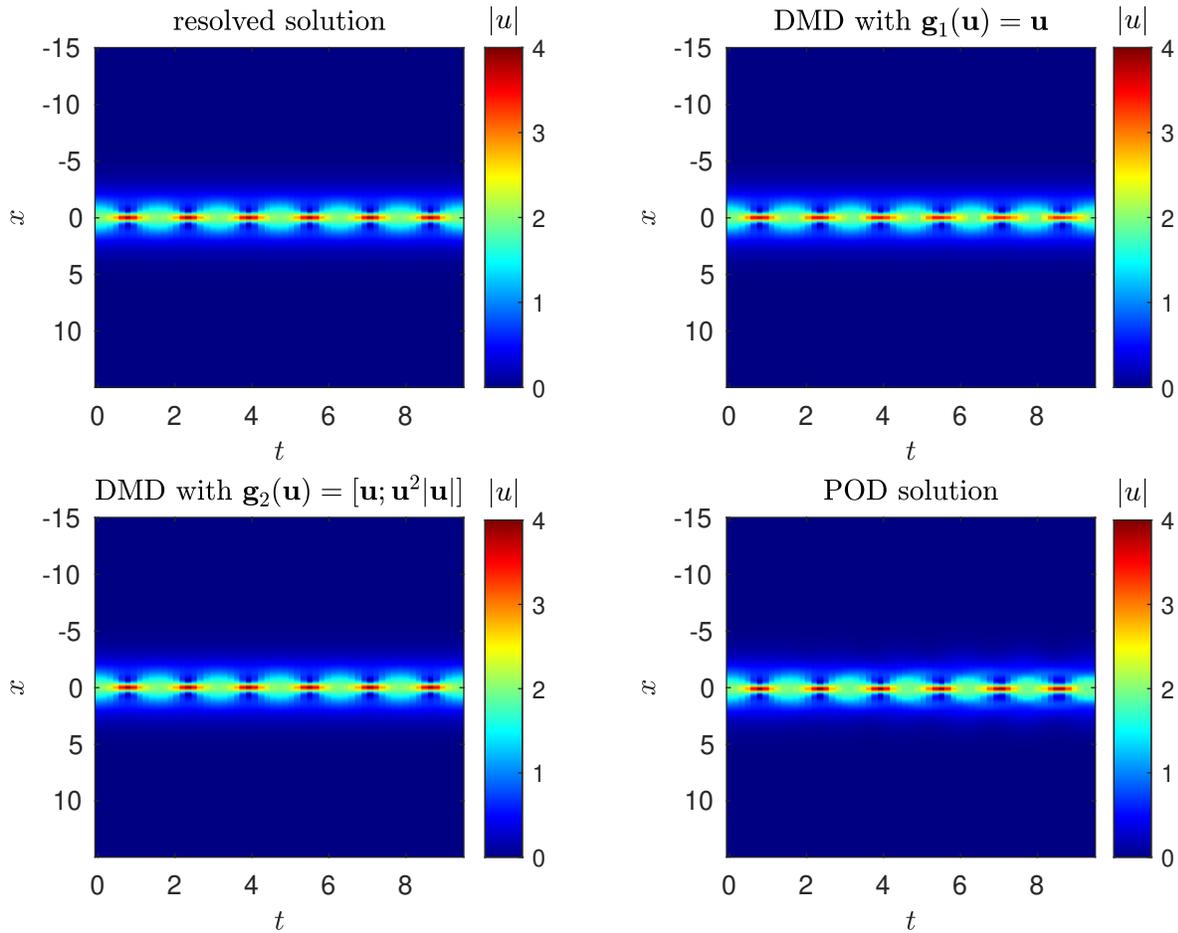


Figure 2.13: Test 4. Resolved solution, DMD solutions and POD solution using  $M = 20$  snapshots; Comparison of POD and DMD errors.

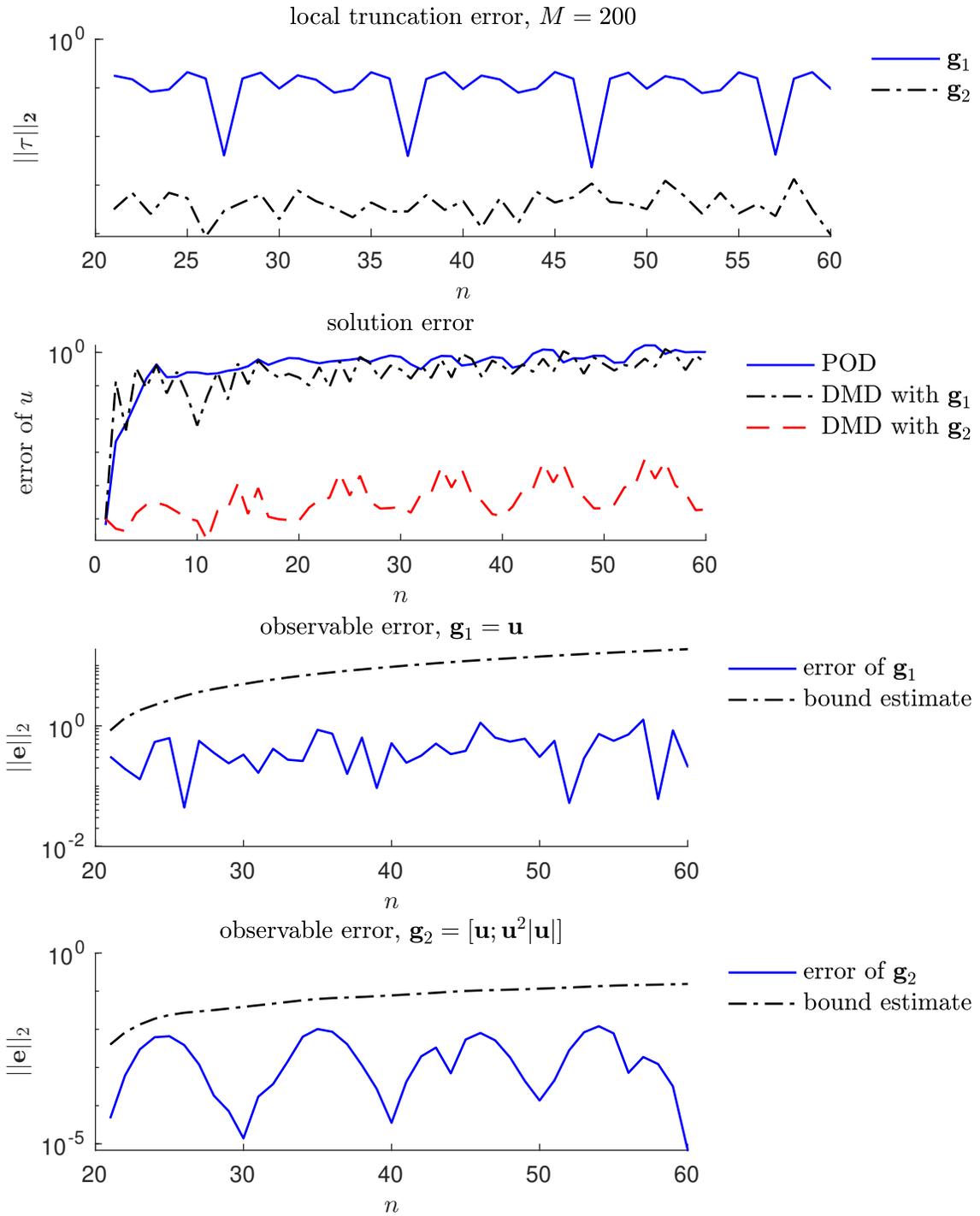


Figure 2.14: Test 4. Comparison of POD and DMD errors; Local truncation error and global error for DMD prediction with  $\mathbf{g}_1$  and  $\mathbf{g}_2$  using  $M = 20$  snapshots.

We reproduce the results reported in [58] and use them to verify our error bound in Figure 2.13 and Figure 2.14. In this case, DMD with the right observable has better performance, in terms of both accuracy and efficiency, than POD. The advantage of taking physical information into account is tremendous.

## 2.3 Summary

We derived error bounds of DMD predictions for linear and nonlinear parabolic PDEs and verified their accuracy on four computational examples of increasing degree of complexity. Our analysis leads to the following major conclusions.

1. When combined with an adequate choice observables, the Koopman operator maps the nonlinear underlying dynamics with the linear observable space, where DMD algorithm can be implemented with good accuracy and efficiency.
2. In the extrapolation (predictive) mode, DMD outperforms other ROM-based method (e.g., POD) in terms of computational efficiency, because it requires no iteration. At the same time, POD has higher predictive accuracy than DMD.
3. Our error estimator is consistent with previous theoretic understanding of DMD algorithm and the Koopman operator theory. More importantly, it provides a quantitative measure of the accuracy of DMD predictions.

In the follow-up studies we will use our error estimators of DMD predictions to address several challenges in scientific computing:

1. For PDEs with random coefficients, e.g., for PDE-based models of flow and transport in (randomly) heterogeneous porous media, DMD predictions with quantitative error bounds might provide a means for accelerating computational expensive Monte Carlo and multiscale simulations.
2. Our error estimators can be used to guide the design of hybrid algorithms that combine DMD predictions with fully resolved solutions of multi-dimensional complex problems.

3. It might be possible to generalize our results to a broader context of advection-diffusion equations. Multi-resolution DMD (mrDMD), instead of DMD, can be used to overcome the translational invariant issues in advection.

# Chapter 3

## Lagrangian DMD for Advection-Dominated Phenomena

Advection-diffusion equations are routinely used as a high-fidelity representation of mass conservation at a variety of spatiotemporal scales in a plethora of applications [119]. These equations become highly nonlinear when advection velocity and/or diffusion coefficient depend(s) on a system state, e.g., in the case of multiphase flows in porous media [119]. High-dimensional complex dynamics described by such nonlinear advection-diffusion equations often possess low-dimensional structures, suggesting the possibility of their replacement with ROMs [21, 120, 121].

SVD can be utilized to extract a low-dimensional structure from the data generated with a HFM, i.e., to construct a conventional ROM. Examples of such ROMs are built by POD and DMD. While the robustness of DMD for parabolic problems has been established with analysis of its accuracy from Chapter 2, both DMD and POD are known to fail in translational problems, such as wave-like phenomena, moving interfaces and moving shocks [23]. It can be explained by the intuition that the dominating advection behavior is traveling through the whole high-dimensional domain, making it impossible to determine a global spatiotemporal basis confined in a low-dimensional subspace. We facilitate this intuitive explanation with a concrete example in section 3.1. In terms of the Koopman operator theory, important physical observables (e.g., advection speed, shock speed, shock formation time) are

unaccounted for in the standard DMD algorithm. Remedies for POD include the deployment of local basis [122], domain decomposition [123], or basis splitting [124]. A similar extension of DMD consists of multi-resolution DMD [72], which separates frequencies of different scales by filtering windows. Unfortunately, these remedies often compromise the ROM's efficiency by increasing its computational complexity. Alternative generalizations of DMD and POD explore symmetry and self-similarity properties to eliminate the translational issue using analytical tools [73, 125–129]. However, such tools are usually problem-dependent and mostly applicable to single-wave dominated problems.

Motivated by the recent work on Lagrangian POD [111], we propose a Lagrangian-based DMD method to construct an efficient ROM. This approach is consistent with the spirit of physics-aware DMD since it accounts for the evolution of characteristic lines. Several computational experiments are used to validate the efficiency of the proposed approach to ROM construction, with comparison with Lagrangian POD in terms of accuracy and computational costs.

For demonstration purposes, we consider a group of PDEs belonging to (1.1) with a scalar state variable  $u(x, t) : [a, b] \times [0, T] \rightarrow \mathbb{R}$ , whose dynamics is described by a one-dimensional nonlinear advection-diffusion equation

$$\frac{\partial u}{\partial t} + c(u) \frac{\partial u}{\partial x} = \frac{\partial}{\partial x} \left( D(x, t, u) \frac{\partial u}{\partial x} \right), \quad c(u) = \frac{\partial C(u)}{\partial u}, \quad (3.1a)$$

subject to the initial condition

$$u(x, t = 0) = u_0(x) \quad (3.1b)$$

and appropriate (arbitrary) boundary conditions at  $x = a$  and  $x = b$ . Among other phenomena, (3.1) describes multiphase flow in porous media, and mass or heat transfer due to convection and diffusion/dispersion. In the former context, the state variable  $u(x, t)$  denotes the saturation of a porous medium with one of the fluids (e.g., water, oil, or  $\text{CO}_2$ ), the nonlinear flux term  $C$  accounts for the viscous and gravity effects on fluid flow, and the nonlinear diffusion term on the right-hand-side provides

a macroscopic (Darcy-scale) representation of the capillary forces (e.g., [130] and the references therein). In the latter context,  $u(x, t)$  represents solute concentration in, or temperature of, a fluid whose advection/convection velocity and diffusion/dispersion flux vary with  $u$  (e.g., [131] and the references therein).

Within the Eulerian framework, the space is fixed and the interval  $[a, b]$  is discretized with a uniform grid  $\mathbf{x} = [x_1 = a, x_2 \cdots, x_{N_{\mathbf{x}}-1}, x_{N_{\mathbf{x}}} = b]^\top$  of mesh size  $\Delta x \equiv x_{j+1} - x_j = (b - a)/N_{\mathbf{x}}$  and  $N_{\mathbf{x}}$  nodes. Likewise, the time interval  $[0, T]$  is discretized uniformly with time step  $\Delta t \equiv t_{n+1} - t_n = T/N_t$  and  $N_t + 1$  nodes so that  $t^0 = 0 < t_1 < \cdots < t_{N_t} = T$ . At the  $n$ th time node, the state variable  $u(x, t)$  is represented by a vector  $\mathbf{u}^n = [u_1^n, \cdots, u_{N_{\mathbf{x}}}^n]^\top$  for  $n = 0, \cdots, N_t$ . For simplicity, (3.1) is solved with a conservative first-order finite-difference upwind scheme with forward Euler for the advection part and finite center difference with backward Euler for the diffusion part,

$$\begin{aligned} u_j^{n+1} = & u_j^n - \frac{\Delta t}{\Delta x} (C_{j+1/2}^n - C_{j-1/2}^n) \\ & + \frac{\Delta t}{(\Delta x)^2} \left[ D_{j+1/2}^{n+1} (u_{j+1}^{n+1} - u_j^{n+1}) - D_{j-1/2}^{n+1} (u_j^{n+1} - u_{j-1}^{n+1}) \right], \end{aligned} \quad (3.2)$$

where

$$\begin{aligned} C_{j+1/2}^n &= \frac{C(u_{j+1}^n) + C(u_j^n)}{2} - |\alpha_{j+1/2}^n| \frac{u_{j+1}^n - u_j^n}{2}, \\ \alpha_{j+1/2}^n &= \begin{cases} \frac{C_{j+1}^n - C_j^n}{u_{j+1}^n - u_j^n} & \text{if } u_{j+1}^n \neq u_j^n, \\ c(u_j) & \text{if } u_{j+1}^n = u_j^n, \end{cases} \\ D_{j+1/2}^{n+1} &= \frac{1}{2} (D_j^{n+1} + D_{j+1}^{n+1}). \end{aligned}$$

In vector form as (1.4), the above scheme reads

$$\mathbf{R}(\mathbf{u}^{n+1}) = \mathbf{u}^{n+1} - \mathbf{u}^n + \Delta t (\mathcal{D}_1^u \mathbf{C}^n) - \Delta t (\mathcal{D}_2 \mathbf{u}^{n+1}) = 0, \quad (3.3)$$

where  $\mathcal{D}_1^u \in \mathbb{R}^{N_{\mathbf{x}} \times N_{\mathbf{x}}}$  and  $\mathcal{D}_2 \in \mathbb{R}^{N_{\mathbf{x}} \times N_{\mathbf{x}}}$  are discrete approximations of the first derivative (using upwind) and second derivative (using center difference), respectively. Here  $\mathbf{C}^n = [C_{1/2}^n, \cdots, C_{N_{\mathbf{x}}-1/2}^n]^\top$  and  $\mathbf{R}$  is the vectorized residual of the scheme. Certain

CFL condition needs to be satisfied to ensure the stability of the scheme depending on the functional forms of  $c$  and  $D$ . Simulation results obtained with the above method constitute our HFM.

### 3.1 Challenge Posed by Translational Problems

Both POD and DMD have been used to construct LFMs for a wide range of problems with high accuracy. However, ROMs constructed with such SVD-based methods are known to have poor performance for translational problems, such as an advection-dominated version of (3.1). To illustrate this phenomenon, we consider a linear advection-diffusion equation, i.e., (3.1) with constant  $c$  and  $D$ , defined on  $(x, t) = [0, 2] \times [0, 1]$ . This equation is subject to the initial condition  $u(x, 0) = 0.5 \exp[-(x - 0.3)^2/0.05^2]$  and the boundary conditions  $u(0, t) = u(2, t) = 0$ . The space domain  $[0, 2]$  is discretized into  $N_x = 2000$  intervals and time domain  $[0, 1]$  is discretized into  $N_t = 1000$  steps. Both DMD and POD algorithms use the same dataset consisting of  $M = 250$  snapshots.

To achieve a diffusion-dominated regime, we set  $c = 10^{-4}$  and  $D = 10^{-2}$  in some consistent units. Figure 3.1 provides a visual comparison of the reference solution with its counterparts obtained with DMD and POD, both with  $r = 20$  SVD rank truncation. Although not shown here, and consistent with the earlier findings reported in Chapter 2 and [2], the DMD- and POD-based LFMs are of spectral accuracy in a relatively small subspace of time ( $t < 0.3$ ), with the relative error increasing with time. POD has slightly better accuracy than DMD due to the iterations in the subspace, but DMD is considerably faster because of its iteration-free nature.

An advection-dominated regime is achieved by setting  $c = 1.0$  and  $D = 10^{-3}$ . Figure 3.2 reveals that both DMD and POD fail to capture the system dynamics, yielding unphysical (oscillatory and negative) predictions. Note that even accurate DMD- or POD-based ROMs are not guaranteed to be positivity-preserving in general because of the rank truncation. But unphysical negative predictions, such as those presented in Figure 3.2, indicate the unambiguous failure of a ROM. This failure cannot be remedied by increasing the SVD rank truncation  $r$ : increasing  $r$  from 20

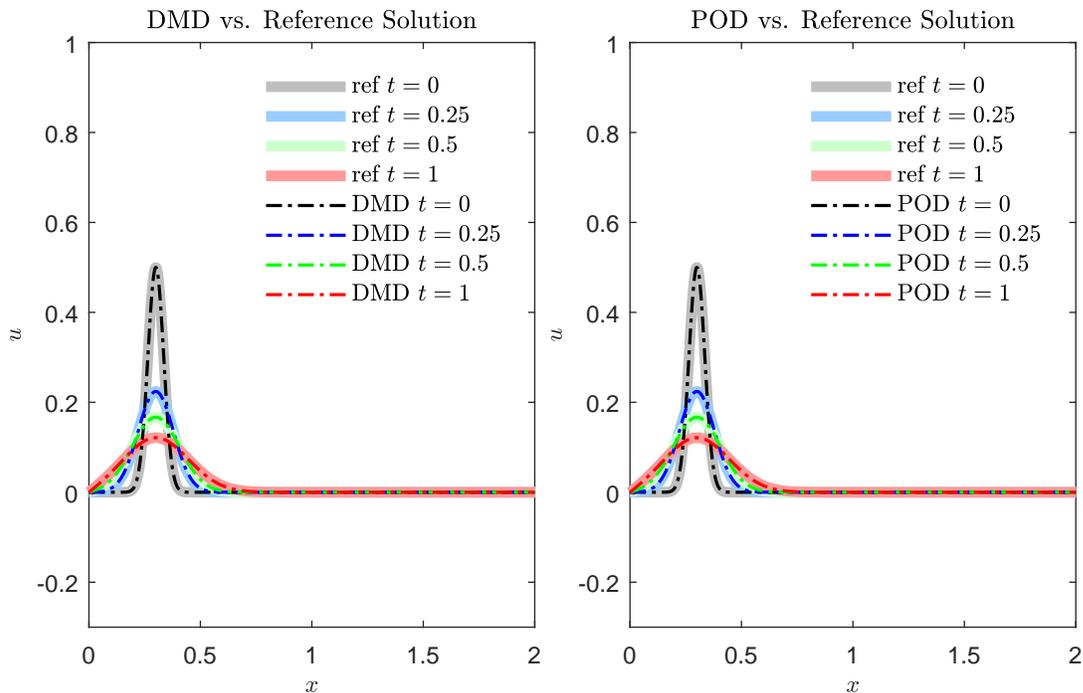


Figure 3.1: Solution profiles  $u(x, \cdot)$ , for several times  $t$ , in the diffusion-dominated regime. These profiles are computed with DMD (left) and POD (right), and compared with the reference solution.

to 30 does not improve the prediction's accuracy, either quantitatively or qualitatively. These results highlight the main challenge translational problems pose for the SVD-based methods. Given the first 250 snapshots of the high-fidelity solution, SVD extracts dominant DMD/POD modes  $\phi_i$  from the region the wave has encountered; in our example, the subdomain  $[0, 1]$ . As time increases, the wave solution encounters other parts of the computational domain; in our example, at later times the dominant signal lies mostly in the subdomain  $[1, 2]$ . Specifically, one can observe that the dominant DMD/POD modes have fluctuations only in the subdomain  $[0, 1]$  and stay flat 0 in the subdomain  $[1, 2]$  in Figure 3.3. It is therefore not surprising that a ROM constructed from dominant modes in  $[0, 1]$  does not serve as an accurate surrogate for the rest of the computational domain.

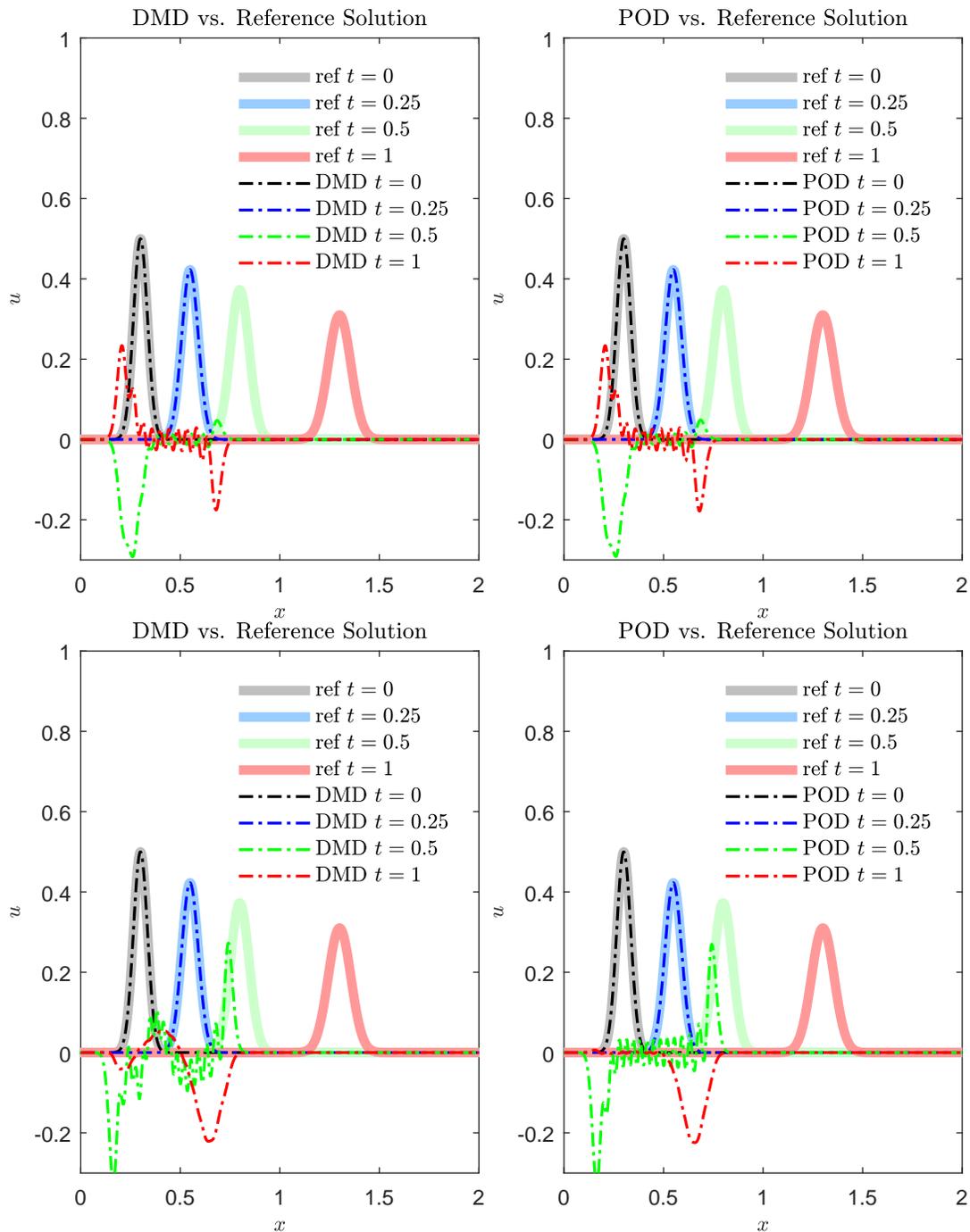


Figure 3.2: Solution profiles  $u(x, \cdot)$ , for several times  $t$ , in the advection-dominated regime. These profiles are computed with DMD (left column) and POD (right column) using the SVD rank of  $r = 20$  (top row) and  $r = 30$  (bottom row), and compared with the reference solution.

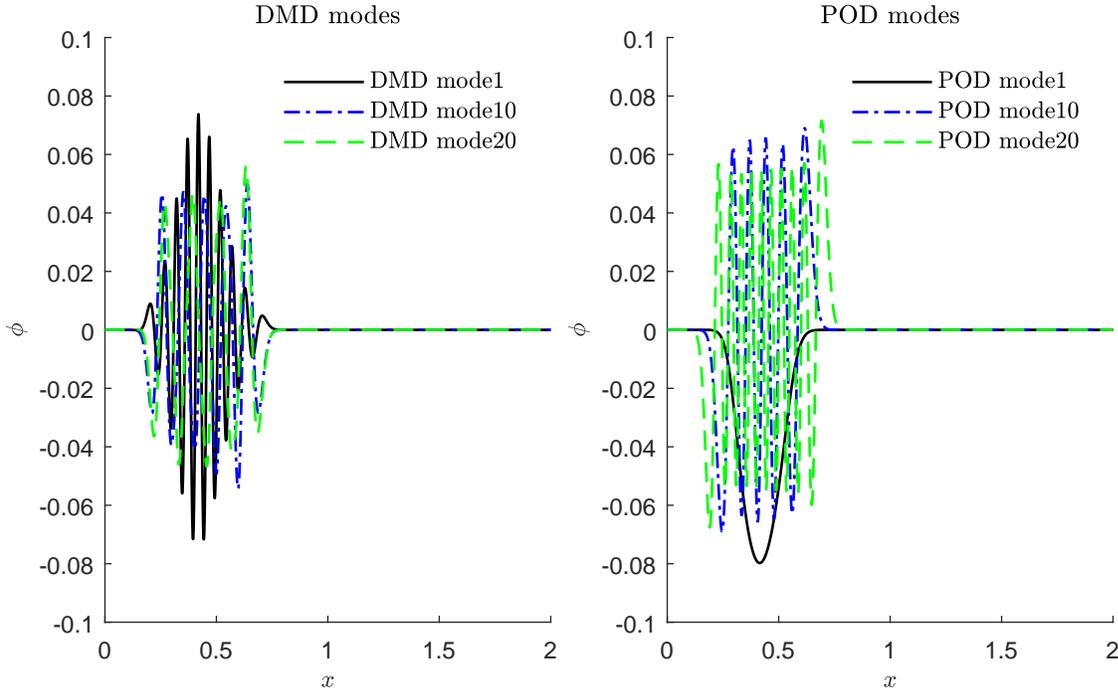


Figure 3.3: Three of the dominant DMD modes (left column) and POD modes (right column) extracted from the first 250 snapshots.

## 3.2 Lagrangian Reduced-Order Models

Motivated by construction of a POD-based ROM for the advection-diffusion equation (3.1) within the Lagrangian framework [111], we propose a Lagrangian DMD. In the semi-Lagrangian frame, (3.1) is written as

$$\begin{cases} \frac{dX(t)}{dt} = c(u(X(t), t)), \\ \left. \frac{du(x, t)}{dt} \right|_{x=X(t)} = \left[ \frac{\partial}{\partial x} \left( D(x, t, u) \frac{\partial u(x, t)}{\partial x} \right) \right] \Big|_{x=X(t)}. \end{cases} \quad (3.4)$$

Applying a first-order discretization to this system gives

$$\begin{cases} \tilde{u}_j^n = \mathcal{P}_0(u_j^n), \\ \tilde{u}_j^{n+1} = \tilde{u}_j^n + \Delta t \frac{D_{j+1/2}^{n+1}(\tilde{u}_{j+1}^{n+1} - \tilde{u}_j^{n+1}) - D_{j-1/2}^{n+1}(\tilde{u}_j^{n+1} - \tilde{u}_{j-1}^{n+1})}{(\Delta x)^2}, \\ u_j^{n+1} = \mathcal{P}_n(\tilde{u}_j^{n+1}), \\ x_j^{n+1} = x_j^n + \frac{\Delta t}{2}(c(u_j^n) + c(u_j^{n+1})), \end{cases} \quad (3.5)$$

where  $\mathcal{P}_n$  stands for the interpolation in the grid  $\bar{\mathbf{x}}^n = [x_1^n, \dots, x_{N_x}^n]^\top$  and  $\bar{\mathbf{x}}^0$  is the starting uniform grid.

Or, in vector form,

$$\begin{cases} \mathbf{R}_x(\bar{\mathbf{x}}^{n+1}) \equiv \bar{\mathbf{x}}^{n+1} - \bar{\mathbf{x}}^n - \frac{\Delta t}{2}(\mathbf{c}(\mathbf{u}^n) + \mathbf{c}(\mathbf{u}^{n+1})) = \mathbf{0}, \\ \mathbf{R}_u(\tilde{\mathbf{u}}^{n+1}) \equiv \tilde{\mathbf{u}}^{n+1} - \tilde{\mathbf{u}}^n - \Delta t \mathcal{D}_2 \tilde{\mathbf{u}}^{n+1} = \mathbf{0}. \end{cases} \quad (3.6)$$

Here  $\bar{\mathbf{x}}^n = [x_1^n, \dots, x_{N_x}^n]^\top$  denotes the locations of the Lagrangian computational grid at the  $n$ th time step,  $\tilde{\mathbf{u}}^n$  is the solution on the Lagrangian grid  $\bar{\mathbf{x}}^n$ ,  $\mathbf{u}^n$  is the linear interpolation from the Lagrangian grid to the Eulerian grid, and  $\mathcal{D}_2$  represents the discrete approximation of the second derivative on the uniform Eulerian grid at the  $n$ th time step.

### 3.2.1 Lagrangian POD

We arrange  $M$  snapshots of the HFM in the Lagrangian framework into a data matrix of size  $2N_x \times M$ ,

$$\mathbf{X} = \begin{bmatrix} | & | & \dots & | \\ \bar{\mathbf{x}}^0 & \bar{\mathbf{x}}^2 & \dots & \bar{\mathbf{x}}^{M-1} \\ | & | & \dots & | \\ \tilde{\mathbf{u}}^0 & \tilde{\mathbf{u}}^2 & \dots & \tilde{\mathbf{u}}^{M-1} \\ | & | & \dots & | \end{bmatrix}. \quad (3.7)$$

Applying the conventional POD of section 1.1.1 to the data matrix in (3.7), one obtains a POD basis  $\Phi$  analogous to (1.8) for the space-solution vector  $[\bar{\mathbf{x}}; \tilde{\mathbf{u}}]^\top$ . Then the Lagrangian solution is approximated by

$$\begin{bmatrix} | \\ \bar{\mathbf{x}}_{\text{POD}}^{n+1} \\ | \\ \tilde{\mathbf{u}}_{\text{POD}}^{n+1} \\ | \end{bmatrix} = \Phi \begin{bmatrix} | \\ \hat{\mathbf{x}}^{n+1} \\ | \\ \hat{\mathbf{u}}^{n+1} \\ | \end{bmatrix}. \quad (3.8)$$

Inserting (3.8) into (3.6) and projecting onto the subspace spanned by  $\Phi$ , one obtains the solution vector  $[\bar{\mathbf{x}}^{n+1}; \tilde{\mathbf{u}}^{n+1}]^\top$  by solving the following equation:

$$\Phi^T \mathbf{R} \left( \Phi \begin{bmatrix} | \\ \hat{\mathbf{x}}^{n+1} \\ | \\ \hat{\mathbf{u}}^{n+1} \\ | \end{bmatrix} \right) = 0. \quad (3.9)$$

Several complications can arise when applying Lagrangian POD in practice. If only Eulerian HFM data are available, i.e., in the absence of the grid deformation  $\bar{\mathbf{x}}^n$  computed with an Eulerian HFM, one can construct an optimal Lagrangian basis by following the strategy proposed in [111, Sec. 3.3]. Another potential complication is a Lagrangian grid entanglement. There is no guarantee that an approximation of the Lagrangian moving grid in the low-dimensional subspace preserves the topological properties of the original HFM simulation. In many cases, e.g., when characteristic lines intersect each other, the Lagrangian grids in the projected space are severely distorted, inducing numerical instabilities. One strategy for ameliorating this problem is to solve the diffusion step back to stationary Eulerian grid by interpolation between the Eulerian and Lagrangian grids [111, Sec. 3.4]. This procedure can reduce the method's efficiency and accuracy.

### 3.2.2 Lagrangian DMD

The fundamental concept behind the Koopman operator theory is to transform the finite-dimensional nonlinear problem (1.3) in the state space into the infinite-dimensional linear problem (1.26) in the observable space. Compared to Eulerian framework, Lagrangian framework contains more informative physical quantities as candidates of the observables, making Lagrangian DMD to fit intuitively and naturally into the Koopman operator theory. Since  $\mathcal{K}_{\Delta t}$  in Definition 1.2 is an infinite-dimensional linear operator, it is equipped with infinitely many eigenvalues  $\{\lambda_i\}_{i=1}^{\infty}$  and eigenfunctions  $\{\phi_i\}_{i=1}^{\infty}$ . In practice, one deals with a finite number of the eigenvalues and eigenfunctions. Assumption 1.1 underpins the finite approximation and is essential to the choice of observables.

In data-driven modeling, judicious selection of the observables is crucial to the accuracy and efficiency of a Koopman operator's approximation. Identification of general rules for choosing the observables continues to be a subject of ongoing research. For example, the use of measurements of the state variable  $u(x, t)$  as an observable led to the poor DMD performance in the advection-dominated regime (Figure 3.2). A Lagrangian formulation of the problem provides a means of identification of optimal observables. Indeed, the physics of advection-dominated systems suggests that the location of a moving particle is a key quantity, which is as important as the value of the state variable at that location. It is therefore natural to introduce an observable function that keeps track of both. Thus we choose our observable to be

$$\mathbf{y}^k = \mathbf{g}(\mathbf{u}^k) = \begin{bmatrix} \bar{\mathbf{x}}^k; \tilde{\mathbf{u}}^k \end{bmatrix}. \quad (3.10)$$

Then, we follow Algorithm 2.

## 3.3 Numerical Tests of Lagrangian DMD

To ascertain the accuracy and robustness of the Lagrangian DMD, we use it to construct ROMs for a series of linear and nonlinear advection-dominated problems. In all tests, the reference solutions are computed in the Eulerian framework using (3.2).

The space domain  $[0, 2]$  is discretized into  $N_x = 2000$  intervals and the time domain  $[0, 1]$  is discretized into  $N_t = 1000$  steps. Both Lagrangian DMD and Lagrangian POD algorithms use  $M = 250$  snapshots (up to  $t = 0.25$ ) as a training dataset. The rank truncation criteria (1.7) with  $\varepsilon = 10^{-8}$  is used. Define

$$\mathbf{e}^n = \mathbf{y}^n - \mathbf{y}_{\text{DMD}}^n. \quad (3.11)$$

The error bound derived in Theorem 2.2 of Chapter 2 is plotted in each example as an estimate of the observable.

### 3.3.1 Linear Advection Equation

We start by considering (3.1) with  $c \equiv 1$  and  $D \equiv 0$ . The resulting linear advection equation is defined on  $(x, t) \in (0, 2) \times (0, 1]$ , and is subject to the initial condition

$$u(x, t = 0) = u_0(x) \equiv \frac{1}{2} \exp \left[ - \left( \frac{x - 0.3}{0.05} \right)^2 \right] \quad (3.12a)$$

and boundary conditions

$$u(0, t) = u(2, t) = 0. \quad (3.12b)$$

Figure 3.4 provides a visual comparison between the reference solution, obtained with the numerical scheme (3.2), and solutions of the ROMs constructed with either Lagrangian DMD or Lagrangian POD. Unlike their conventional (Eulerian) counterparts (see Figure 3.2), both Lagrangian DMD and Lagrangian POD capture the solution dynamics in the extrapolating mode, i.e., for  $t > 0.25$ .

A more quantitative comparison of the relative performance of the two SVD-based strategies is presented in Figure 3.5 in terms of the global truncation error defined in (3.11). Both Lagrangian DMD and Lagrangian POD capture the advection with high accuracy. Due to the linearity and conservation property of this problem, the ROMs constructed by the two methods are of machine error. Thus, the error bound developed in [2] is not tight but sufficient to serve as an indicator of successful approximation.

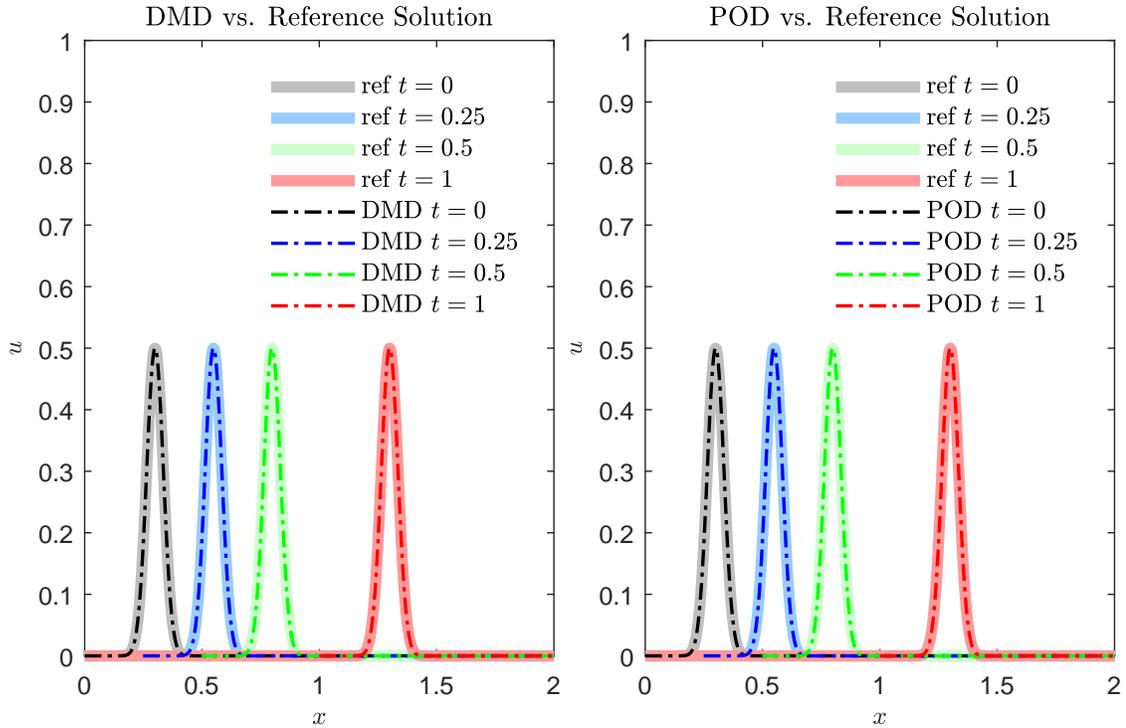


Figure 3.4: Solutions of the linear advection equation,  $u(x, t)$ , alternatively obtained with the numerical method (3.2) and the ROMs constructed via Lagrangian DMD (left) and Lagrangian POD (right).

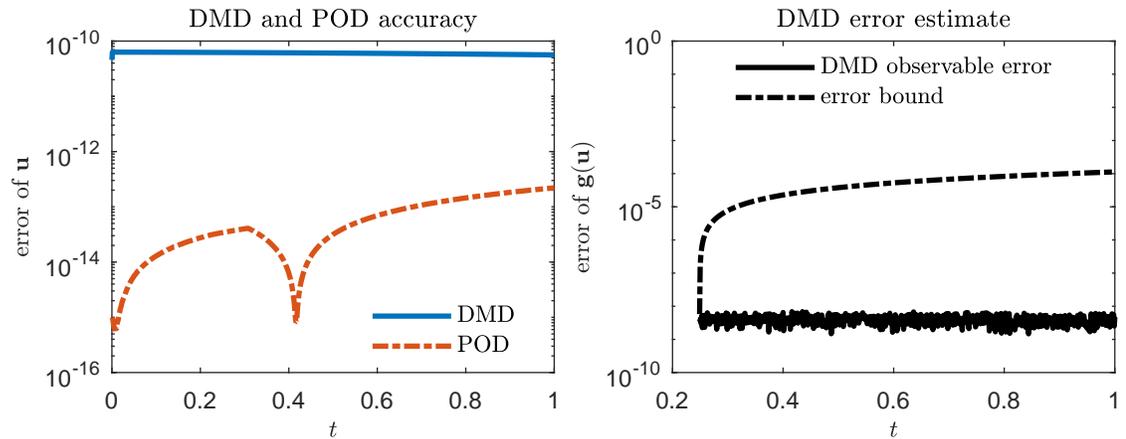


Figure 3.5: Errors of the Lagrangian DMD- and POD-based ROMs for the linear advection equation: error in reconstructing the state variable  $u(x, t)$  (left) and its observables  $\mathbf{g}(\mathbf{u})$  (right). The error bound for  $\mathbf{g}(\mathbf{u})$  is derived in [2].

### 3.3.2 One-Dimensional Advection-Diffusion Equation

We consider (3.1) with  $c \equiv 1$  and  $D \equiv 0.01$ . The resulting linear advection-diffusion equation is defined on  $(x, t) \in (0, 2) \times (0, 1]$ , and is subject to the initial and boundary conditions (3.12). The choice of the parameter values  $c$  and  $D$  ensures that the system is in the advection-dominated regime, for which the conventional POD and DMD fail.

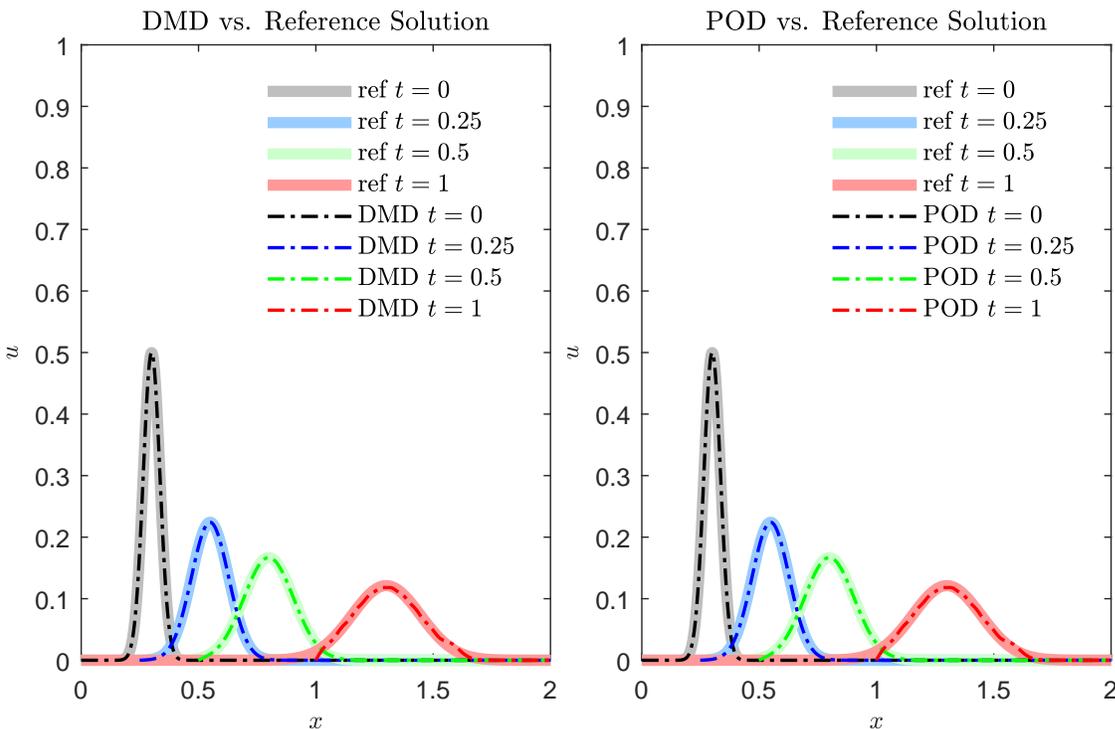


Figure 3.6: Solutions of the linear advection-diffusion equation,  $u(x, t)$ , alternatively obtained with the numerical method (3.2) and the ROMs constructed via Lagrangian DMD (left) and Lagrangian POD (right).

Figure 3.6 provides a visual comparison between the reference solution  $u(x, t)$  and those obtained with the ROMs. The latter capture the system’s dynamics, although their estimates of the solution tails become less accurate with time. This suggests that Lagrangian DMD and POD are capable of identifying the low-rank structure of the advection-diffusion dynamics in the advection-dominated regime.

Figure 3.7 indicates that the Lagrangian DMD and POD have a near identical accuracy, which deteriorates with extrapolation time  $t > 0.25$ . The error bound for

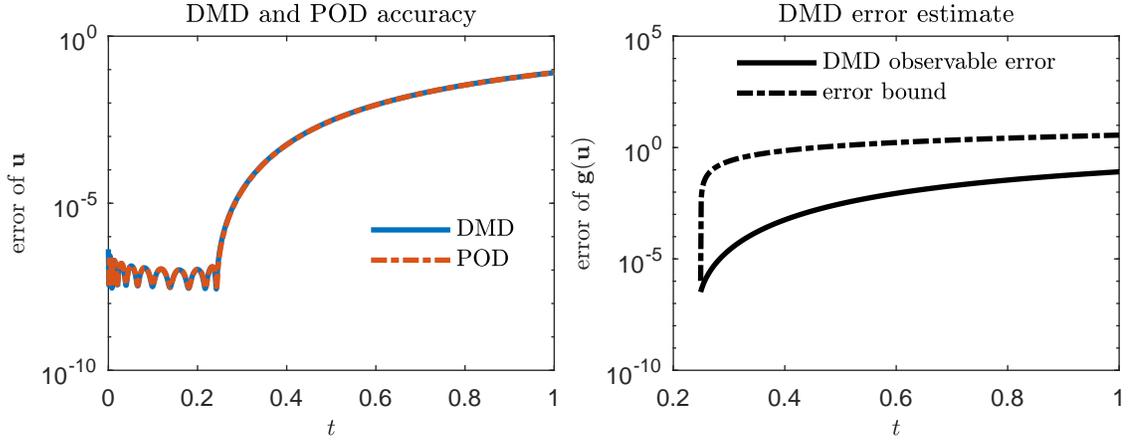


Figure 3.7: Errors of the Lagrangian DMD- and POD-based ROMs for the linear advection equation: error in reconstructing the state variable  $u(x, t)$  (left) and its observables  $\mathbf{g}(\mathbf{u})$  (right). The error bound for  $\mathbf{g}(\mathbf{u})$  is derived in [2].

the DMD estimate of the observable  $\mathbf{g}(\mathbf{u})$  is appreciably tighter than in the case of advection (Figure 3.5). With the error bounds, one can design an algorithm combining short-term computation of HFM with long-term computation of LFM.

### 3.3.3 Two-Dimensional Advection-Diffusion Equation

Consider a two-dimensional linear advection-diffusion problem on  $(x, y, t) \in [0, 4] \times [0, 4] \times [0, 0.5]$ ,

$$\begin{cases} \frac{\partial u}{\partial t} + c_x \frac{\partial u}{\partial x} + c_y \frac{\partial u}{\partial y} = D \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right), \\ u(x, y, t = 0) = u_0(x, y) = \exp[-((x - 1)^2 + (y - 1)^2)/0.1], \\ c_x = 0.1, c_y = 0.2, D = 0.001. \end{cases} \quad (3.13)$$

The spatial domain is discretized into  $N_x = N_y = 40$  points in the  $x$  and  $y$  directions, yielding the total of  $N_{\mathbf{x}} = 1600$  grid points. The time interval is discretized into  $N_t = 1000$  steps. Both Lagrangian DMD and Lagrangian POD use  $M = 250$  snapshots as a training dataset. The observable  $\mathbf{y}^k$  is constructed in two steps:

- Step 1: Transform one snapshot of the  $N_x$  by  $N_y$  solution matrix  $u_{i,j}^k$  into a

vector of length  $N_x \cdot N_y$ ; transform one snapshot of the  $N_x$  by  $N_y$  moving grid  $X_{i,j}^k$  into a vector of length  $N_x \cdot N_y$ .

- Step 2: Concatenate these two  $(N_x \cdot N_y)$ -long vectors into a vector  $\mathbf{y}^k$  of length  $2 \cdot N_x \cdot N_y$ .

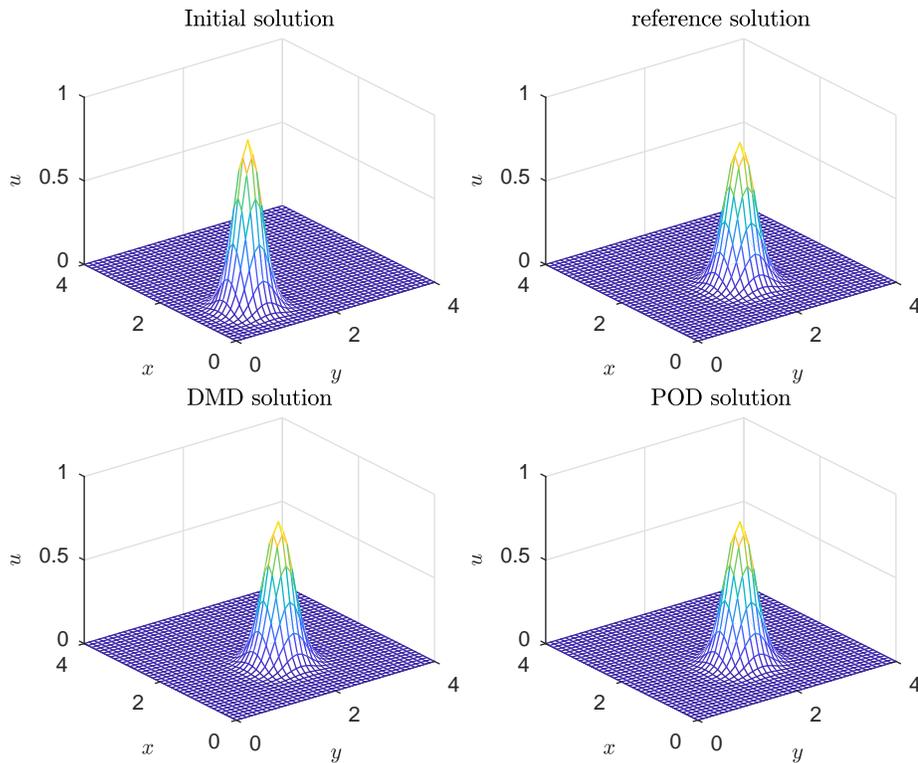


Figure 3.8: Solutions of the two-dimensional linear advection-diffusion equation,  $u(x, y, t)$ , obtained alternatively with the numerical method (3.2) and the ROMs constructed via Lagrangian DMD (left bottom) and Lagrangian POD (right bottom).

Figure 3.8 provides a visual comparison between the reference solution and those obtained with the Lagrangian DMD and the Lagrangian POD. Both ROMs capture the advection-dominated phenomena. Figure 3.9 indicates that, in the extrapolating regime, the Lagrangian POD is more accurate than the Lagrangian DMD. However, its computational cost is higher than that of the iteration-free DMD, which becomes increasingly important in two- and higher-dimensional problems. This is because the projection step (3.9) requires memory of formulating a large forward matrix; in our

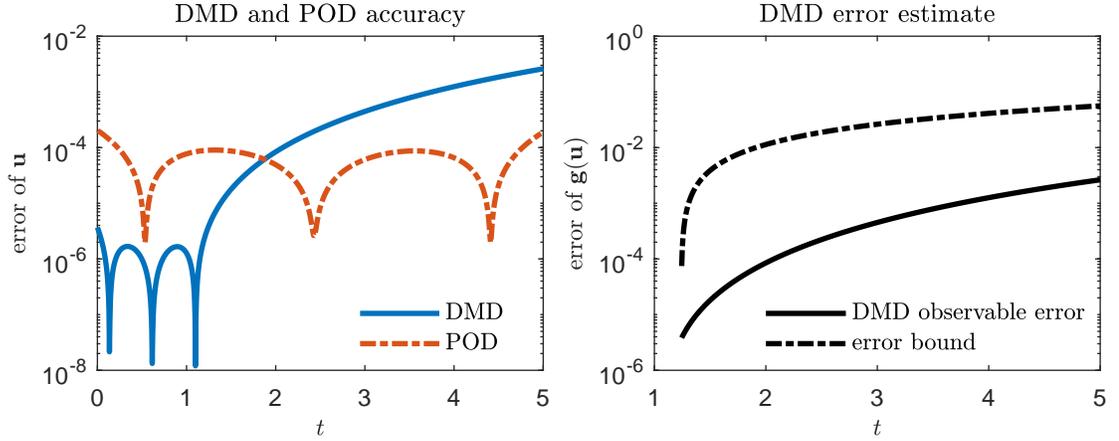


Figure 3.9: Errors of the Lagrangian DMD- and POD-based ROMs for the two-dimensional advection-diffusion equation: error in reconstructing the state variable  $u(x, y, t)$  (left) and its observables  $\mathbf{g}(\mathbf{u})$  (right). The error bound for  $\mathbf{g}(\mathbf{u})$  is derived in [2].

example, its size is  $N_x \cdot N_y$  by  $N_x \cdot N_y$ . Our error bounds enable one to monitor and control the error in a reasonable range.

### 3.3.4 Inviscid Burgers' Equation

The inviscid Burgers' equation is recovered from (3.1) by setting  $c \equiv u$  and  $D \equiv 0$ ,

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = 0. \quad (3.14)$$

We define this equation on  $(x, t) \in (0, 2\pi) \times (0, 1]$ , subject to the initial conditions

$$u(x, t = 0) = u_0(x) \equiv 1 + \sin(x) \quad (3.15a)$$

and the periodic boundary conditions

$$u(0, t) = u(2\pi, t). \quad (3.15b)$$

This formulation ensures a smooth, shock-free solution of the Burgers' equation. The current Lagrangian ROM formulation fails in the presence of shocks and will be

extended to such cases in Chapter 4.

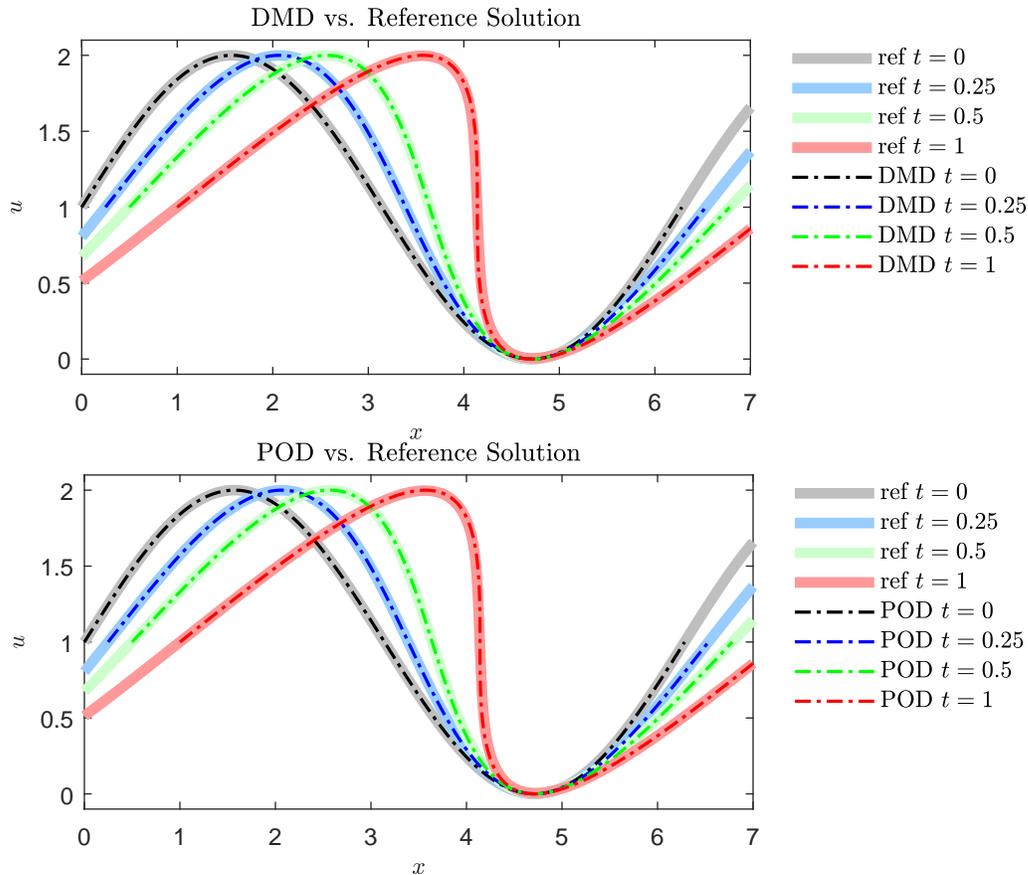


Figure 3.10: Solutions of the inviscid Burgers' equation,  $u(x, t)$ , alternatively obtained with the numerical method (3.2) and the ROMs constructed via Lagrangian DMD (top) and Lagrangian POD (bottom).

Figure 3.10 provides a graphical illustration of the Lagrangian ROMs' ability to capture the system state dynamics in this nonlinear hyperbolic problem. Since the Lagrangian description treats first-order hyperbolic conservation laws, such as the inviscid Burgers' equation, exactly via the method of characteristics, the addition of the particle trajectories  $x(t)$  to the set of observables ensures that the Lagrangian POD and DMD are both of machine error accuracy (Figure 3.11). Again the error bound serves as an indicator of accurate ROMs.

The level-set method provides an alternative way to interpret the first-order hyperbolic conservation laws. In the Appendix A, we report our experiments with the

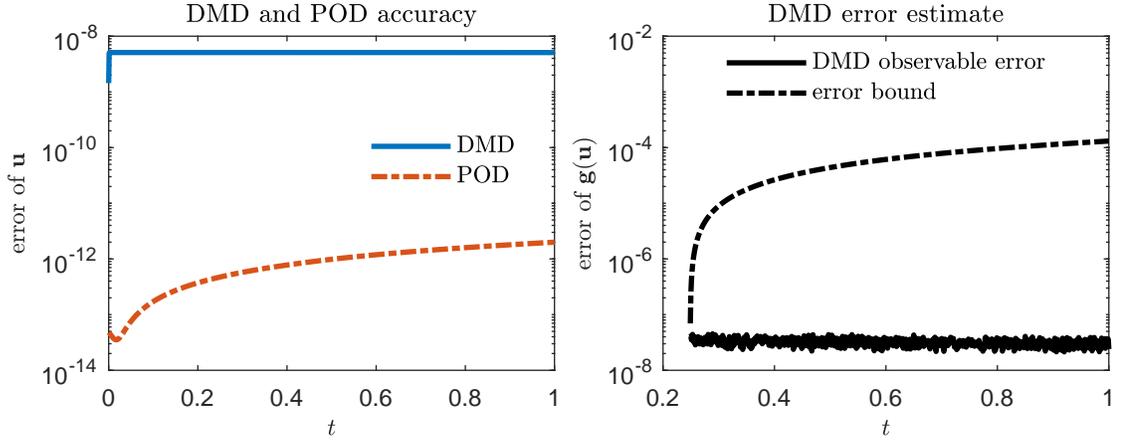


Figure 3.11: Errors of the Lagrangian DMD- and POD-based ROMs for the inviscid Burgers' equation: error in reconstructing the state variable  $u(x, t)$  (left) and its observables  $\mathbf{g}(\mathbf{u})$  (right). The error bound for  $\mathbf{g}(\mathbf{u})$  is derived in [2].

level-set DMD, which is essentially a Lagrangian DMD for two-dimensional linear advection equation.

### 3.3.5 Viscous Burgers' Equation

The viscous Burgers' equation is obtained from (3.1) by setting  $c \equiv u$  and  $D \equiv 0.1$ . Again, this equation is defined on  $(x, t) \in (0, 2\pi) \times (0, 1]$  and is subject to the initial and boundary conditions (3.15).

For this nonlinear problem, Lagrangian DMD is visually more accurate than Lagrangian POD (Figure 3.12), especially at later times. This is confirmed by plotting the error in Figure 3.13. As mentioned in [132], the Lagrangian grid might become distorted (especially in the presence of large gradients  $\partial_x u$ ) during the compressing process of ROM in the POD algorithm. The error estimate of the observable does a good job in evaluating the bound.

### 3.3.6 Computational Costs

Table 3.1 collates the rank of the ROMs and the computational times of the HFM and the Lagrangian DMD- and POD-based ROMs. (The simulations were carried

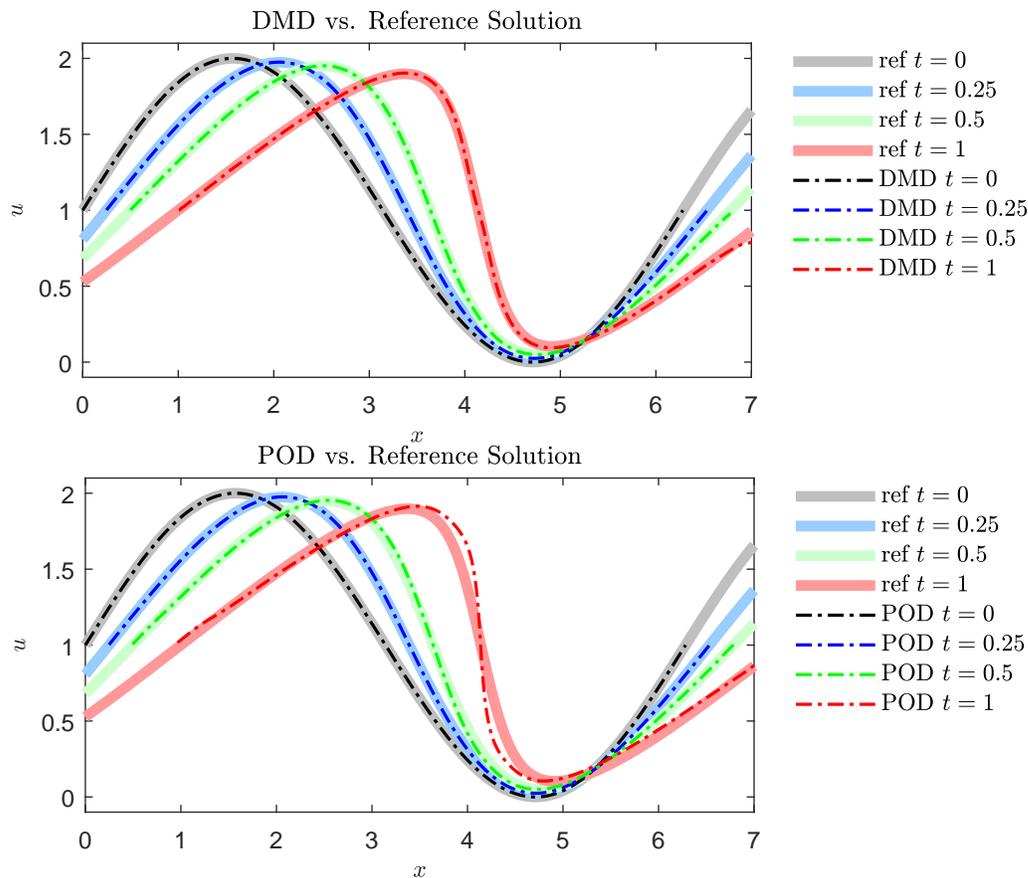


Figure 3.12: Solutions of the viscous Burgers' equation,  $u(x, t)$ , alternatively obtained with the numerical method (3.2) and the ROMs constructed via Lagrangian DMD (top) and Lagrangian POD (bottom).

out on a machine with Intel(R) Core(TM) i7 – 6700 at 3.40 GHz processor.) Test 1 refers to advection problem (in section 3.3.1); Test 2 to advection-diffusion problem (in section 3.3.2); and Tests 3 and 4 to inviscid and viscous Burgers' equations (in section 3.3.4 and section 3.3.5), respectively. In some cases, the SVD dominates the computational time of the ROM. Once the basis is constructed, the computation in the low rank subspace is much faster. This explains why the POD-based ROM of Test 3 takes more time to compute than the HFM. In other cases, the ROMs are much more efficient than the HFM computations. DMD is the most efficient methods because of its iteration-free nature.

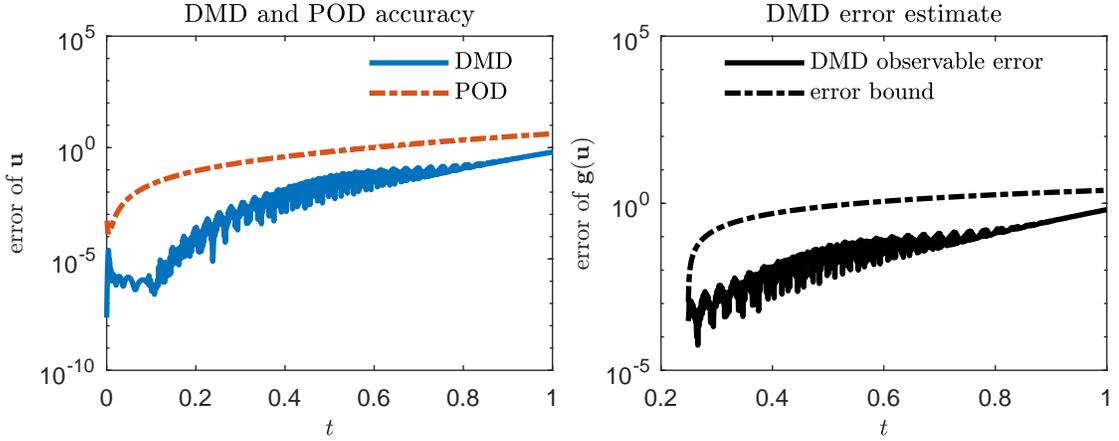


Figure 3.13: Errors of the Lagrangian DMD- and POD-based ROMs for the viscous Burgers' equation: error in reconstructing the state variable  $u(x, t)$  (left) and its observables  $\mathbf{g}(\mathbf{u})$  (right). The error bound for  $\mathbf{g}(\mathbf{u})$  is derived in [2].

	Test 1	Test 2	Test 3	Test 4
Rank truncation $r$	3	10	3	14
DMD	0.114718	0.046531	0.048450	0.055641
POD	0.153869	0.320905	0.435342	0.795566
Eulerian HFM	1.390459	29.782079	0.034519	55.713132
Lagrangian HFM	0.023568	27.020414	0.039063	55.246262

Table 3.1: Computational time (sec) of the HFMs and the corresponding Lagrangian DMD- and POD-based ROMs.

### 3.4 Summary

In this paper, we investigate the issue of translational problem for conventional POD and DMD in the Eulerian framework. A new physics-aware DMD, based on the Lagrangian framework, is proposed to overcome the shortcomings of ROMs of advection-dominated nonlinear phenomena. Characteristic lines, an important physical quantity in such systems, are taken into account in order to learn the Koopman operator of the underlying dynamics. The Lagrangian framework provides an optimal choice of observable functions for learning the Koopman operator. It allows one to construct a ROM in a relatively small subspace by using the DMD algorithm with satisfactory accuracy. Compared to the Lagrangian POD method, physics-aware DMD is more

efficient computationally thanks to its iteration-free nature.

One possible direction for future work is to investigate the advection-diffusion system in Lagrangian coordinates [133]. Interpolation between Eulerian grid and Lagrangian grid will not be needed anymore but careful discretization of the diffusion operator will need to be handled. Existing numerical studies in Lagrangian coordinates and related methods [134, 135] could be explored as guidelines of choosing physical observables in reduced order modeling.

All the numerical tests presented in this paper are shock-free. Once shock is formed, the Lagrangian formulation (3.4) becomes invalid. Although one can still make the scheme (3.5) work by numerical remedies, instability or unphysical solutions could appear when sharp gradients or shocks occur. The instability could become more severe in the compressed low-dimensional space [132]. The modifications in [132] bypass this issue by compensating computational costs in projecting back to the Eulerian grid. From the perspective of physic-aware data-driven modeling, we realize that significant information like shock formation time, shock location and shock speed is not interpreted well enough from data. In another word, other quantities should be chosen as essential observables in order to learn the underlying Koopman operator. This line of research is pursued in [31] and presented in the next Chapter 4.

## Chapter 4

# Hodograph DMD for Hyperbolic Problems with Shocks

Since introduction of Euler equations, hyperbolic conservation laws play a significant role in gas dynamics, astrophysics, plasma, traffic flow, multiphase flow in porous media [136–140] and other fields of science and engineering. Wave-like solutions of hyperbolic equations can exhibit various rarefaction and shock behaviors, whose occurrence strongly depends on a functional form of the flux function. Discontinuity and uniqueness of such solutions pose challenges in theoretical treatment of hyperbolic conservation laws [141, 142]. Theoretical advances, such as entropy conditions and the concept of a weak solution [143, 144], ameliorate this difficulty by providing physical interpretation to these solutions. Likewise, numerical high-resolution methods have been designed to resolve nonlinearities and accurately capture shocks [145–147]. Although continued developments in scientific computing have improved the performance of high-resolution simulations, their computational cost is often too high to model complex systems at spatiotemporal resolutions and scales of interest. The cost can become prohibitive when used in the context of uncertainty quantification or data assimilation, both of which require a large number of repeated forward model runs.

ROMs provide an efficient alternative to their high-fidelity, physics-based counterparts that can be deployed in large-scale multiphysics simulations. Robust tools for construction of ROMs for problems described by ODEs or PDEs include POD [19–21]

and DMD [1, 22, 112, 113]. The challenge of extending these techniques to hyperbolic or advection-dominated parabolic PDEs with smooth solutions was met by [30] through development of the physics-aware DMD and POD approaches within a Lagrangian framework. However, in the presence of strong shocks and/or sharp gradients, Lagrangian POD methods can generate numerical instability caused by grid distortion [111]. Once characteristics of a nonlinear hyperbolic PDE intersect each other, the projection from a high-dimensional manifold of the HFM onto a low-dimensional subspace of the LFM, e.g., ROM, is not guaranteed and typically fails to preserve topological properties of the original HFM. We elaborate on this point in section 4.1, in terms relevant to DMD.

We use hodograph transformation [148] to resolve this outstanding issue in construction of ROMs for PDEs with discontinuous solutions and shocks. Hodograph diagrams have originated in meteorology to plot wind from soundings of Earth's atmosphere. Since then, hodograph transformation morphed into a technique designed to transform nonlinear PDEs into linear ones by interchanging the dependent and independent variables. Hodograph-type transformations have been used to find quasilinear analogues of semi-linear equations, and to derive new analytical solutions to special classes of PDEs [149]. Advantages of mapping nonlinear PDEs onto their linear counterparts are self-evident: analytical tools available for linear PDEs provide better understanding of the behavior of a solution and numerical solvers for linear systems are both easier to implement and computationally cheaper.

The Koopman operator theory [39] shares the goal of hodograph transformation: a Koopman operator is an infinite-dimensional linear operator that represents the underlying finite-dimensional nonlinear dynamic system by judiciously choosing observable functions. It is also similar in its goal to integral transformations that map certain classes of nonlinear PDEs onto their linear counterparts. For example, the Cole-Hopf transformation and the Kirchhoff transformation map, respectively, Burgers' equation and a class of nonlinear diffusion (heat conduction) equations onto a linear diffusion equation. These integral transformations have been used in the context of the Koopman operator theory and DMD/POD to construct ROMs for Burgers' equation [150] and a nonlinear diffusion equation [2]. A major goal of our study is

to establish clear connections between hodograph transformation and the Koopman theory. This relationship between the two is then used both to identify observables for a Koopman operator via hodograph transformation and to construct ROMs for hyperbolic conservation laws with shocks via DMD.

Besides the choice of the observable functions, another key ingredient of the success of a DMD algorithm is data availability. As proved theoretically by [75] and verified numerically by [2], data have to be sufficiently rich for the learning algorithm to capture all essential features of the underlying dynamics. Therefore, a key condition in our DMD framework for mixed wave problems is that the data be collected until and after all forms of propagation occurs. A resulting ROM remains valuable as a predictor of the continuing propagation. For example, the Buckley-Leverett’s equation (see section 4.4.5) is widely used in oil and gas industry to describe the water injection and oil production processes. A shock profile forms right after the injection begins. Quantities of interest are “breakthrough time” (i.e., the time when the shock front exits the domain) and “water-cut curve” (i.e., the cumulative rarefaction curve after the breakthrough), which can be efficiently predicted by a successful ROM.

## 4.1 ROM Failure for Problems with Shocks

Consider a state variable  $u(x, t) : [a, b] \times [0, T] \rightarrow \mathbb{R}$ , where the constants  $a, b \in \mathbb{R}$  and  $T \in \mathbb{R}^+$ . The dynamics of  $u(x, t)$  is described by a one-dimensional scalar conservation law

$$\frac{\partial u}{\partial t} + \frac{\partial C(x, t, u)}{\partial x} = 0 \quad \text{or} \quad \frac{\partial u}{\partial t} + c(x, t, u) \frac{\partial u}{\partial x} = 0, \quad c(x, t, u) = \frac{\partial C(x, t, u)}{\partial u}. \quad (4.1)$$

This hyperbolic PDE is subject to the initial condition  $u(x, t = 0) = u_0(x)$  and, when appropriate (i.e., when  $|a|, |b| < \infty$ ), boundary conditions at  $a$  and/or  $b$ . This group of PDEs belongs to (1.1) and (3.1) (with  $D = 0$ ).

A numerical solution provided by (3.2) with  $D \equiv 0$  using sufficiently small  $\Delta t$  and  $\Delta x$ , satisfying corresponding CFL condition, are referred to as a reference HFM throughout this chapter.

Standard (Eulerian) approaches to construction of a ROM for (4.1) often fail due to the traveling-wave nature of its solution (see section 3.1 and [30, 111]). In a shock-free scenario, the Lagrangian framework (see section 3.2) can resolve the translational issue in the POD or DMD approaches to ROMs by keeping track of the characteristic lines.

One of the most studied examples of (4.1) is the inviscid Burgers equation

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = 0, \quad u(x, 0) = u_0(x), \quad (4.2)$$

which we define on the space-time domain  $(x, t) \in [0, 2\pi] \times [0, 1]$ . Depending on the boundary and initial conditions, this problem admits both smooth and discontinuous solutions  $u(x, t)$ . For example, a smooth solution is obtained for the periodic boundary conditions,  $u(0, t) = u(2\pi, t)$ , and the initial data  $u_0(x) = 1 + \sin(x)$  (see section 3.3.4). In this setting, standard (Eulerian) ROMs fail due to the inability of SVD to represent a low rank structure of translational problem, while the ROMs based on either Lagrangian POD or Lagrangian DMD perform well in terms of both accuracy and computational efficiency [30].

A solution to (4.2) develops shocks in finite time for, e.g., a Gaussian-type initial data,

$$u_0(x) = 0.8 + 0.5 \exp \left[ -\frac{(x - 0.3)^2}{0.001} \right]. \quad (4.3)$$

In the pure Lagrangian approach (3.4), the discretization has to account for shock formation. Once the characteristic lines cross each other, the Lagrangian mesh becomes sensitive to the choice of discretization of  $u(x(t), t)$ . For instance, a discretization of (3.4) with  $c(\cdot, u) = u$ ,

$$\begin{cases} u_j^{n+1} = u_j^n, \\ x_j^{n+1} = x_j^n + \Delta t u_j^{n+1}, \end{cases} \quad (4.4)$$

would lead to the so-called “overshoot” that admits multi-value solutions (Figure 4.1a), which contradicts the entropy condition. This is a typical problem with the Lagrangian framework. It should come as no surprise that an attempt to build a ROM with the Lagrangian DMD based on the faulty discretization (4.4) likewise results

in failure (Figure 4.1b). The Lagrangian DMD faithfully reproduces the unphysical solution obtained with the faulty discretization scheme (4.4). In other words, the resulting unphysical ROM is not caused by the DMD algorithm itself; the data from the full Lagrangian model (4.4) provide inaccurate and incomplete (without shock) information from the very beginning.

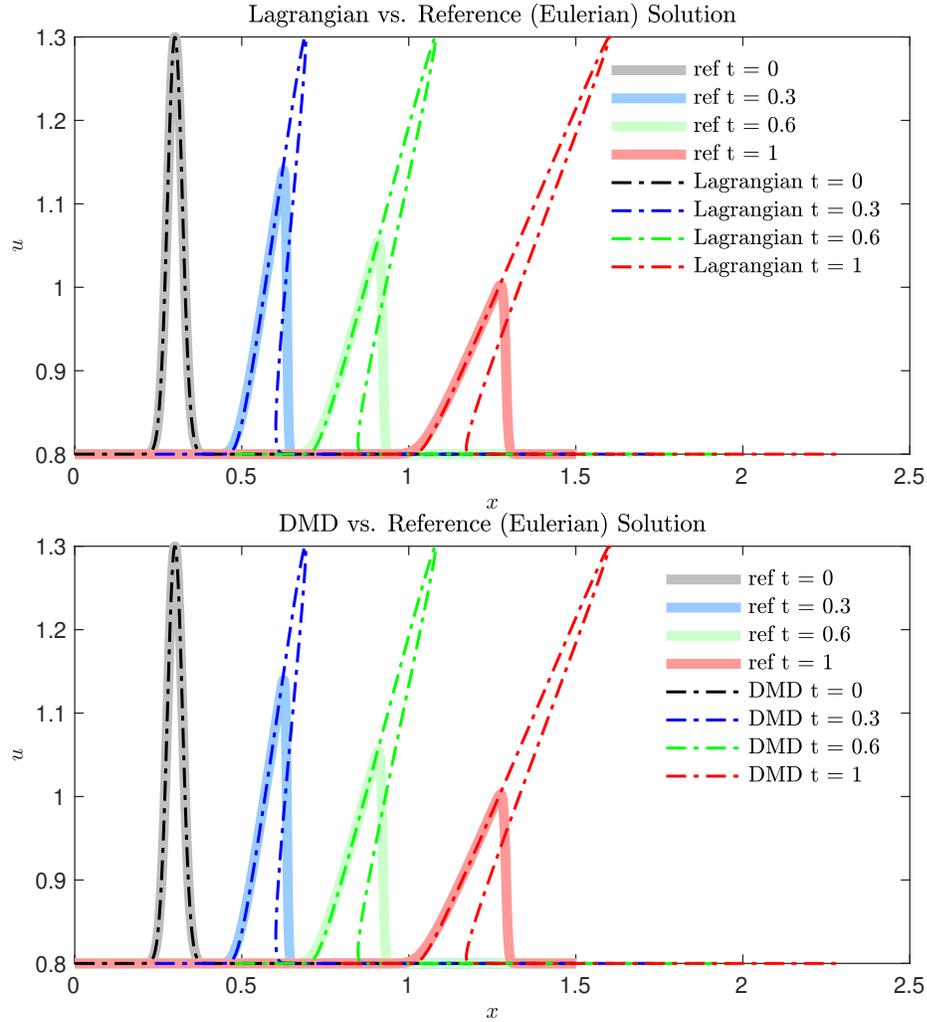


Figure 4.1: Solutions of the inviscid Burgers equation with a shock. **(a)** Top: the full solution obtained with the Lagrangian numerical scheme (4.4) leading to the overshoot. **(b)** Bottom: the Lagrangian DMD solution trained on a few snapshots of the faulty full solution. The reference solution is obtained with (3.2).

We consider a numerical scheme that is known for its ability to handle shocks:

the backward semi-Lagrangian method (BSLM),

$$\begin{cases} u_j^{n+1} = u_j^n, \\ x^* = x_j^n + \frac{\Delta t}{2} u_j^n, \\ x_j^{n+1} = x_j^n + \frac{\Delta t}{2} (u(x^*, t^n) + u(x^*, t^{n+1})), \end{cases} \quad (4.5)$$

and employ the (explicit) mid-point rule to avoid implicit iterations. Since  $u(\cdot, t^{n+1})$  is unchanged from  $u(\cdot, t^n)$ , the evaluations at  $x^*$  can be calculated via interpolation, e.g., via linear interpolation used below. Figure 4.2a reveals that this numerical scheme is indeed capable to accurately approximate the solution of the inviscid Burgers equation with shocks. However, the Lagrangian DMD algorithm using snapshots from the full solution (4.5) suffers from instability once a shock is about to form (Figure 4.2b). The grid becomes severely distorted once the characteristic lines intersect each other at the interface where sharp gradients of  $u(x, t)$  occur. At the intersect, one arrival location of  $x$  corresponds to two different departure values of  $u$ . However, the DMD modes projection from the HFM to the ROM does not keep the topological information about this multivalued mapping in the ROM process, resulting in the Lagrangian grid distortion.

**Remark 4.1.** *The Lagrangian POD approach suffers from similar problems [111]. Moreover, the POD projection on the accurate Lagrangian scheme (4.5) would still require interpolation in the high-dimensional space. One might need techniques such as DEIM [37] to keep the resulting ROM's efficiency. Nevertheless, extensions of POD are beyond the scope of our study; we focus on DMD-based ROMs due to their iteration-free nature.*

## 4.2 Hodograph Transformation

We start with a mathematical definition of hodograph transformation reproduced from [149].

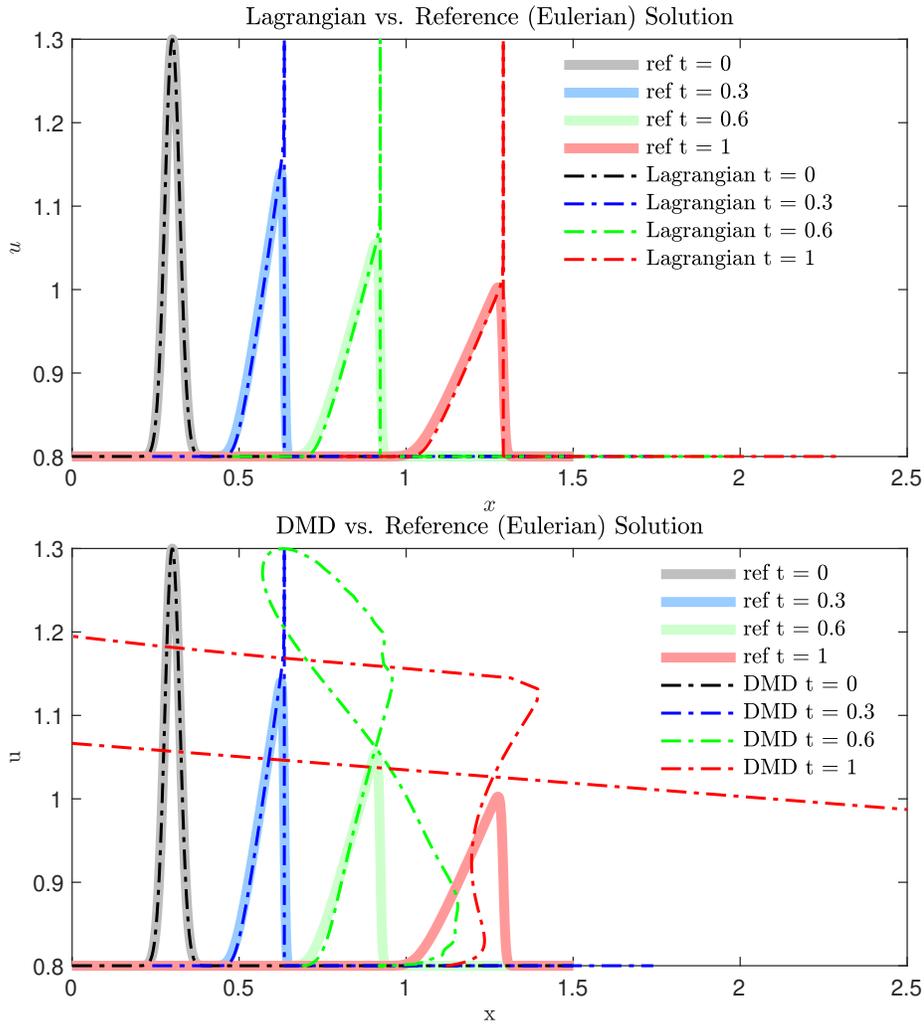


Figure 4.2: Solutions of the inviscid Burgers equation with a shock. **(a)** Top: the full solution obtained with the appropriate Lagrangian numerical scheme (4.5). **(b)** Bottom: the Lagrangian DMD solution trained on a few snapshots of the accurate full solution. The reference solution is obtained with (3.2).

**Definition 4.1.** *A pure hodograph transform is a transformation of the form*

$$\tau = t, \quad \xi = u(x, t). \quad (4.6)$$

For the inviscid Burgers equation (4.2), we first consider a scenario where only one shock is developed from the initial data  $u_0(x)$  in finite time. This necessitates the following assumption.

**Assumption 4.1.** *The function  $u_0(x)$  satisfies four conditions:*

- $u_0(x)$  is smooth;
- $u_0(x)$  decreases monotonically, i.e.,  $u'_0(x) < 0$  for all  $x$ ; and  $\lim_{x \rightarrow +\infty} u_0(x) = u_R$ ,  $\lim_{x \rightarrow -\infty} u_0(x) = u_L$  with constants  $u_R < u_L$ ;
- $u_0(x)$  has a unique inflection point  $(x^*, u^*)$  with  $u^* = u_0(x^*)$ , meaning  $u''_0(x^*) = 0$ ;
- $u'''_0(x^*) > 0$ .

This assumption ensures existence of an inverse function  $x(t, u) : [0, T] \times [u_R, u_L] \rightarrow [a, b]$  of the monotonic function  $u(t, x) : [0, T] \times [a, b] \rightarrow [u_R, u_L]$ . The last two assumptions ensure single-shock formation for illustration purposes. It follows from Definition 4.1 that the inverse function  $x(t, u) = x(\tau, \xi)$  is a pure hodograph transform based on (4.6).

### 4.2.1 Solution Before Shock Formation

With  $u$  acting as the independent variable and  $x$  as the dependent variable, hodograph transformation  $x = x(t, u)$  maps the equation for characteristics (3.4) of the inviscid Burgers equation (4.2), before the shock formation time  $t^*$  (defined later), onto

$$\frac{dx(t, u)}{dt} = u, \quad x(0, u) = x_0(u); \quad \text{for } (t, u) \in [0, t^*) \times [u_R, u_L]. \quad (4.7)$$

Assumption 4.1 translates into conditions on the function  $x_0(u)$ :

- $x'_0(u) < 0$ ;
- $x_0(u)$  has a unique inflection point at  $(u^*, x^*)$ ;
- $x'''_0(u^*) < 0$ .

Differentiation of (4.7) with respect to  $u$  gives

$$\frac{\partial^2 x}{\partial t \partial u}(t, u) = 1, \quad (4.8)$$

from which

$$\frac{\partial x}{\partial u}(t, u) = x'_0(u) + t, \quad \text{for } (t, u) \in [0, t^*] \times [u_R, u_L]. \quad (4.9)$$

Let  $t^* = \min_u[-x'_0(u)] = -x'_0(u^*)$  denote the time of shock formation; the shock location is  $x^* = x(u^*, t^*)$ . Since  $x'_0(u) < 0$ , the derivative  $\partial_u x(t, u) < 0$  as long as  $t < t^*$ .

### 4.2.2 Solution After Shock Formation

At times  $t$  larger than  $t^* = -x'_0(u^*)$ , i.e., once the shock forms, (4.7) is no longer valid. In the  $(x, u)$  plane, one would use the entropy (Rankine-Hugoniot) condition to construct a weak formulation of Burgers' equation. Its analog in the  $(u, x)$  plane gives an equation for the shock speed  $s$ :

$$s = \frac{1}{2} \frac{u_1^2 - u_2^2}{u_1 - u_2} = \frac{u_1 + u_2}{2}, \quad (4.10)$$

where  $u_1(t)$  and  $u_2(t)$  are defined as the limits of  $u(t)$  from the top and bottom of the shock, respectively. They are computed as solutions of a system of ordinary differential equations (see [151] for detailed derivation)

$$\begin{cases} \frac{du_1}{dt} = \frac{1}{2} \frac{u_1 - u_2}{\psi(u_1) - t}, \\ \frac{du_2}{dt} = -\frac{1}{2} \frac{u_1 - u_2}{\psi(u_2) - t}, \end{cases} \quad (4.11)$$

where  $\psi(u) \equiv -x'_0(u)$ . These ODEs are subject to initial conditions  $u_1(t^*) = u^*$  and  $u_2(t^*) = u^*$ . Since  $s = dx^*(t)/dt$ , an equation for the shock trajectory  $x^*(t)$  is

$$\frac{dx^*(t)}{dt} = \frac{u_1 + u_2}{2}. \quad (4.12)$$

### 4.2.3 Summary of Hodograph Solution

Under Assumption 4.1, the hodograph-transformed Burgers' equation (4.2) takes the form of the following ODEs for  $x(t, u)$ :

$$\begin{cases} t < t^* : & \text{Equation (4.7)} \\ t > t^* : & \begin{cases} \text{Equation (4.7)} & \text{for } u \in (u_R, u_2) \cup (u_1, u_L) \\ \text{Equation (4.12)} & \text{for } u \in (u_2, u_1), \end{cases} \end{cases} \quad (4.13)$$

where  $t^* = -x'_0(u^*)$ , and  $u_1$  and  $u_2$  are solutions of (4.11).

**Remark 4.2.** *One can show that  $u_1(t)$  is a monotonically increasing function and  $u_2(t)$  is a monotonically decreasing function, such that*

$$u_1 \geq u^*, \quad u_2 \leq u^*, \quad x'_0(u_1) + t \leq 0, \quad x'_0(u_2) + t \leq 0. \quad (4.14)$$

*In many cases of interests, and in some our numerical experiments, either  $u_1$  and  $u_2$  are known or  $|u_2 - u_R|, |u_1 - u_L| \ll \Delta t$  (so that  $u_2 \approx u_R$  and  $u_1 \approx u_L$ ). This allows one to focus on shock propagation, i.e., on (4.12), without having to solve (4.11).*

**Remark 4.3.** *Functions  $u_0(x)$  that do not satisfy Assumption 4.1, such as (4.3), require a decomposition of the initial data into monotonic parts. Each monotonic piece of  $u_0(x)$  would have a unique inverse function  $x_0(u)$ . The entropy condition implies that the increasing  $x_0(u)$ , i.e.  $x'_0(u) > 0$ , results in a rarefaction solution, which satisfies (4.7). The union of the rarefaction pieces and shock pieces would give the full solution.*

**Remark 4.4.** *The inviscid Burgers equation is an example of hyperbolic conservation*

laws with monotonically increasing flux functions  $C(\cdot, u)$ . Generalization to hyperbolic conservation laws with a convex flux is presented in Appendix B.

### 4.3 Physics-Aware DMD for Conservation Laws with Shocks

Previous theoretic investigations, e.g., by [2], demonstrated that the key to the success of the DMD in capturing nonlinear dynamics is to identify the underlying Koopman operator. Several numerical studies [43, 65, 150] confirmed this finding. The Koopman operator theory ensures that a DMD algorithm utilizes all relevant physical information to learn the dynamics. We refer to this approach as physics-aware DMD to distinguish it from the conventional DMD that learns only from (simulations-generated) data.

The Koopman operator theory is reviewed in section 1.1.2. There is no principled way to choose the observables without expert knowledge of a system under consideration. Selection of observables remains a grand challenge and an active research area, e.g., machine learning and deep learning techniques were recently employed to identify the underlying Koopman operator [132]. In the context of conservation laws with shocks, the equivalency between hodograph transformation and the Koopman operator, established in this study, facilitates a “smart” choice of the observables. It is implemented by the construction of observable  $\mathbf{g}$  via (1.27) where  $g_j^k = x(k\Delta t, \mathbf{u}^k)$  with  $j = 1, \dots, p - 1$  is the inverse function of  $u(t, x)$ , evaluated at the prescribed mesh  $\bar{\mathbf{u}}$ . The last observable  $g_p^k$  is a problem-dependent recording of shock information. Then, we follow Algorithm 2 and (1.30) to (1.31).

**Remark 4.5.** *Numerically,  $\mathbf{g}$  can be obtained by interpolation from a uniform mesh in the  $(x, u)$  plane to a uniform mesh in the  $(u, x)$  plane, and so can  $\mathbf{g}^{-1}$ . The monotonicity Assumption 4.1 ensures that the observable functions are one-to-one maps.*

**Remark 4.6.** *The challenge of incorporating the shock information into the Lagrangian DMD algorithm of [30] is the dependence of shock speed on the dependent*

variable  $u$ . Hodograph transformation facilitates the incorporation of this implicitly nonlinear information by turning  $u$  an independent variable and by rendering the shock speed given by the Rankine-Hugoniot condition linear.

**Remark 4.7.** *For problems with shocks, one needs to collect snapshots until and after a shock forms. Otherwise, the Koopman operator cannot learn the shock dynamics.*

**Remark 4.8.** *For mixed wave problems, one needs to collect snapshots until and after all forms of propagation occurs. This requires pre-observation, pre-processing and understanding of the data. General initial data  $u_0(x)$  has to be separated into monotonic sub-regions. Physical quantities, such as shock speed and intersection point of shock and rarefaction propagation, must be understood from given data features. They give an explicit form of the shock observable function  $g_p$ ; although problem-dependent, all the shock information is linear with respect to  $u$ .*

**Remark 4.9** (Algorithm’s accuracy). *The error of our physics-aware DMD algorithm stems from two sources. The first is the error due to order reduction in the observable space; it represents the accuracy with which (1.30) predicts the true observable  $\mathbf{y}^{n+1}$ . According to the error estimator of [2], this prediction accuracy depends on the number of snapshots  $M$ ; the rank truncation criteria, as in (1.7); and the linear operator  $\mathbf{K}$  induced by  $\mathbf{g}$ . The second is the error due to forward and backward mapping  $\mathbf{g}$ ; in the discrete setting, it presents an error in projection between the  $(x, u)$  space and the  $(u, x)$  space. The model-order-reduction error and projection error are studied numerically in section 4.4.6.*

## 4.4 Numerical Tests of Hodograph DMD

We apply the physics-aware DMD to construct ROMs of scalar conservation laws in different scenarios, including a shock, rarefaction and a mixture of both. These hyperbolic conservation laws take the forms of the inviscid Burgers equation (sections 4.4.1–4.4.4), the Buckley-Leverett’s equation (section 4.4.5), and the Euler equations (section 4.4.6). The conservative first-order upwind scheme (3.2) is employed as

a reference solution (using  $N_x = N_u$ ), except when an analytical solution is available. The rank truncation criterion (1.7) with  $\varepsilon = 10^{-4}$  is used in all cases.

The observable data  $\mathbf{y}^n$  in (1.27) now relies on the mesh  $\bar{\mathbf{u}} = [\bar{u}_1, \dots, \bar{u}_{N_u}]^\top$  with mesh size  $\Delta\bar{u} = \bar{u}_{j+1} - \bar{u}_j$  and the minimum and maximum values  $\bar{u}_1$  and  $\bar{u}_{N_u}$ , respectively. In general, there are two strategies of collecting data on mesh  $\bar{\mathbf{u}}$ :

- (a) Solving (4.13) for  $x(t, u)$  in the  $(u, x)$  plane on the mesh  $\bar{\mathbf{u}}$  at discrete time  $t_n, n = 1, \dots, N_t$ ;
- (b) Solving (3.2) for  $u(t, x)$  in the  $(x, u)$  plane on the mesh for  $x$  at discrete time  $t_n, n = 1, \dots, N_t$ , then interpolating for  $x(t, u)$  in the  $(u, x)$  plane on the mesh  $\bar{\mathbf{u}}$ .

In the case of Burgers' equation and the Buckley-Leverett's equation, both strategies are applicable. We choose strategy (a) for the examples in section 4.4.1–4.4.5 to avoid the projection error due to the mapping between the  $(x, u)$  and  $(u, x)$  planes. In the case of the Euler equations (section 4.4.6), only strategy (b) is feasible. Regardless of the strategy used, the physics-aware DMD algorithm 2, applied to the  $M$  observables  $\mathbf{y}^0, \dots, \mathbf{y}^{M-1}$  in (1.27), allows one to predict  $\mathbf{y}_{\text{DMD}}^n$  for  $n \geq M$  and to obtain  $\mathbf{u}_{\text{DMD}}^n$  by concatenating the interpolation of  $\mathbf{y}_{\text{DMD}}^n$  on the mesh for  $x$ .

#### 4.4.1 Riemann Problem for Burgers' Equation with Shock

Consider the inviscid Burgers equation (4.2) defined for  $(x, t) \in [-0.5, 1.5] \times [0, 1]$  and with initial data

$$u_0(x) = \begin{cases} 2 & \text{for } -0.5 \leq x < 0 \\ 0 & \text{for } 0 \leq x \leq 1.5. \end{cases} \quad (4.15)$$

This problem admits an analytical solution

$$u(x, t) = \begin{cases} 2 & \text{for } -0.5 \leq x < st \\ 0 & \text{for } st < x \leq 1.5, \end{cases} \quad (4.16)$$

where the shock speed  $s = 1$  is determined from the Rankine-Hugoniot condition.

The discontinuous initial data  $u_0(x)$  in (4.15) do not satisfy Assumption 4.1. Thus, we approximate the step function  $u_0(x)$  with a smooth function, e.g., the hyperbolic tangent

$$u_0 \approx 1 - \tanh\left(\frac{x}{\delta}\right), \quad \delta \ll 1, \quad (4.17)$$

which satisfies Assumption 4.1. In the  $(u, x)$  plane,

$$x_0 \approx \frac{\delta}{2} \log\left(\frac{2 - u_0}{u_0}\right), \quad \delta \ll 1. \quad (4.18)$$

This approximation is valid in the neighborhood of the shock interface; away from it, (4.15) is used. It follows from (4.18) that

- $x'_0(u) = \frac{\delta}{(u-2)u} < 0$  for  $u \in (0, 2)$ ;
- $x_0(u)$  has a unique inflection point at  $(u^* = 1, x^* = 0)$ ;
- $x'''_0(u) = \frac{4(3u^2 - 6u + 4)}{(u-2)^3 u^3}$  and thus  $x'''_0(u^*) = -4 < 0$ .

Snapshots of  $x(t, u)$  on a uniform mesh  $\bar{\mathbf{u}} = [\bar{u}_1, \dots, \bar{u}_{N_u}]^\top$ , which consists of  $N_u = 2000$  equidistant points, are collected at  $M = 250$  times until  $T = 0.25$ . The ROM is used to predict the solution  $u(x, t)$  for larger times,  $t > T$ . The shock-related information is contained in  $u_1(t)$  and  $u_2(t)$ , first defined in (4.11). Since both are constant in this example, no extra observable  $g_p$  is needed to record the shock information. Each column  $\mathbf{y}^n$  in the data matrix is of length  $p = 2000$  and elements  $\mathbf{y}_j^n = x(t^n, \bar{u}_j)$  for  $j = 1, \dots, N_u$ . If one were to add the constant shock information  $u_1$  and  $u_2$  to the observables, then  $\mathbf{y}^n$  would have length  $p = 2002$  such that elements  $\mathbf{y}_j^n = x(t^n, \bar{u}_j)$  for  $j = 1, \dots, N_u$  are supplemented with two extra elements  $\mathbf{y}_{2001}^n = u_1 = 2$  and  $\mathbf{y}_{2002}^n = u_2 = 0$ . Doing so would lead to the same prediction: the DMD algorithm is able to learn the pattern of the last two observables being constant in time.

Figure 4.3 demonstrates the the physics-aware DMD algorithm with hodograph transformation captures the behavior of the shock propagation. Only  $r = 2$  modes are needed to construct the ROM, which remains accurate for relatively long time in the extrapolation mode. Hodograph transformation converts the nonlinear conservation

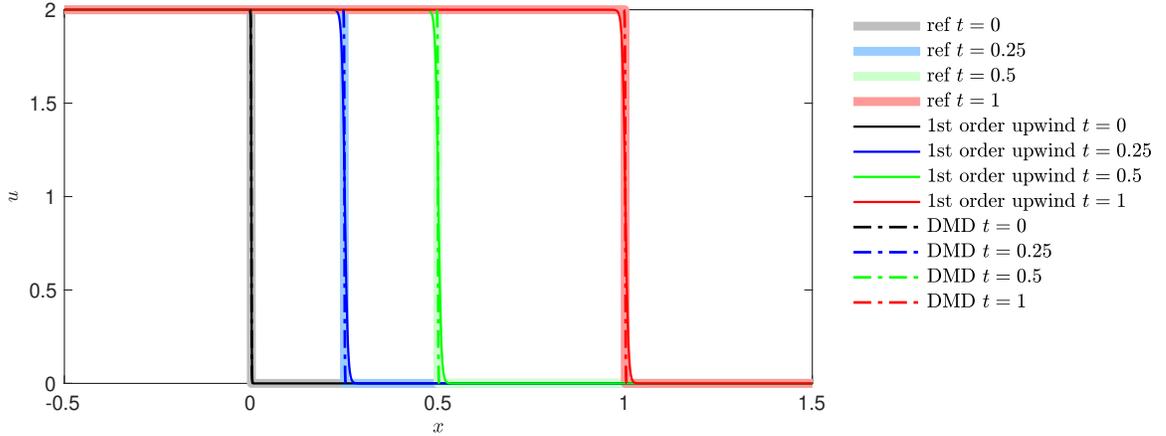


Figure 4.3: Physics-aware DMD solution of the inviscid Burgers equation with a shock. The reference solution is given by analytic solution (4.16). 1st order upwind scheme by (3.2) is also plotted here in solid line.

law (4.2) with discontinuous initial data (4.15) into a linear shift with constant speed, which is readily learned from data. Due to the monotonicity constraint, the solution using hodograph transformation (and the concomitant DMD prediction) is only valid in the neighborhood of the shock interface, which is often of interest in itself. Away from the discontinuity, the Lagrangian DMD [30] is accurate and should be used instead.

#### 4.4.2 Riemann Problem for Burgers' Equation with Rarefaction Wave

Consider the inviscid Burgers equation (4.2) defined for  $(x, t) \in [-1, 1] \times [0, 1]$  and with initial data

$$u_0(x) = \begin{cases} -1 & \text{for } -1 \leq x < 0 \\ 1 & \text{for } 0 \leq x \leq 1. \end{cases} \quad (4.19)$$

This problem admits an analytical solution in the form of a rarefaction wave,

$$u(x, t) = \begin{cases} -1 & \text{for } -1 \leq x < -t \\ x/t & \text{for } -t < x < t \\ 1 & \text{for } t < x \leq 1. \end{cases} \quad (4.20)$$

A hyperbolic-tangent approximation analogous to (4.17) is used to deal with the discontinuity in the initial data  $u_0(x)$ . And the same structure of data matrix is used in the physics-aware DMD algorithm 2 with  $N_u = 2000$  and  $M = 250$  until  $t = 0.25$ . Since there is no shock in this scenario, no extra observable is needed to record shock-related information.

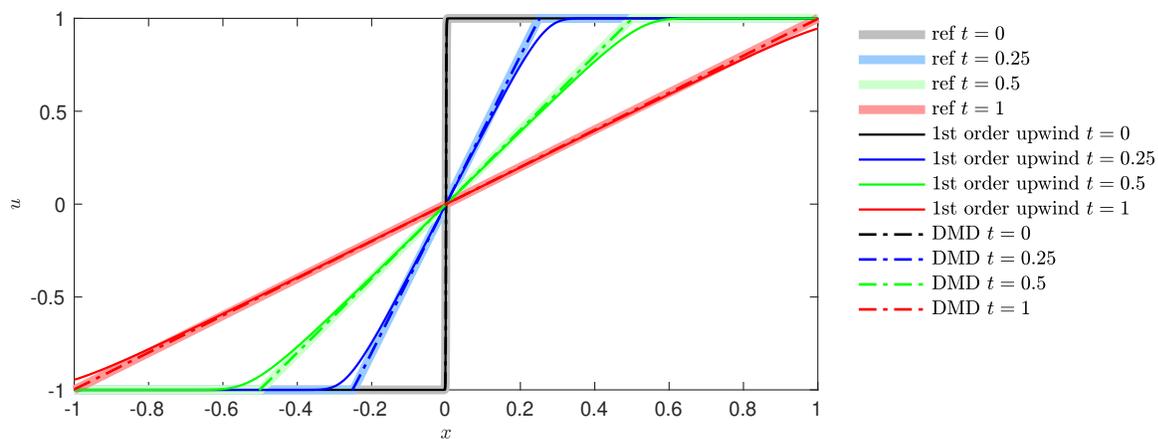


Figure 4.4: Physics-aware DMD solution of the inviscid Burgers equation with a rarefaction wave. The reference solution is given by analytic solution (4.20). 1st order upwind scheme by (3.2) is also plotted here in solid line.

Figure 4.4 shows the same satisfactory ROM results for this problem with a rarefaction wave. Only  $r = 2$  modes are needed to obtain accurate predictions because the hodograph transform,  $x = x(t, u)$ , satisfies a linear ODE (4.7), with  $u$  acting as an independent variable. The absence of a shock suggests that this rarefaction scenario can also be handled with the Lagrangian DMD algorithm, with similar results.

Given an accurate approximation of the initial discontinuity, i.e., selecting  $\delta$  to be sufficiently small, the ROM trained on the data generated from the solution of (4.7)

is even more accurate than the HFM solution. The upwind scheme (3.2) has first-order accuracy,  $O(\Delta t)$ , while the physics-aware DMD algorithm can have spectral accuracy. Figures 4.3 and 4.4 show that the physics-aware DMD solution has a much sharper interface than that estimated with the first-order upwind scheme (3.2).

#### 4.4.3 Smooth Solution of Riemann Problem for Burgers' Equation with Non-monotonic Initial Data

Consider Burgers' equation (4.2) defined for  $(x, t) \in [-\pi/2, 3\pi/2] \times [0, 1]$  and with initial data

$$u_0(x) = 1 + \sin(x). \quad (4.21)$$

Since these initial data violate Assumption 4.1, we decompose the interval  $[0, 2\pi]$  into two parts: in the left part,  $x \in [-\pi/2, \pi/2]$ ,  $u_0(x)$  monotonically increases; in the right part,  $x \in [\pi/2, 3\pi/2]$ , it monotonically decreases. Each part has a unique inverse function of  $x_0(t, u_0)$ ; we denote  $x_l(t, u)$  as the evolution of the left part and  $x_r(t, u)$  as the evolution of the right part. Since the shock formation time is  $t^* = 1$ , the equation of characteristics for this Riemann problem is equivalent to (4.7) on any finite-time interval  $[0, T] \subset [0, 1]$ . Although this is a shock-free scenario, the two parts have different wave propagation behaviors. The numerical scheme (3.2) with  $N_x = 2000$  spatial discretization points and  $N_t = 1000$  time steps provides the reference solution. The data used to inform our DMD method consist of  $M = 250$  snapshots with observables  $\mathbf{y}_j^k = x_l(t_k, \bar{u}_j)$  for  $j = 1, \dots, N_u$  and  $\mathbf{y}_j^k = x_r(t_k, \bar{u}_{j-N_u})$  for  $j = N_u + 1, \dots, 2N_u$ . No extra observable is needed to record shock-related information since there is no shock formation in the considered time interval.

Figure 4.5 demonstrates the ability of the ROM based on our physics-aware DMD algorithm to capture these nonlinear dynamics. Only  $r = 2$  modes are needed to obtain accurate predictions due to the linearity after hodograph transform. The ROM was trained on the early ( $t \leq 0.25$ ) data, which exhibit smooth gradients. Yet, it accurately captures sharp gradients at later times, e.g.,  $t = 1$ . That is because, in the  $(u, x)$  domain of the hodograph transform, higher gradients of  $u(\cdot, x)$  translate into flatter horizontal plots of  $x(\cdot, u)$ .

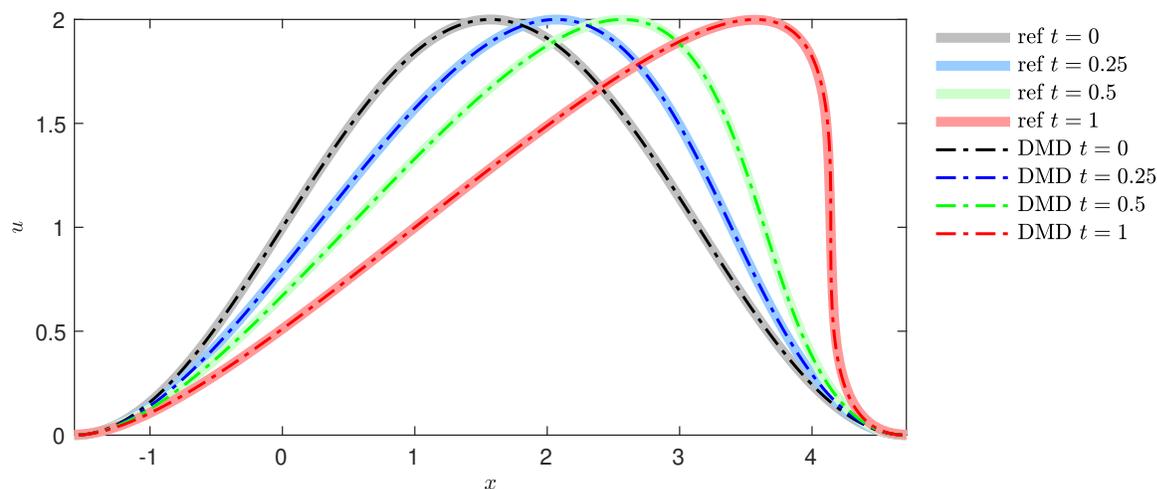


Figure 4.5: Physics-aware DMD solution of the inviscid Burgers equation subject to nonmonotonic initial condition. The reference solution is computed with (3.2).

#### 4.4.4 Riemann Problem for Burgers' Equation with Rarefaction and Shock

Consider Burgers' equation (4.2) defined for  $(x, t) \in [0, 2] \times [0, 1]$  and with the Gaussian-type initial data in (4.3). This is the setting we used to illustrate the failure of the Lagrangian DMD in section 4.1 (Figure 4.2). The numerical scheme (3.2) with  $N_x = 2000$  spatial discretization points and  $N_t = 10^5$  time steps provides the reference solution. The finer time discretization is needed to satisfy the CFL constraints. The data used to inform our physics-aware DMD method consist of  $M = 3000$  snapshots of solving (4.13). These data are sufficiently rich to identify the rarefaction and shock behavior of the solution.

A decomposition of the initial data  $u_0(x)$  in (4.3) is needed to enforce monotonicity. The increasing branch of  $u_0(x)$  is responsible for the rarefaction and its decreasing branch gives rise to the shock. It follows from (4.13) that

$$x_l(u, t^0) = 0.3 - \sqrt{-0.001 \ln(2u - 1.6)} \quad \text{and} \quad x_r(u, t^0) = 0.3 + \sqrt{-0.001 \ln(2u - 1.6)}.$$

We can verify that  $x_r(u; t^0)$  has a unique inflection point  $(u^*, x^*)$  with  $u^* = 0.8 +$

$1/(2\sqrt{e})$  and  $x^* = \sqrt{0.002e}$  and, for  $u \in (0.8, 1.3)$ , both  $x'_r(u; t^0) < 0$  and  $x'''_r(u; t^0) < 0$ . Consequently, the data are generated from (4.13) as follows.

1. A uniform mesh  $\bar{\mathbf{u}} = [\bar{u}_1, \dots, \bar{u}_{N_u}]^\top$  is constructed with mesh size  $\Delta\bar{u} = \bar{u}_{j+1} - \bar{u}_j$ , and  $\bar{u}_1 = 0.8$  and  $\bar{u}_{N_u} = 1.3$ .

2. For  $x_l(u, t)$ , no shock develops. The full discretization (4.7) gives

$$\begin{cases} \frac{x_l(\bar{u}_j, t_{n+1}) - x_l(\bar{u}_j, t_n)}{\Delta t} = \bar{u}_j, \\ x_l(\bar{u}_j, t_0) = 0.3 - \sqrt{-0.001 \ln(2\bar{u}_j - 1.6)}. \end{cases} \quad (4.22)$$

3. For  $x_r(u, t)$ , the shock formation time  $t^*$  is calculated as

$$t^* = \min_u \{-x'_r(u, t_0)\} \approx 0.074. \quad (4.23)$$

(a) If  $t_{n+1} \leq t_*$ , then

$$\begin{cases} \frac{x_r(\bar{u}_j, t_{n+1}) - x_r(\bar{u}_j, t_n)}{\Delta t} = \bar{u}_j, \\ x_r(\bar{u}_j, t_0) = 0.3 + \sqrt{-0.001 \ln(2\bar{u}_j - 1.6)}. \end{cases} \quad (4.24)$$

(b) If  $t_{n+1} > t_*$ ,  $u_2(t_n) = u_R = 0.8$  is known and  $u_1(t_n)$  is approximated by the intersection of  $x_l(u, t_n)$  and  $x_r(u, t_n)$  due to the continuity of the solution, i.e.,

$$\begin{cases} u_1(t_n) = \underset{\bar{u}_j}{\operatorname{argmin}} |x_r(\bar{u}_j, t_n) - x_l(\bar{u}_j, t_n)|, & u_2(t_n) = u_R = 0.8, \\ \frac{x_r(\bar{u}_j, t_{n+1}) - x_r(\bar{u}_j, t_n)}{\Delta t} = \frac{u_1(t_n) + u_2(t_n)}{2}. \end{cases} \quad (4.25)$$

4. The observable vector  $\mathbf{y}^n$  of length  $p = 2N_u + 2$  comprises  $\mathbf{y}_j^k = x_l(\bar{u}_j, t_k)$  for  $j = 1, \dots, N_u$ ;  $\mathbf{y}_j^k = x_r(\bar{u}_{j-N_u}, t_k)$  for  $j = N_u + 1, \dots, 2N_u$ ; and  $\mathbf{y}_{2N_u+1}^k = u_1(t_k)$  and  $\mathbf{y}_{2N_u+2}^k = u_2(t_k)$ .

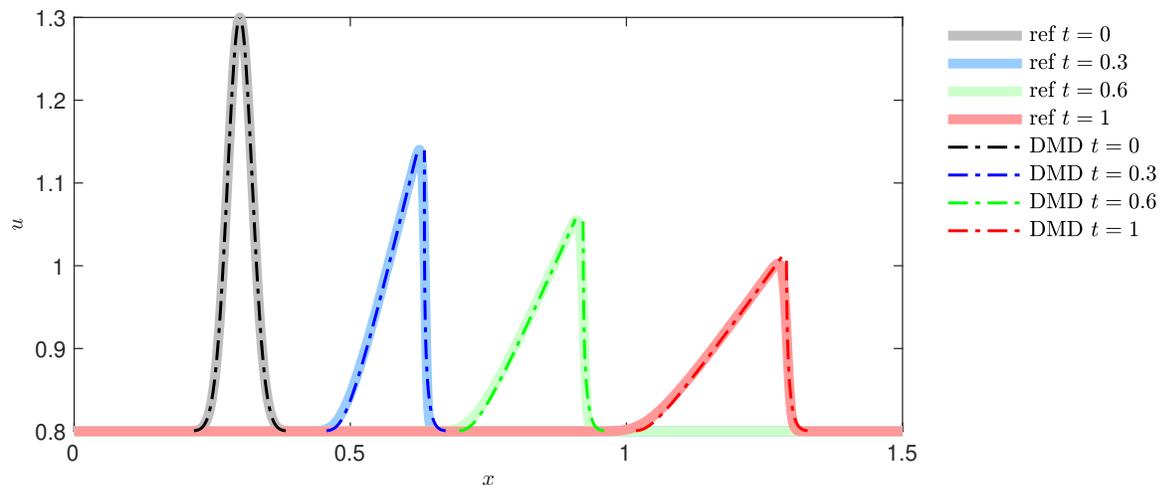


Figure 4.6: Physics-aware DMD solution of the inviscid Burgers equation with a rarefaction wave and shock. The reference solution is computed with (3.2).

Figure 4.6 shows that the physics-aware DMD based on hodograph transformation provides an accurate ROM for this Riemann problem, which could not be treated with the original Lagrangian DMD. The physical shock information, which is needed for the last two observable functions, includes the shock speed and the intersection point of the rarefaction wave and the shock trajectory. In this setting, the shock speed varies with time but is still linear with respect to  $u$ . The physics-aware DMD algorithm can learn this linear relationship from the data with no difficulties. Only  $r = 4$  modes are needed and all of the advantages of linearity are achieved with the hodograph transform.

#### 4.4.5 Riemann Problem for Buckley-Leverett's Equation

Consider the hyperbolic conservation law (4.1) with a nonmonotonic flux function,

$$C = \frac{u^2}{u^2 + a(1-u)^2}, \quad a = 0.5, \quad (4.26)$$

that is defined for  $(x, t) \in [0, 2] \times [0, 0.5]$  and is equipped with initial data

$$u_0(x) = \begin{cases} 1 & \text{for } 0 \leq x < 1 \\ 0 & \text{for } 1 \leq x \leq 2. \end{cases} \quad (4.27)$$

The Buckley-Leverett equation, (4.1) and (4.26), with initial condition (4.27) is widely used to describe the injection process of immiscible two-phase flow in porous media.

Similar to (4.17), the initial discontinuity is approximated with the hyperbolic tangent function. The hodograph treatment of this more general problem is provided in Appendix B. The numerical scheme (3.2) with  $N_x = 2000$  spatial discretization points and  $N_t = 1000$  time steps provides the reference solution up to  $t = 0.5$ . The data used to inform our physics-aware DMD method consist of  $M = 250$  snapshots of solving (B.10) with  $N_u = 2000$  discretization points until  $t = 0.125$ . This set of snapshots is sufficiently rich to reveal a self-similar structure of the solution.

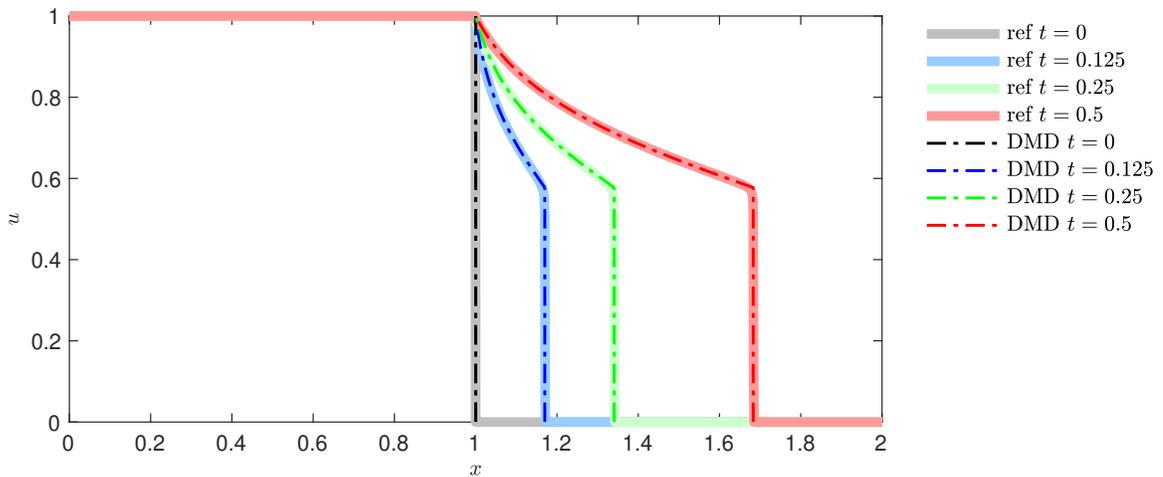


Figure 4.7: Physics-aware DMD solution of the Buckley-Leverett equation, which has a nonmonotonic flux function. The reference solution is computed with (3.2).

Although the initial data  $u_0$  are monotonic, their decomposition is needed according to the convex hull construction of the flux function (Appendix B). The reformulation involves two branches of different linear equations with two sets of the disjoint

initial data. Similar as the previous section, the last two observation functions comprises the shock speed as well as the intersection point of the rarefaction wave and the shock trajectory. This intersection point defines the magnitude of the shock and informs the convex hull construction of the flux function.

Figure 4.7 demonstrates that the physics-aware DMD with  $r = 4$  modes accurately captures the future states in relatively long time. Hodograph transformation allows one to determine the underlying linear Koopman operator in the nonlinear conservation laws. The iteration-free feature of DMD enhances its effectiveness and efficiency.

**Remark 4.10.** *In laboratory experiments with multiphase flows in porous media, one often measures the flow rates of two fluids and approximates the displacement profiles. Such observational data can be used to construct an effective ROM after interpolating them from the  $(x, u)$  plane to the  $(u, x)$  plane. Such a construction would not require any knowledge of the empirical flux function, such as (4.26), and its parameters, such as  $a$ .*

#### 4.4.6 Riemann Problem for Euler Equations

Consider a one-dimensional Sod shock tube problem,

$$\begin{aligned} \frac{\partial}{\partial t} \begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix} + \frac{\partial}{\partial x} \begin{pmatrix} \rho u \\ \rho u^2 + p \\ u(E + p) \end{pmatrix} &= 0, \quad (x, t) \in [-0.5, 0.5] \times [0, 0.25], \\ E &= \frac{p}{\gamma - 1} + \frac{1}{2} \rho u^2, \quad \gamma = 1.4, \end{aligned} \quad (4.28)$$

with initial conditions

$$\rho(x, 0) = \begin{cases} 1 & x < 0.5 \\ 0.125 & x \geq 0.5 \end{cases}, \quad p(x, 0) = \begin{cases} 1 & x < 0.5 \\ 0.1 & x \geq 0.5 \end{cases}, \quad u(x, 0) = 0. \quad (4.29)$$

This problem admits an analytic solution obtained, e.g., via the method of characteristics. Figure 4.8 shows that the solution is a mixture of a rarefaction wave and

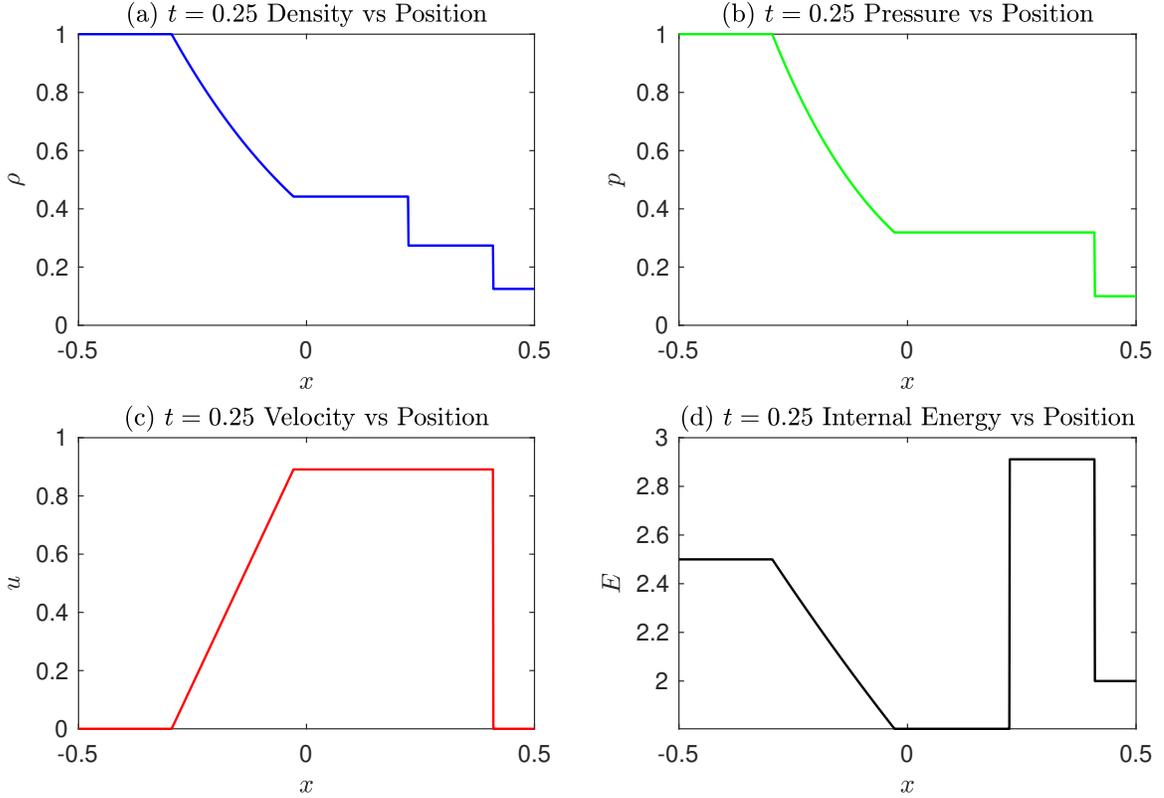


Figure 4.8: The analytical solution to the Sod shock tube problem (4.28): (a) density, (b) pressure, (c) velocity, (d) energy; all evaluated at time  $t = 0.25$ .

a shock, which undermines the performance of conventional ROM approaches. For example, Figure 4.9(a) shows the failure of the standard (Eulerian) DMD to construct a ROM in the  $(x, \rho)$  plane, with similarly unsatisfactory performance for  $p$  and  $u$ . In these simulations,  $x$  is discretized using  $N_x = 1000$  equidistant points, and  $M = 250$  snapshots data are collected from  $t = 0$  to  $t = 0.0625$ . The prediction at later time,  $t = 0.125$ , already shows tremendous errors and loss of correct rarefaction/shock features.

Our hodograph-based physics-aware DMD resolves this challenge. Figure 4.9(b) demonstrates that it yields an accurate ROM by interpolating the  $M = 250$  snapshots of the analytical solution onto the  $(\rho, x)$  plane in which the  $\rho$  coordinate is discretized with a  $N_\rho = 1000$  equidistant mesh. Here,  $x(\rho, t)$  is selected as the observable and

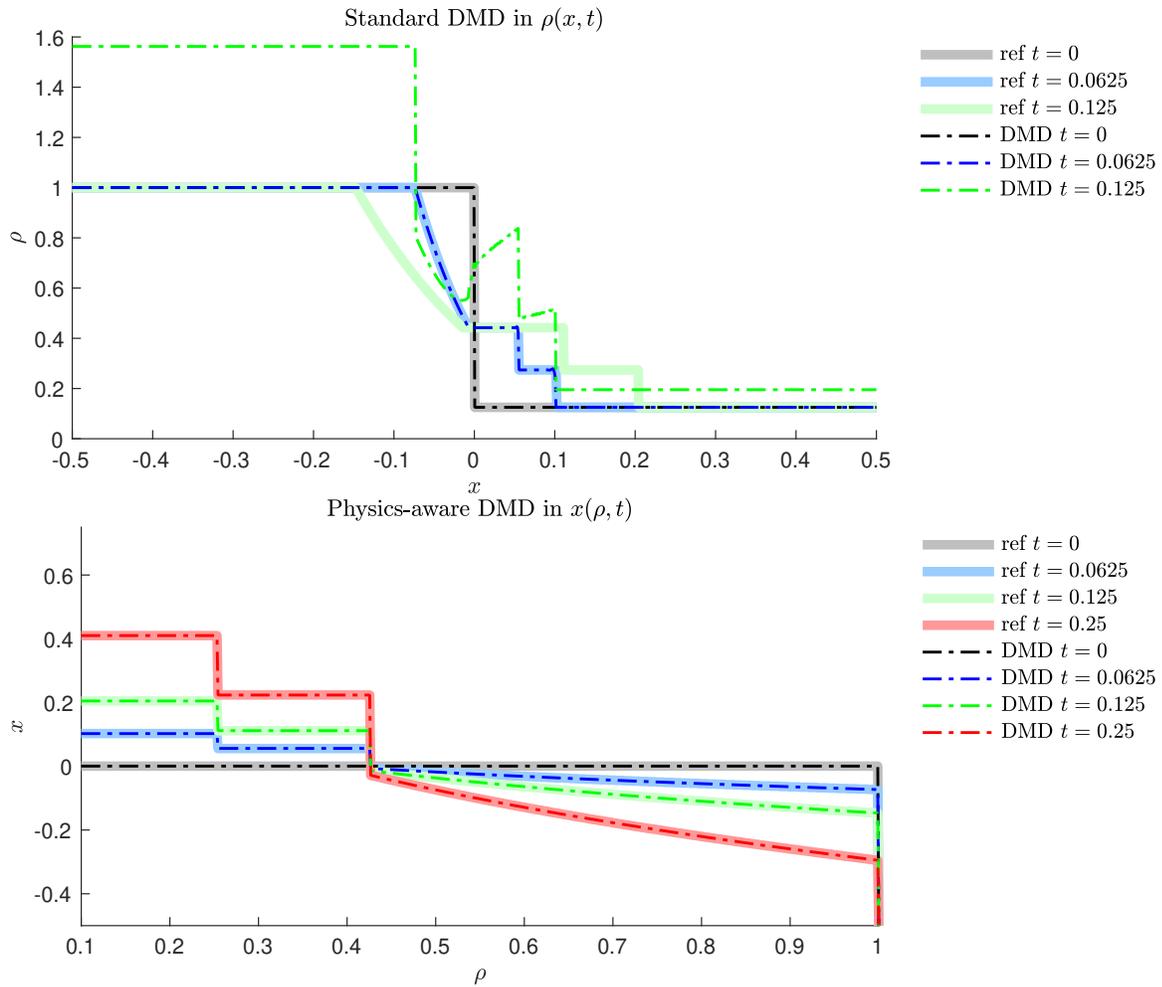


Figure 4.9: ROMs constructed by (a) top: the standard DMD and (b) bottom: the physics-aware DMD. The former uses snapshots of  $\rho(x, t)$  in the  $(x, \rho)$  plane, while the latter relies on snapshots of  $x(\rho, t)$  in the  $(\rho, x)$  plane.

the linear Koopman operator on  $x(\rho, t)$  is efficiently represented by  $r = 210$  modes in the physics-aware DMD algorithm. Nearly 4/5 rank reduction is obtained and the computational time for DMD prediction is negligible due to its iteration-free nature. Although an equation for  $x(\rho, t)$  is not available, the hodograph transform for one-dimensional scalar hyperbolic equations motivates this proper selection of an observable. The hodograph transform for multidimensional PDEs is more complicated (e.g., [152, 153]) and not directly applicable to our current ROM framework. Our results demonstrate that while the connection between the hodograph transform and the Koopman theory is unclear in multiple dimensions, it can still guide the selection of observables.

Finally, we investigate the model-order reduction error  $e_x$  and the projection error  $e_{\text{pr}}$  discussed in Remark 4.9. In this example,  $e_x$  is the DMD prediction error in terms of  $x(\rho, t)$ ,

$$\begin{aligned} e_x^n &= \|\mathbf{y}^n - \mathbf{y}_{\text{DMD}}^n\|, \\ \mathbf{y}^n &= [x(\bar{\rho}_1, t^n), \dots, x(\bar{\rho}_{N_\rho}, t^n)]^\top \\ &= \mathbf{g}([\rho(x_1, t^n), \dots, \rho(x_{N_x}, t^n)]^\top). \end{aligned} \quad (4.30)$$

This quantity provides a measure of the discrepancy between the reference observable values and their DMD prediction, without isolating various sources of the error. We also consider  $e_{\text{pr}}$ , the error in creating the observable  $x(\rho, t)$  via projection from the  $(x, \rho)$  plane onto the  $(\rho, x)$  plane during preprocessing and in estimating the solution  $\rho(x, t)$  backwards during post-processing. Both  $e_x$  and  $e_{\text{pr}}$  contribute to the total error  $e_\rho$  in DMD estimation of the solution  $\rho(x, t)$ ,

$$\begin{aligned} e_\rho^n &= \|\boldsymbol{\rho}^n - \boldsymbol{\rho}_{\text{DMD}}^n\|, \\ \boldsymbol{\rho}^n &= [\rho(x_1, t^n), \dots, \rho(x_{N_x}, t^n)]^\top, \\ \boldsymbol{\rho}_{\text{DMD}}^n &= \mathbf{g}^{-1}(\mathbf{y}_{\text{DMD}}^n). \end{aligned} \quad (4.31)$$

Temporal evolution of the errors  $e_x$  and  $e_\rho$  is plotted in Figure 4.10 with both errors (4.30) and (4.31) reported in  $L_2$  norm and defined on the corresponding meshes  $N_\rho = 1000$  and  $N_x = 1000$ . The model-order-reduction error  $e_x$  decreases with the

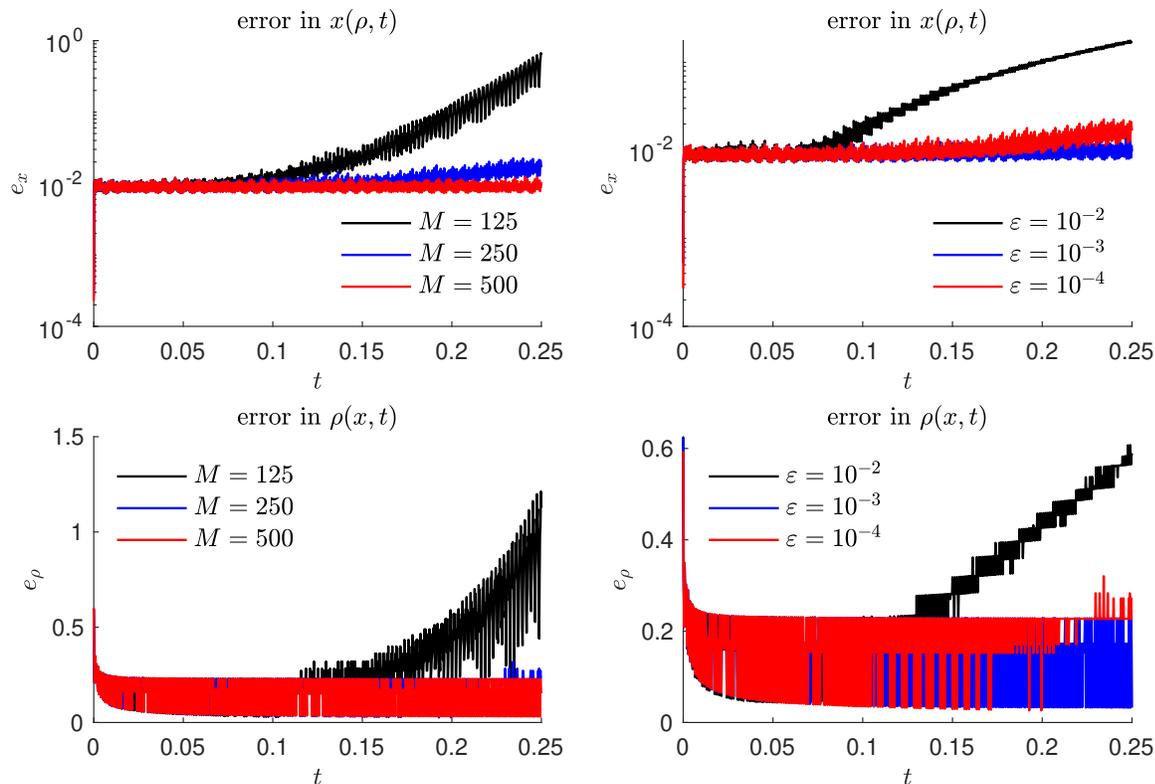


Figure 4.10: Prediction errors of the ROMs constructed by the physics-aware DMD with different parameters. The top row shows the model-order reduction error  $e_x$  defined in (4.30); the total DMD error  $e_\rho$ , defined in (4.31), is shown in the bottom row.

number of snapshots  $M$ , resulting in a more accurate prediction. This is consistent with the intuition that DMD can better capture the dynamics by learning from richer/larger data sets. For example, insufficient data ( $M = 125$ ) fail to sample the essential features in the dynamics of  $x(\rho, t)$ . The rank truncation  $\varepsilon$  also plays a crucial role in the model-order-reduction error  $e_x$ : significant accuracy is sacrificed if essential singular values are truncated in SVD, e.g., by setting  $\varepsilon = 10^{-2}$ ; but retaining too many singular values, e.g., by setting  $\varepsilon = 10^{-4}$ , increases the impact of noise. This issue has been discussed in several DMD studies, e.g., by [23]. The choice of a rank-truncation criteria is nontrivial and, thus, the optimal truncation is problem-dependent. The total DMD error  $e_\rho$  is shown in the bottom row of Figure 4.10. In addition to  $e_x$ ,

this error also accounts for the projection between the  $(x, \rho)$  plane and the  $(\rho, x)$  plane, which was conducted via a shape-preserving interpolation method. For a well controlled model-reduction error,  $e_x \sim \mathcal{O}(10^{-2})$ , the total error,  $e_\rho \sim \mathcal{O}(10^{-1})$ , is dominated by the projection error. For insufficient data ( $M = 125$ ) or low rank truncation ( $\varepsilon = 10^{-2}$ ), the large total error in the prediction regime,  $e_\rho \sim \mathcal{O}(1)$ , is dominated by the model-reduction error,  $e_x \sim \mathcal{O}(1)$ . The time evolution of the errors is oscillatory due to the interpolation during the transformation between observables and state-space.

## 4.5 Summary

The Lagrangian DMD [30] provides a robust tool to construct ROMs of hyperbolic conservation laws, a class of problems for which standard (Eulerian) DMD methods fail. However, this algorithm is limited to problems that admit smooth strong solutions. We extended it to problems with shocks and rarefaction waves, thus addressing a long-standing challenge in ROM construction. This challenge stems from severe grid distortion typical of Lagrangian POD and DMD algorithms. Lacking information about shocks and discontinuities, DMD mode projection from the HFM to a ROM does not preserve the topological structure of the interface where characteristic lines cross each other. We resolved this issue by combining hodograph transformation with physics-aware DMD algorithm [30]. The relevant research codes are available at [https://github.com/DDMS-ERE-Stanford/dmd\\_hodograph](https://github.com/DDMS-ERE-Stanford/dmd_hodograph).

Hodograph transforms are consistent with the Koopman operator theory in that both aim to identify linear structures in the underlying nonlinear dynamics. Our physics-aware DMD algorithm enhanced by hodograph transformation is capable of predicting the dynamics of weak solutions, which satisfies the entropy condition. We demonstrated the accuracy and robustness of our algorithm on several numerical tests.

To the best of our knowledge, our study is the first to establish a connection between hodograph transformation and the Koopman operators. By providing a principled way for identifying the observables needed by the Koopman operator theory, this connection opens a door to construct ROMs for a wide range of nonlinear

PDEs that are linearizable by hodograph transformation [149]. There is an algorithmic method to do the linearization via *extended* hodograph transforms. As a result, one can take advantage of the linearity and design robust iteration-free physics-aware DMD. Moreover, data-driven modeling and uncertainty quantification can be further explored using this framework. Our numerical experiments demonstrated that many physical quantities, such as the shock speed in Burgers' equation and the mobility constant in the Buckley-Leverett equation, can be learned from (simulation) data as long as one analyzes them in a "smart" way.

We established a connection between the hodograph transforms and the Koopman operator theory for one-dimensional scalar hyperbolic PDEs. The similar idea was carried over to one-dimensional hyperbolic systems. The construction of ROMs for multi-dimensional hyperbolic systems remains an open challenge. We leave such multidimensional interpretations of the Koopman operator theory in terms of hodograph (or other) transforms for future studies. Another future work direction is to improve the current framework in handling experimental data, which are potentially contaminated by measurement noise. Due to the regularity at the shock front, one would expect Eulerian DMD approaches and regular DMD approaches to have stability problems. The current framework is expected to be more robust as the hodograph transform improves the regularity at the shock front (i.e., shocks become flat constants). For mixed wave problems, the current framework can still be sensitive to noise at the intersection of different waves. In an attempt to deal with experimental data, we would consider combining the proposed method with noise filters.

# Chapter 5

## Extended DMD for Inhomogeneous Problems

Complexity of many, if not most, physical and biological phenomena and paucity of measurements undermine the reliability of purely statistical descriptors. Instead, models of such systems are inferred or “learned” from both observational and simulated data and reflect the fundamental laws of nature (e.g., conservation of mass and energy). Various sparse regression techniques [44, 45, 154, 155] use a proposed dictionary to “discover” the governing equations from data. The dictionary, comprising plausible spatial and/or temporal derivatives of a state variable, provides functional approximations of different physical laws; DMD was used to compute the modes of the Koopman operator approximated from a preselected dictionary basis [65]. Subsequent studies provided a theoretical analysis of convergence [115], practical guidelines for efficient construction of the dictionary [63], and other aspects of this approach. The data for sparse regression are allowed to be noisy [156], corrupted [157], and limited [158]. Various flavors of deep neural networks (DNN) provide a related dictionary-based approach to PDE learning [159, 160]. These and other techniques of equation discovery are as good as a dictionary on which they are based.

A conceptually different, dictionary-free, framework for data-informed predictions is to construct a surrogate (aka reduced-order) model, instead of learning a governing

PDE. This framework is often classified as “unsupervised learning” and “equation-free”. Much of the research in this field deals with dynamical systems, for which training data are generated by either ODEs or PDEs after spatial discretization. In this context, DMD can be used to construct an optimal linear approximation model for the unknown system [22] and to learn the unknown dynamics of chosen observables, rather than of the state itself [161]. The latter is accomplished by utilizing the Koopman theory [39] in order to construct linear models on the observable space, instead of seeking for nonlinear models on the state space [162]. Physics-guided selection of observables provides not only better accuracy [2, 30, 150], but also a bridge between the understanding of data and physics. Likewise, DNN can be used to build nonlinear surrogate models for ODEs [29, 163] and PDEs [164–166]. DNN-based surrogates and ROMs [28, 167] are invaluable in applications that require a large number of model solves, such as inverse modeling [168, 169] and uncertainty quantification [170–173].

Our study contributes to this second framework by extending the range of applicability of DMD-based ROMs to dynamical systems described by inhomogeneous PDEs with inhomogeneous boundary conditions. In various application scenarios, many variants of the standard DMD algorithms were introduced to improve the performance of model reduction. The low-rank approximators are determined from optimization problems that are adapted to control inputs [62], time embeddings [75], higher-order approximations [174], optimal approximation error [175], etc. We adopt a similar methodology by modifying the optimization problem in a way that allows us to cope with the inhomogeneity of the problems. A major benefit of our new algorithm is a theoretically guaranteed accuracy improvement relative to the standard DMD algorithm with awareness and identification of the inhomogeneity at almost no extra computational cost. Our extended dynamic mode decomposition (xDMD) borrows ideas from the recent work on residual neural networks (ResNet) to provide an optimal linear approximation model for such systems. Our generalization of the standard DMD includes two ingredients: an added bias term and residual learning. The first builds upon the generalized ResNet [176] that introduces a bias term to model the dynamics described by underlying inhomogeneous ODEs. We extend this idea

to systems described by inhomogeneous PDEs and prove the accuracy improvement induced by the added bias term. The second ingredient of xDMD is the learning of effective increments (i.e., the residual of subtracting identity from a flow map) rather than the flow map itself. Although mathematically equivalent to flow map learning, this strategy proved to be highly advantageous in practice and gained traction in the deep-learning community [177], including in its applications to equation recovery [29]. To the best of our knowledge, xDMD is first to fuse these two features and to provide a theoretical estimate.

## 5.1 Problem Formulation and Extended DMD

We consider a group of PDEs belonging to (1.1) where

$$f(u, \mathbf{x}) = f_d(u) + f_s(\mathbf{x}). \quad (5.1)$$

$f_d(u)$  is a (linear or nonlinear) differential operator that involves spatial derivatives;  $f_s(\mathbf{x})$  represent sources/sinks. A suitable numerical approximation of (1.1) yields (1.2), where  $\mathbf{s}$  comes from both  $f_s(\mathbf{x})$  and  $b(\mathbf{x})$ .

Our goal is to learn the dynamic system  $\mathbf{f}$  in (1.2), or, more precisely, its reduced-order surrogate, using  $M$  temporal snapshots of the solutions (1.4). The simulation data consist of  $M - 1$  pairs  $\{(\mathbf{u}^k, \mathbf{u}^{k+1})\}_{k=0}^{M-2}$ , such that

$$\mathbf{u}^{k+1} = \mathcal{F}_{\Delta t}(\mathbf{u}^k; \mathbf{s}), \quad k = 0, \dots, M - 2. \quad (5.2)$$

**Lemma 5.1.** *Assume  $\mathbf{f}$  to be Lipschitz continuous with a Lipschitz constant  $L$  on a solution manifold  $\mathcal{M} \subset \mathbb{R}^{N^*}$ . Define*

$$\mathcal{M}_{\Delta t} = \{\mathbf{u} \in \mathcal{M} : \mathcal{F}_{\Delta t}(\mathbf{u}; \mathbf{s}) \in \mathcal{M}\}. \quad (5.3)$$

*Then,  $\mathcal{F}_{\Delta t}$  is Lipschitz continuous on  $\mathcal{M}_{\Delta t}$ . Specifically, for any  $\mathbf{z}, \tilde{\mathbf{z}} \in \mathcal{M}_{\Delta t}$ ,*

$$\|\mathcal{F}_{\Delta t}(\mathbf{z}; \mathbf{s}) - \mathcal{F}_{\Delta t}(\tilde{\mathbf{z}}; \mathbf{s})\| \leq e^{L\tau} \|\mathbf{z} - \tilde{\mathbf{z}}\|, \quad 0 \leq \tau \leq \Delta t. \quad (5.4)$$

*Proof.* The proof follows directly from the classical numerical analysis results in, e.g., [178, p. 109].  $\square$

Lemma 5.1 imposes requirements on the snapshots data pairs  $\{(\mathbf{u}^k, \mathbf{u}^{k+1})\}_{k=0}^{M-2}$ : the number of data pairs  $M - 1$  should be sufficiently large, and the data should be sufficiently rich for the data space to cover the solution space of interest. These requirements are consistent with the core of the Koopman operator theory, which underpins the DMD algorithm, e.g., [23, p. 47] and others [43, 65, 161]. The error analysis of the DMD algorithms [2, 115] (see Chapter 2) also verifies the impact of the selection of observables on the success of Koopman methods.

### 5.1.1 Standard DMD

Given a dataset of snapshots,  $\{(\mathbf{u}^k, \mathbf{u}^{k+1})\}_{k=0}^{M-2}$ , DMD constructs a best-fit linear operator  $\mathbf{K} \in \mathbb{R}^{N_x \times N_x}$  such that

$$\mathbf{u}^{k+1} \approx \mathbf{K}\mathbf{x}^k, \quad k = 0, \dots, M - 2. \quad (5.5)$$

Therefore, the matrix  $\mathbf{K}$  is determined in a least square sense by (1.16). Typically,  $\mathbf{K}$  is computed from (1.16) as

$$\mathbf{K} = \mathbf{X}_2 \mathbf{X}_1^\dagger \quad (\text{standard DMD}), \quad (5.6)$$

where  $\dagger$  denotes the Moore-Penrose inverse.

**Remark 5.1.** *The Moore-Penrose inverse is computed via SVD, which requires certain truncation criteria to maintain computational stability. In all our numerical tests, we use the default truncation in the `pinv` command of Matlab.*

**Remark 5.2.** *In a typical DMD algorithm, e.g., [23, p. 7], a reduced-order model  $\mathbf{K}_r$  is derived by projecting  $\mathbf{K}$  onto the POD modes. Since the major goal of our study is to obtain a linear approximation model of inhomogeneous PDEs, for which standard DMD algorithms fail, we omit the order-reduction procedure for simplicity.*

### 5.1.2 Generalized DMD

In order to cope with potential inhomogeneity of the underlying dynamics, the following modification is made in [176]:

$$\mathbf{u}^{k+1} \approx \mathbf{K}_g \mathbf{u}^k + \mathbf{b}, \quad k = 0, \dots, M - 2. \quad (5.7)$$

The matrix  $\mathbf{K}_g$  and the vector  $\mathbf{b} \in \mathbb{R}^{N_x}$  are computed by solving the optimization problem

$$(\mathbf{K}_g, \mathbf{b}) = \underset{\hat{\mathbf{K}} \in \mathbb{R}^{N_x \times N_x}, \hat{\mathbf{b}} \in \mathbb{R}^{N_x}}{\operatorname{argmin}} \frac{1}{M-1} \sum_{k=0}^{M-2} \|\mathbf{u}^{k+1} - \hat{\mathbf{K}}\mathbf{u}^k - \hat{\mathbf{b}}\|^2. \quad (5.8)$$

Let us introduce

$$\tilde{\mathbf{X}}_1 := \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{1} \end{bmatrix}_{(N_x+1) \times (M-1)} \quad (5.9)$$

where  $\mathbf{1} := [1, 1, \dots, 1]$  is a vector of size  $1 \times (M - 1)$ . Then  $\mathbf{K}_g$  and  $\mathbf{b}$  are obtained by

$$[\mathbf{K}_g, \mathbf{b}] = \mathbf{X}_2 \tilde{\mathbf{X}}_1^\dagger \quad (\text{generalized DMD or gDMD}). \quad (5.10)$$

**Remark 5.3.** *gDMD can be regarded as a special arrangement of DMD with control (DMDc) [62]. In the latter, the augmented data matrix (5.9) is constructed by stacking the state variable snapshots  $\mathbf{u}^k$  and the control input snapshots, which in our context are set to 1 since there is no control. Applications of DMD and the Koopman theory in control are an active research area [179–182], but lie outside the scope of our analysis that focuses on improving the performances of DMD in learning unknown dynamical systems. Of more direct relevance is a connection of gDMD and DMD to dictionary learning (e.g., [63, 65, 115]): the dictionary composition is treated as the state variable itself and set to 1 in the augmented matrix. However, the computational cost of identifying relevant terms from a proposed dictionary can be prohibitively large for high-dimensional dynamical systems and discretized PDEs. Instead, our framework includes only the bias term, which has physical interpretations in inhomogeneous PDEs.*

### 5.1.3 Residual DMD

The residual DMD or rDMD borrows a key idea behind ResNet. The latter explicitly introduces the identity operator in a neural network and forces the network to approximate the “residual” of the input-output map. Although mathematically equivalent, this simple transformation proved to improve network performance and became increasingly popular in the machine learning community.

Writing  $\mathbf{K} = \mathbf{I} + \mathbf{B}$ , where  $\mathbf{I}$  is the  $(N_{\mathbf{x}} \times N_{\mathbf{x}})$  identity matrix and  $\mathbf{B}$  is the remainder, recasts (5.5) as

$$\mathbf{u}^{k+1} \approx \mathbf{u}^k + \mathbf{B}\mathbf{u}^k. \quad (5.11)$$

The matrix  $\mathbf{B}$  is determined by

$$\mathbf{B} = (\mathbf{X}_2 - \mathbf{X}_1)\mathbf{X}_1^\dagger \quad (\text{residual DMD or rDMD}). \quad (5.12)$$

It provides an approximation of the “effective increment” [29, definition 3.1],  $\varphi_{\Delta t}$ , that is defined as

$$\varphi_{\Delta t}(\mathbf{u}; \mathbf{f}, \mathbf{s}) = \Delta t \mathbf{f}(\mathcal{F}_\tau(\mathbf{u}; \mathbf{s})) \quad (5.13)$$

for some  $0 \leq \tau \leq \Delta t$  such that

$$\mathbf{u}(t + \Delta t) = \mathbf{u}(t) + \varphi_{\Delta t}(\mathbf{u}; \mathbf{f}, \mathbf{s}). \quad (5.14)$$

### 5.1.4 Extended DMD

Combining the modification used in the previous two subsections, we arrive at our extended DMD or xDMD,

$$\mathbf{u}^k \approx \mathbf{u}^k + \mathbf{B}_g \mathbf{u}^k + \mathbf{b}, \quad (5.15)$$

where  $\mathbf{B}_g$  and  $\mathbf{b}$  are computed as

$$[\mathbf{B}_g, \mathbf{b}] = (\mathbf{X}_2 - \mathbf{X}_1)\tilde{\mathbf{X}}_1^\dagger \quad (\text{extended DMD or xDMD}). \quad (5.16)$$

## 5.2 Relative Performance of Different DMD Formulations

**Theorem 5.1.** *In the least square sense, gDMD in section 5.1.2 fits the  $M$  snapshots data  $\mathbf{X}_1$  and  $\mathbf{X}_2$  better than the standard DMD from section 5.1.1 does, i.e.,*

$$\frac{1}{M-1} \sum_{k=0}^{M-2} \|\mathbf{u}^{k+1} - \mathbf{K}_g \mathbf{u}^k - \mathbf{b}\|^2 \leq \frac{1}{M-1} \sum_{k=0}^{M-2} \|\mathbf{u}^{k+1} - \mathbf{K} \mathbf{u}^k\|^2. \quad (5.17)$$

*Proof.* The optimization problem (5.8) gives rise to

$$\begin{aligned} & \frac{1}{M-1} \sum_{k=0}^{M-2} \|\mathbf{u}^{k+1} - \mathbf{K}_g \mathbf{u}^k - \mathbf{b}\|^2 \\ &= \min_{\hat{\mathbf{K}} \in \mathbb{R}^{N_x \times N_x}, \hat{\mathbf{b}} \in \mathbb{R}^{N_x}} \frac{1}{M-1} \sum_{k=0}^{M-2} \|\mathbf{u}^{k+1} - \hat{\mathbf{K}} \mathbf{u}^k - \hat{\mathbf{b}}\|^2 \\ &\leq \min_{\hat{\mathbf{K}} \in \mathbb{R}^{N_x \times N_x}, \hat{\mathbf{b}} \in \mathbb{R}^{N_x}} \frac{1}{M-1} \sum_{k=0}^{M-2} \left( \|\mathbf{u}^{k+1} - \hat{\mathbf{K}} \mathbf{u}^k\|^2 + \|\hat{\mathbf{b}}\|^2 \right) \\ &= \min_{\hat{\mathbf{K}} \in \mathbb{R}^{N_x \times N_x}} \left( \frac{1}{M-1} \sum_{k=0}^{M-2} \|\mathbf{u}^{k+1} - \hat{\mathbf{K}} \mathbf{u}^k\|^2 \right) + \min_{\hat{\mathbf{b}} \in \mathbb{R}^{N_x}} \|\hat{\mathbf{b}}\|^2 \\ &= \frac{1}{M-1} \sum_{k=0}^{M-2} \|\mathbf{u}^{k+1} - \mathbf{K} \mathbf{u}^k\|^2 + \min_{\hat{\mathbf{b}} \in \mathbb{R}^{N_x}} \|\hat{\mathbf{b}}\|^2. \end{aligned} \quad (5.18)$$

The inequality is derived by triangle inequality and the last equality is achieved by (1.16). Since the equality is achieved with  $\hat{\mathbf{b}} = 0$ , gDMD is equivalent to the standard DMD only when the bias term  $\mathbf{b} = 0$ .  $\square$

**Remark 5.4.** *Theorem 5.1 implies that xDMD from section 5.1.4 fits the  $M$  snapshots data  $\mathbf{X}_1$  and  $\mathbf{X}_2$  better than rDMD from section 5.1.3 in the least square sense, i.e.,*

$$\frac{1}{M-1} \sum_{k=0}^{M-2} \|\mathbf{u}^{k+1} - \mathbf{u}^k - \mathbf{B}_g \mathbf{u}^k - \mathbf{b}\|^2 \leq \frac{1}{M-1} \sum_{k=0}^{M-2} \|\mathbf{u}^{k+1} - \mathbf{u}^k - \mathbf{B} \mathbf{u}^k\|^2. \quad (5.19)$$

**Corollary 5.1.1.** *Let  $\mu_M$  be an empirical measure defined on a given dataset  $\{\mathbf{u}^0, \dots, \mathbf{u}^{M-2}\}$  by*

$$\mu_M = \frac{1}{M-1} \sum_{k=0}^{M-2} \delta_{\mathbf{u}^k}, \quad (5.20)$$

where  $\delta_{\mathbf{u}^k}$  denotes the Dirac measure at  $\mathbf{u}^k$ . Then, for any  $\mathbf{u} \in \mathcal{M}_{\Delta t}$ ,

$$\|\mathcal{F}_{\Delta t}(\mathbf{u}) - \mathbf{K}_g \mathbf{u} - \mathbf{b}\|^2 \leq \|\mathcal{F}_{\Delta t}(\mathbf{u}) - \mathbf{K} \mathbf{u}\|^2 \quad \text{a.s.}, \quad (5.21)$$

i.e., the inequality (5.21) holds in the sense of distribution.

*Proof.* The integral of a test function  $g$  with respect to  $\mu_M$  is given by

$$\int_{\mathcal{M}} g(\mathbf{u}) d\mu_M(\mathbf{u}) = \frac{1}{M-1} \sum_{k=0}^{M-2} g(\mathbf{u}^k). \quad (5.22)$$

It follows from (5.17) and the definition of  $\mathbf{u}^{k+1}$  in (5.2) that

$$\frac{1}{M-1} \sum_{k=0}^{M-2} \left( \|\mathcal{F}_{\Delta t}(\mathbf{u}^k) - \mathbf{K} \mathbf{u}^k\|^2 - \|\mathcal{F}_{\Delta t}(\mathbf{u}^k) - \mathbf{K}_g \mathbf{u}^k - \mathbf{b}\|^2 \right) \geq 0. \quad (5.23)$$

Thus, by virtue of (5.22),

$$\int_{\mathcal{M}} \left( \|\mathcal{F}_{\Delta t}(\mathbf{u}) - \mathbf{K} \mathbf{u}\|^2 - \|\mathcal{F}_{\Delta t}(\mathbf{u}) - \mathbf{K}_g \mathbf{u} - \mathbf{b}\|^2 \right) d\mu_M(\mathbf{u}) \geq 0. \quad (5.24)$$

Hence, the inequality (5.21) holds in the sense of distributions.  $\square$

**Remark 5.5.** *By the same token,*

$$\|\varphi_{\Delta t}(\mathbf{u}) - \mathbf{B}_g \mathbf{u} - \mathbf{b}\|^2 \leq \|\varphi_{\Delta t}(\mathbf{u}) - \mathbf{B} \mathbf{u}\|^2, \quad \text{a.s.} \quad (5.25)$$

**Theorem 5.2.** *Suppose that the assumptions of Lemma 5.1 hold, and further assume that*

1.  $\|\mathcal{F}_{\Delta t} - \mathbf{K} \mathbf{u}\|_{L^\infty(\mathcal{M}_{\Delta t})} < +\infty$  and  $\|\mathcal{F}_{\Delta t} - \mathbf{K}_g \mathbf{u} - \mathbf{b}\|_{L^\infty(\mathcal{M}_{\Delta t})} < +\infty$ ;
2.  $\mathbf{u}^k \in \mathcal{M}_{\Delta t}$  for  $k = 0, \dots, M-1$ .

Let  $\mathbf{u}_{\text{DMD}}^n$  and  $\mathbf{u}_{\text{gDMD}}^n$  denote solutions, at time  $t_n \equiv t_0 + n\Delta t$ , of the DMD and gDMD models, respectively. Let  $\mathbf{u}^n$  denote the true solution at time  $t_n$ , induced by the flow map  $\mathcal{F}_{\Delta t}$ . Then errors of the DMD and gDMD models at time  $t_n$ ,

$$\mathcal{E}_{\text{DMD}}^n = \|\mathbf{u}^n - \mathbf{u}_{\text{DMD}}^n\|^2 \quad \text{and} \quad \mathcal{E}_{\text{gDMD}}^n = \|\mathbf{u}^n - \mathbf{u}_{\text{gDMD}}^n\|^2, \quad (5.26)$$

satisfy inequalities

$$\begin{aligned} \mathcal{E}_{\text{DMD}}^n &\leq e^{nL\Delta t} \mathcal{E}_{\text{DMD}}^0 + \|\mathcal{F}_{\Delta t} - \mathbf{K}\|_{L^\infty(\mathcal{M})} \frac{e^{nL\Delta t} - 1}{e^{L\Delta t}}, \\ \mathcal{E}_{\text{gDMD}}^n &\leq e^{nL\Delta t} \mathcal{E}_{\text{gDMD}}^0 + \|\mathcal{F}_{\Delta t} - \mathbf{K}_g - \mathbf{b}\|_{L^\infty(\mathcal{M})} \frac{e^{nL\Delta t} - 1}{e^{L\Delta t}}, \end{aligned} \quad (5.27)$$

*Proof.* The proof follows similar derivations as Theorem 4.3 in [29] using triangle inequality:

$$\begin{aligned} \mathcal{E}_{\text{DMD}}^n &= \|\mathcal{F}_{\Delta t}(\mathbf{u}^{n-1}) - \mathbf{K}\mathbf{u}_{\text{DMD}}^{n-1}\|^2 \\ &\leq \|\mathcal{F}_{\Delta t}(\mathbf{u}^{n-1}) - \mathcal{F}_{\Delta t}(\mathbf{u}_{\text{DMD}}^{n-1})\|^2 + \|\mathcal{F}_{\Delta t}(\mathbf{u}_{\text{DMD}}^{n-1}) - \mathbf{K}\mathbf{u}_{\text{DMD}}^{n-1}\|^2 \\ &\leq e^{L\Delta t} \|\mathbf{u}_{\text{DMD}}^{n-1} - \mathbf{u}^{n-1}\|^2 + \|\mathcal{F}_{\Delta t} - \mathbf{K}\|_{L^\infty(\mathcal{M}_{\Delta t})}^2 \\ &= e^{L\Delta t} \mathcal{E}_{\text{DMD}}^{n-1} + \|\mathcal{F}_{\Delta t} - \mathbf{K}\|_{L^\infty(\mathcal{M}_{\Delta t})}^2 \\ &\leq e^{2L\Delta t} \mathcal{E}_{\text{DMD}}^{n-2} + \|\mathcal{F}_{\Delta t} - \mathbf{K}\|_{L^\infty(\mathcal{M}_{\Delta t})}^2 (1 + e^{L\Delta t}) \\ &\leq \dots \\ &\leq e^{nL\Delta t} \mathcal{E}_{\text{DMD}}^0 + \|\mathcal{F}_{\Delta t} - \mathbf{K}\|_{L^\infty(\mathcal{M}_{\Delta t})}^2 \sum_{k=0}^{n-1} e^{kL\Delta t} \\ &= e^{nL\Delta t} \mathcal{E}_{\text{DMD}}^0 + \|\mathcal{F}_{\Delta t} - \mathbf{K}\|_{L^\infty(\mathcal{M})} \frac{e^{nL\Delta t} - 1}{e^{L\Delta t}}. \end{aligned} \quad (5.28)$$

A proof for the error bound for  $\mathcal{E}_{\text{gDMD}}^n$  is similar.  $\square$

**Remark 5.6.** The above error estimates indicate that gDMD has a tighter error bound than DMD a.s. because Corollary 5.1.1 indicates  $\|\mathcal{F}_{\Delta t} - \mathbf{K}_g - \mathbf{b}\|_{L^\infty(\mathcal{M}_{\Delta t})}^2 \leq \|\mathcal{F}_{\Delta t} - \mathbf{K}\|_{L^\infty(\mathcal{M}_{\Delta t})}^2$  a.s..

**Remark 5.7.** Similarly, xDMD has a tighter error bound than rDMD a.s.

**Remark 5.8.** *This error bound provides a general guideline for the error growth, in order to compare the DMD and gDMD models. The magnitude of the errors depends on the specific dynamics of the flow map  $\mathcal{F}_{\Delta t}$ . Many DMD studies (e.g., [2]) have showed that the linear operator  $\mathbf{K}$  is not guaranteed to be a good approximator of the general flow map  $\mathcal{F}_{\Delta t}$ , especially when the latter is highly nonlinear. In another word,  $\|\mathcal{F}_{\Delta t} - \mathbf{K}\|_{L^\infty(\mathcal{M}_{\Delta t})}^2$  and, similarly,  $\|\mathcal{F}_{\Delta t} - \mathbf{K} - \mathbf{b}\|_{L^\infty(\mathcal{M}_{\Delta t})}^2$  can be large in the error bound estimate. A way to construct ROMs in these highly nonlinear scenarios is to approximate the so-called “Koopman operator” via mapping the state variables onto observables. The discussion is beyond the scope of this work; we refer the interested reader to [23, 39].*

### 5.3 Numerical Tests of Extended DMD

We use a series of numerical experiments to demonstrate that xDMD outperforms other DMD variants and to validate our error estimates. Additional numerical experiments are reported in Appendix C. Snapshots (training data) are obtained from reference solutions during time  $[0, T]$ , with input-output time-lag  $\Delta t$ , i.e., (1.18).

These datasets are assumed to be sufficiently large and rich to satisfy Lemma 5.1. We construct DMD and xDMD (and the other intermediate variants) by finding the best fit  $\mathbf{K}$  or  $(\mathbf{B}_g, \mathbf{b})$ , which yields a set of linear approximation models for the  $\Delta t$  time-lag input and output. The ability of learning the unknown dynamics is tested in terms of

- *Representation:* Compare the difference between  $\mathbf{u}^k$  and  $\mathbf{K}^k \mathbf{u}^0$ , or between  $\mathbf{u}^k$  and  $(\mathbf{I} + \mathbf{B}_g)^k \mathbf{u}^0 + \sum_{i=0}^{k-1} (\mathbf{I} + \mathbf{B}_g)^i \mathbf{b}$ , for  $k = 1, \dots, M - 1$ . The error is essentially the least square fitting error, aka “training error” in machine learning. The training error reflects the accuracy of a trained ROM on the available  $M - 1$  training data points in  $\mathbf{X}_2$  of (1.18). It is usually embedded in the target “loss function”, e.g., (1.16) for DMD, which is optimized in the process of training. The trained model is then validated on another, different than  $\mathbf{X}_2$ , dataset (i.e., test dataset); the resulting discrepancy is referred to as “test error”. The test error reflects the accuracy and robustness of the trained model; small test

errors indicate that the obtained ROMs reflect physics rather than simply fit the data. We use this metric to investigate our ROM's capacity for extrapolation, interpolation and generalizability.

- *Extrapolation:* Draw another set of the reference solution  $\{\mathbf{u}^k\}_{k=M}^{2M}$  from time interval  $[T, 2T]$  following the same  $\Delta t$  time-lag trajectory for the convenience of testing. Compare the difference between  $\mathbf{u}^k$  and  $\mathbf{K}^k \mathbf{u}^0$  (DMD extrapolation to  $[T, 2T]$ ), and between  $\mathbf{u}^k$  and  $(\mathbf{I} + \mathbf{B}_g)^k \mathbf{u}^0 + \sum_{i=0}^{k-1} (\mathbf{I} + \mathbf{B}_g)^i \mathbf{b}$  (xDMD extrapolation to  $[T, 2T]$ ), for  $k = M, \dots, 2M$ .
- *Interpolation:* Select a random subset of the dataset, i.e.,

$$\mathbf{X}_{1s} = \begin{bmatrix} | & | & \dots & | \\ \mathbf{u}^{s_0} & \mathbf{u}^{s_1} & \dots & \mathbf{u}^{s_m} \\ | & | & & | \end{bmatrix}, \quad \mathbf{X}_{2s} = \begin{bmatrix} | & | & \dots & | \\ \mathbf{u}^{s_0+1} & \mathbf{u}^{s_1+1} & \dots & \mathbf{u}^{s_m+1} \\ | & | & & | \end{bmatrix}, \quad (5.29)$$

where  $s_0 = 0$ ,  $\{s_1, \dots, s_m\} \subset \{0, \dots, M-1\}$ , with  $m < M$ . Then determine  $\mathbf{K}$  and  $(\mathbf{B}_g, \mathbf{b})$  based on the selected dataset  $\mathbf{X}_{1s}$  and  $\mathbf{X}_{2s}$ . Compare the difference between  $\mathbf{u}^k$  and  $\mathbf{K}^k \mathbf{u}^0$  (DMD interpolation to  $[0, T]$ ), and between  $\mathbf{u}^k$  and  $(\mathbf{I} + \mathbf{B}_g)^k \mathbf{u}^0 + \sum_{i=0}^{k-1} (\mathbf{I} + \mathbf{B}_g)^i \mathbf{b}$  (xDMD interpolation to  $[0, T]$ ), for  $k = 1, \dots, M-1$ . In our examples, the selected number of snapshots,  $m$ , is smaller than  $M/2$ .

- *Generalizability:* Determine  $\mathbf{K}$  and  $(\mathbf{B}_g, \mathbf{b})$  from the datasets  $\mathbf{X}_1$  and  $\mathbf{X}_2$ , and obtain a linear approximation model of the discretized PDE. Compute another set of reference solutions  $\{\mathbf{v}^k\}_{k=0}^{M-1}$  from a different initial input  $\mathbf{v}^0 \neq \mathbf{u}^0$  and the same boundary condition and source. Compare the difference between  $\mathbf{v}^k$  and  $\mathbf{K}^k \mathbf{v}^0$ , and between  $\mathbf{v}^k$  and  $(\mathbf{I} + \mathbf{B}_g)^k \mathbf{v}^0 + \sum_{i=0}^{k-1} (\mathbf{I} + \mathbf{B}_g)^i \mathbf{b}$ , for  $k = 1, \dots, M-1$ . In our examples, the input  $\mathbf{v}^0$  has completely different features than the training  $\mathbf{u}^0$ .
- *Accuracy:* The accuracy is compared in terms of the log relative errors,

$$\varepsilon_{\text{DMD}}^n := \lg \left( \frac{\|\mathbf{u}^n - \mathbf{u}_{\text{DMD}}^n\|_2^2}{\|\mathbf{u}^n\|_2^2} \right), \quad \varepsilon_{\text{xDMD}}^n := \lg \left( \frac{\|\mathbf{u}^n - \mathbf{u}_{\text{xDMD}}^n\|_2^2}{\|\mathbf{u}^n\|_2^2} \right), \quad (5.30)$$

where  $\|\cdot\|_2$  denotes the  $L_2$  norm.

All comparisons between DMD and xDMD are made using the same dataset and the same SVD truncation criteria in the pseudo-inverse part (using the default truncation criteria in Matlab).

### 5.3.1 Inhomogeneous PDEs

We start by examining the performance of the aforementioned DMD variants in learning a PDE with inhomogeneous source terms. Consider a one-dimensional diffusion equation with a source and homogeneous boundary conditions,

$$\begin{cases} \frac{\partial u}{\partial t} = 0.1 \frac{\partial^2 u}{\partial x^2} + S(x), & x \in (0, 1), \quad t > 0.1; \\ u(x, 0) = \exp[-20(x - 0.5)^2]; \\ u_x(0, t) = 0, \quad u_x(1, t) = 0. \end{cases} \quad (5.31)$$

The reference solution is obtained by an implicit finite-difference scheme with  $\Delta x = 0.01$  and  $\Delta t = 0.01$ . Training datasets consist of  $M = 80$  snapshots collected from  $t = 0$  to  $t = 0.8$ . The extrapolation is tested from  $t = 0.8$  to  $t = 1.6$ . The interpolation training set consists of  $m = 20$  snapshots randomly selected from the  $M = 80$  snapshots.

The left column of Figure 5.1 provides a comparison between the reference solution and its DMD and xDMD approximations in the three modes: representation, extrapolation and interpolation. The DMD and xDMD models have a reduced rank of 17. As predicted by the theory, DMD fails in all three modes. For a fixed time, the DMD error grows with  $x$ , which is to be expected since standard DMD algorithms are not designed to handle inhomogeneous PDEs, such as (5.31) in which the source term is  $S(x) = x$ . If a source term lies outside the span of the training data, as happens in this test, then it cannot be represented as a linear combination of the available snapshots. The DMD model always lies within the span of the training data, while the true solution grows out of that subspace because of the source. On the other hand, the xDMD model captures the true solution in all modes thanks to the bias

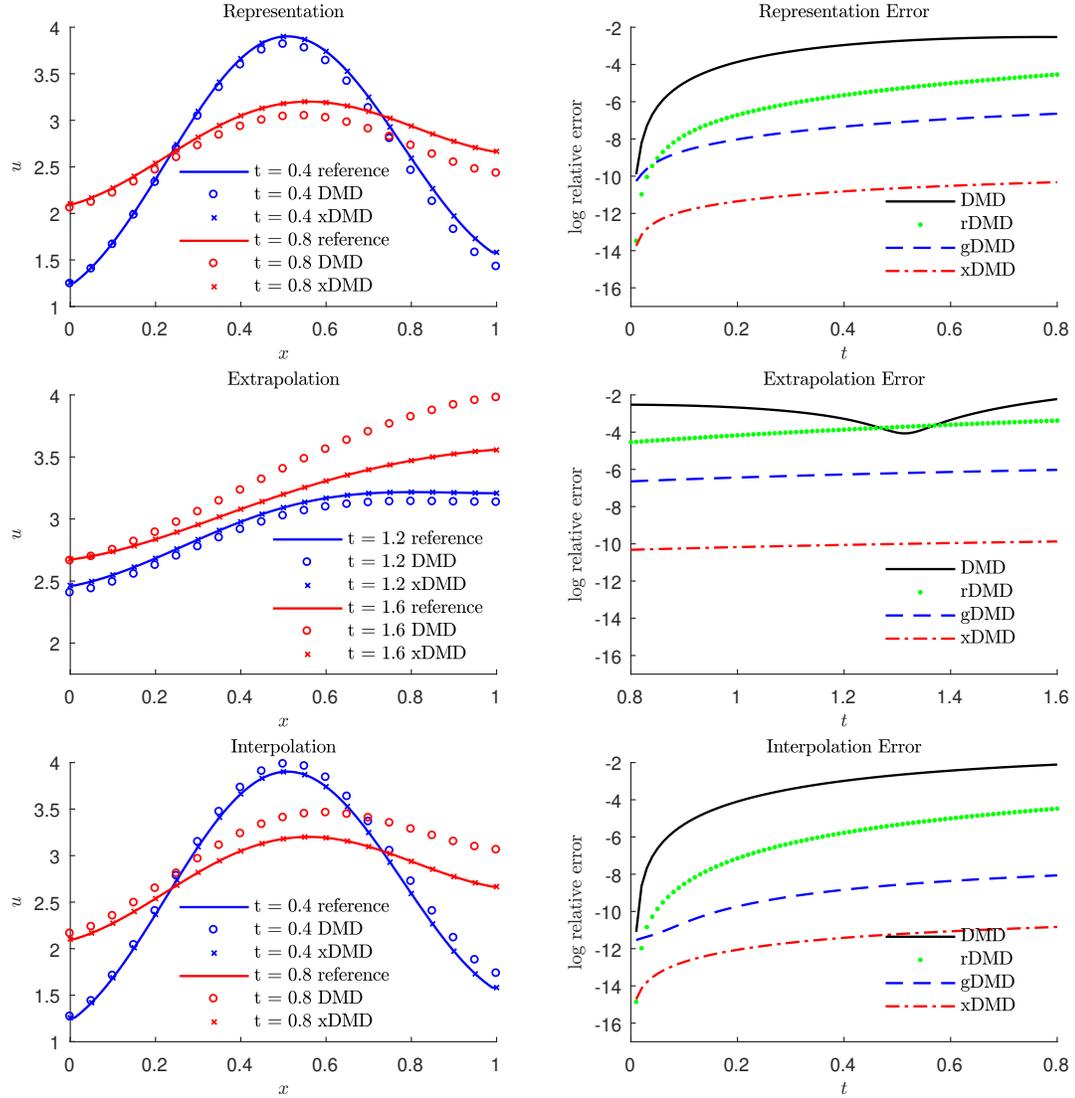


Figure 5.1: Reference solution of (5.31) and its DMD (5.6) and xDMD (5.16) approximations (left column); the log relative error of the DMD (5.6), gDMD (5.10), rDMD (5.12), and xDMD (5.16) models (right column).

term that accounts for the solution expansion outside the training data span.

The right column of Figure 5.1 shows the accuracy of the DMD (5.6), gDMD (5.10), rDMD (5.12), and xDMD (5.16) models. Although DMD and rDMD are mathematically equivalent, the identity subtraction in rDMD reduces the solution error in all three modes (representation, extrapolation, and interpolation). Addition of the bias term in xDMD contributes to further orders-of-magnitude reduction in the error, consistent with the theoretical proof in section 5.2. In all modes, the proposed xDMD outperforms the other DMD variants by several orders of magnitude, achieving almost machine accuracy.

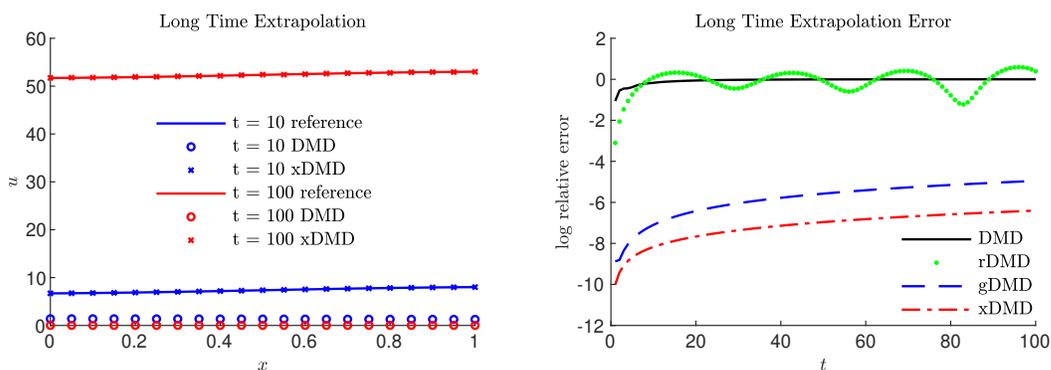


Figure 5.2: Reference solution of (5.31) and its DMD (5.6) and xDMD (5.16) approximations (left); the log relative error of the DMD (5.6), gDMD (5.10), rDMD (5.12), and xDMD (5.16) models (right) for long time extrapolation.

Figure 5.2 shows the performance of different ROMs in long-time extrapolation, up to  $t = 100$ . Exhibiting errors that grow slowly in time, the gDMD and xDMD predictions accurately capture the underlying dynamics for very long time due to the advantageous role of the bias term. That is in contrast to the DMD and rDMD models, which make poor unphysical predictions without awareness of the inhomogeneous source term.

An added benefit of gDMD and xDMD is their ability to infer a source function,  $S(x)$ , in an inhomogeneous PDE from temporal snapshots of the solution (Figure 5.3). Both methods recover  $S(x)$ , regardless of whether it is linear ( $S = x$ ) or nonlinear ( $S = e^x$ ), and have comparable errors. While DMD lumps together the differential operator and the source, gDMD and xDMD treat them separately. This endows

them with the ability to learn both the operator (the system itself) and the source (external forces acting on the system), as long as the latter does not vary with time. This self-learning feature carries almost no extra computational cost.

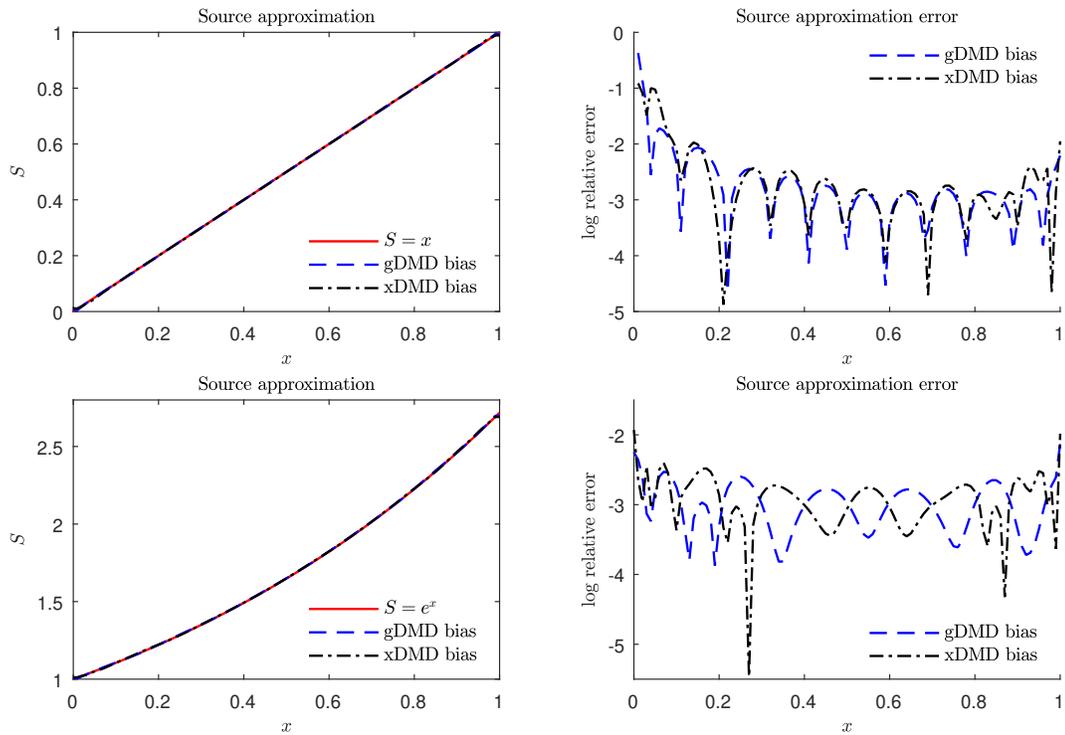


Figure 5.3: Estimation of the source term  $S(x) = x$  and  $e^x$  in (5.31) by gDMD and xDMD: the eyeball measure (left) and the log relative error (right).

### 5.3.2 Inhomogeneous Boundary Conditions and Data Errors

Next, we examine the ability of DMD and xDMD to handle inhomogeneous boundary conditions and data errors. Consider a two-dimensional diffusion equation in a multi-connected domain  $\mathcal{D}$  with inhomogeneous boundary conditions,

$$\begin{cases} \frac{\partial u}{\partial t} = \nabla^2 u, & (x, y) \in \mathcal{D}, \quad t \in (0, 10000]; \\ u(x, y, 0) = 0; \\ u(0, y, t) = 3, \quad u(800, y, t) = 1, \\ \frac{\partial u}{\partial y}(x, 0, t) = \frac{\partial u}{\partial y}(x, 800, t) = 0, \quad u(x, y, t) = 2 \quad \text{on } \partial\mathcal{S} \text{ (red)}. \end{cases} \quad (5.32)$$

The domain  $\mathcal{D}$  is the  $800 \times 800$  square with an S-shaped cavity (Figure 5.4). The Dirichlet boundary conditions are imposed on the left and right sides of the square and the cavity surface. The top and bottom of the square are impermeable. The reference solution is obtained via Matlab PDE toolbox on the finite-element mesh with 1633 elements shown in Figure 5.4. The solution from early transient time ( $t = 2000$ ) until steady state ( $t = 10000$ ) is presented in Figure 5.5.

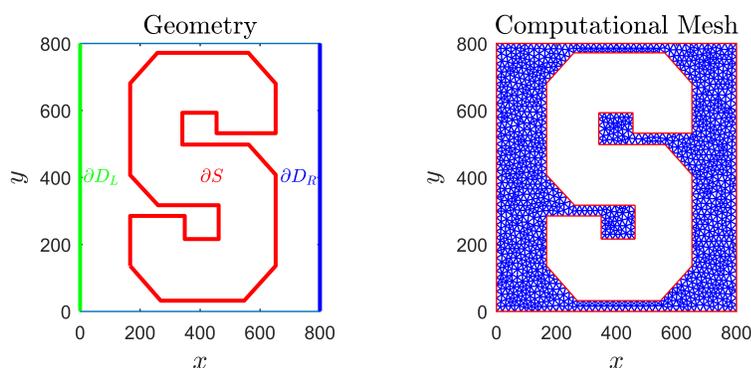


Figure 5.4: Multi-connected simulation domain  $\mathcal{D}$  (left) and the mesh used in the finite-element solution of (5.32).

With the total simulation time (time sufficient to reach steady state)  $t = 10000$ , we generate snapshots spaced by  $\Delta t = 5$  and use those to conduct four tests. First, the

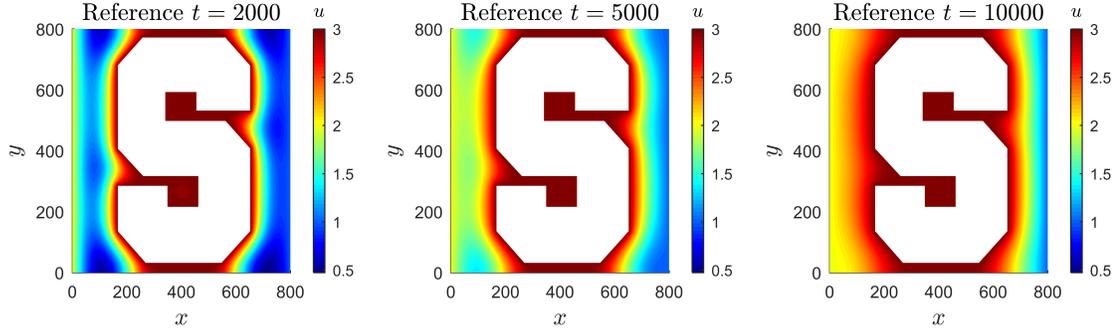


Figure 5.5: The reference solution of (5.31),  $u(x, y, t)$ , at times  $t = 2000$ ,  $t = 5000$  and  $t = 10000$ .

leading  $M = 1200$  snapshots are used to inform DMD and xDMD and to ascertain their representation errors. Second, the DMD and xDMD models are deployed to extrapolate until  $t = 10000$  and compare the extrapolation error of the two models. Third, randomly selected  $M = 600$  snapshots from the first 1200 snapshots are used for interpolation and to compare the interpolation error of DMD and xDMD. Finally, we repeat these representation/extrapolation/interpolation tests on data corrupted by addition of zero-mean white noise whose strengths at any  $(x, t)$  is 0.1% of the nominal value of  $u(x, t)$  at that point.

Figure 5.6 reveals that, for noiseless data, the accuracy of xDMD is orders of magnitude higher than that of DMD in the representation and interpolation modes with the same reduced rank of 30; in the extrapolation mode, the error is dominated by the extrapolation error, which increases with time, but xDMD is still about 9% more accurate than DMD at later times. DMD has a good performance in this case because the inhomogeneity from the boundary conditions happens to lie inside the span of the training data (i.e., can be approximated by a linear combination of the available snapshots), which is not guaranteed for all inhomogeneous boundary conditions (see a counterexample of Figure C.1 in Supplemental Material). However, in the presence of measurement noise, xDMD has no better performance than DMD, it is even less accurate in the extrapolation and interpolation regimes. This sensitivity to noise mirrors the over-fitting issue in machine learning: models with more parameters fit the limited number of available data (solution snapshots) too closely and, consequently,

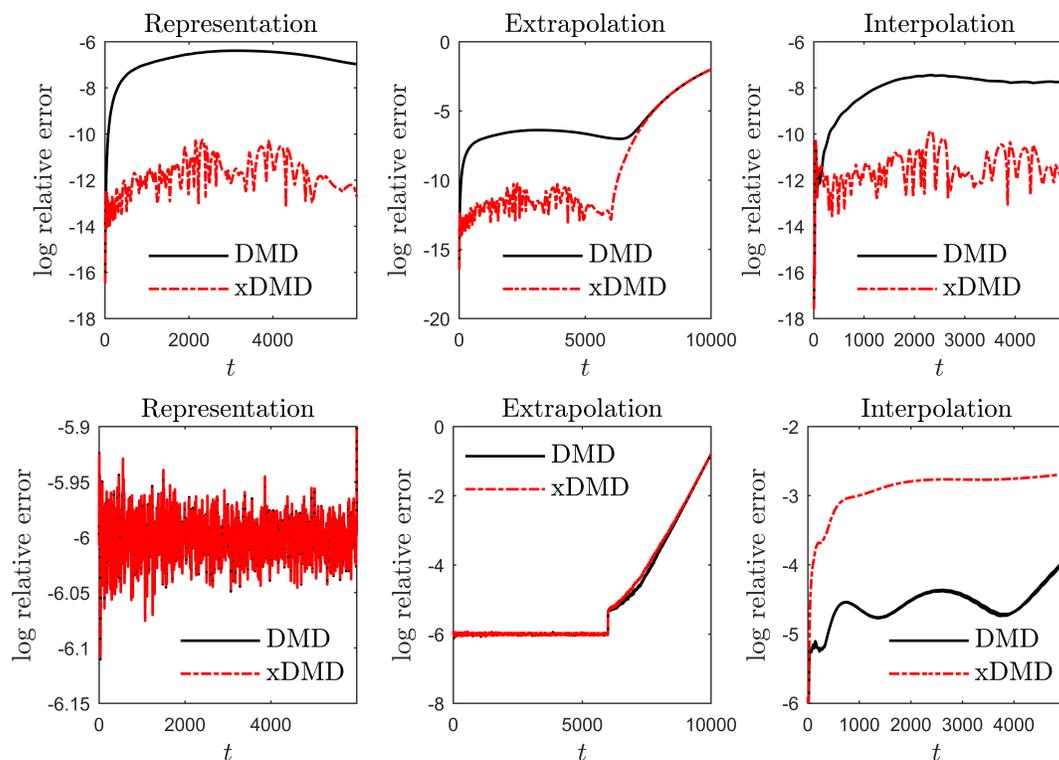


Figure 5.6: Dependence of the log relative error of the DMD and xDMD models on time in the representation, extrapolation and interpolation modes. These errors are reported for noiseless data (top row) and data corrupted by addition of zero-mean white noise whose strengths at any  $(x, t)$  is 0.1% of the nominal value of  $u(x, t)$  at that point (bottom row).

fail to fit additional data or to reliably predict future observations. Since xDMD has more parameters than DMD due to the bias term, one should expect the former to be more sensitive to noise than the latter.

### 5.3.3 Coupled Nonlinear PDEs

Common sense suggests that the success of linear models, such as DMD and xDMD, to approximate nonlinear dynamics is not guaranteed. In machine learning, data augmentation by feature map is widely used to deal with the nonlinearity. Similarly, judiciously chosen observables play a crucial role in the success of data-driven (DMD)

modeling [30, 41, 115]. The selection of observables requires prior knowledge of the underlying process, which is out of scope of this study. Instead, we assume no prior knowledge and apply no data augmentation, i.e., our observables are the state itself. To satisfy the assumptions in Lemma 5.1, we restrict our attention to nonlinear PDEs, whose solutions are confined in certain subspace  $\mathcal{M}$ . Our numerical experiments deal with the two-dimensional viscous Burgers' equation (reported in the Supplemental Material) and the two-dimensional Navier-Stokes equations. The goal of these tests is to assess the ability of DMD and xDMD to learn complex flow maps.

We consider two-dimensional flow of an incompressible fluid with density  $\rho = 1$  and dynamic viscosity  $\nu = 1/600$  (these and other quantities are reported in consistent units) around an impermeable circle of diameter  $D = 0.1$ . The flow, which takes place inside a rectangular domain  $\mathcal{D} = \{\mathbf{x} = (x, y)^\top : (x, y) \in [0, 2] \times [0, 1]\}$ , is driven by an externally imposed pressure gradient; the center of the circular inclusion is  $\mathbf{x}_{\text{circ}} = (0.3, 0.5)^\top$ . Dynamics of the three state variables, flow velocity  $\mathbf{u}(\mathbf{x}, t) = (u, v)^\top$  and fluid pressure  $p(\mathbf{x}, t)$ , is described by the two-dimensional Navier-Stokes equations,

$$\begin{cases} \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0; \\ \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = -\frac{1}{\rho} \frac{\partial p}{\partial x} + \nu \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right), \\ \frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} = -\frac{1}{\rho} \frac{\partial p}{\partial y} + \nu \left( \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right); \end{cases} \quad \mathbf{x} \in \mathcal{D}, \quad t > 0; \quad (5.33)$$

subject to initial conditions  $\mathbf{u}(x, y, 0) = (0, 0)^\top$  and  $p(x, y, 0) = 0$ ; and boundary conditions

$$\begin{aligned} p(2, y, t) = 0, \quad \frac{\partial p}{\partial \mathbf{n}}|_{\partial \mathcal{D} \setminus \{x=2\}} = 0, \quad \mathbf{u}(0, y, t) = (1, 0)^\top, \\ \frac{\partial \mathbf{u}(2, y, t)}{\partial \mathbf{n}} = 0, \quad \mathbf{u}(x, 0, t) = \mathbf{u}(x, 1, t) = \mathbf{0}. \end{aligned}$$

Here  $\mathbf{n}$  denotes the unit normal vector. This combination of parameters results in the Reynolds number  $\text{Re} = 60$ .

The reference solution is obtained with the Matlab code [183], which implements

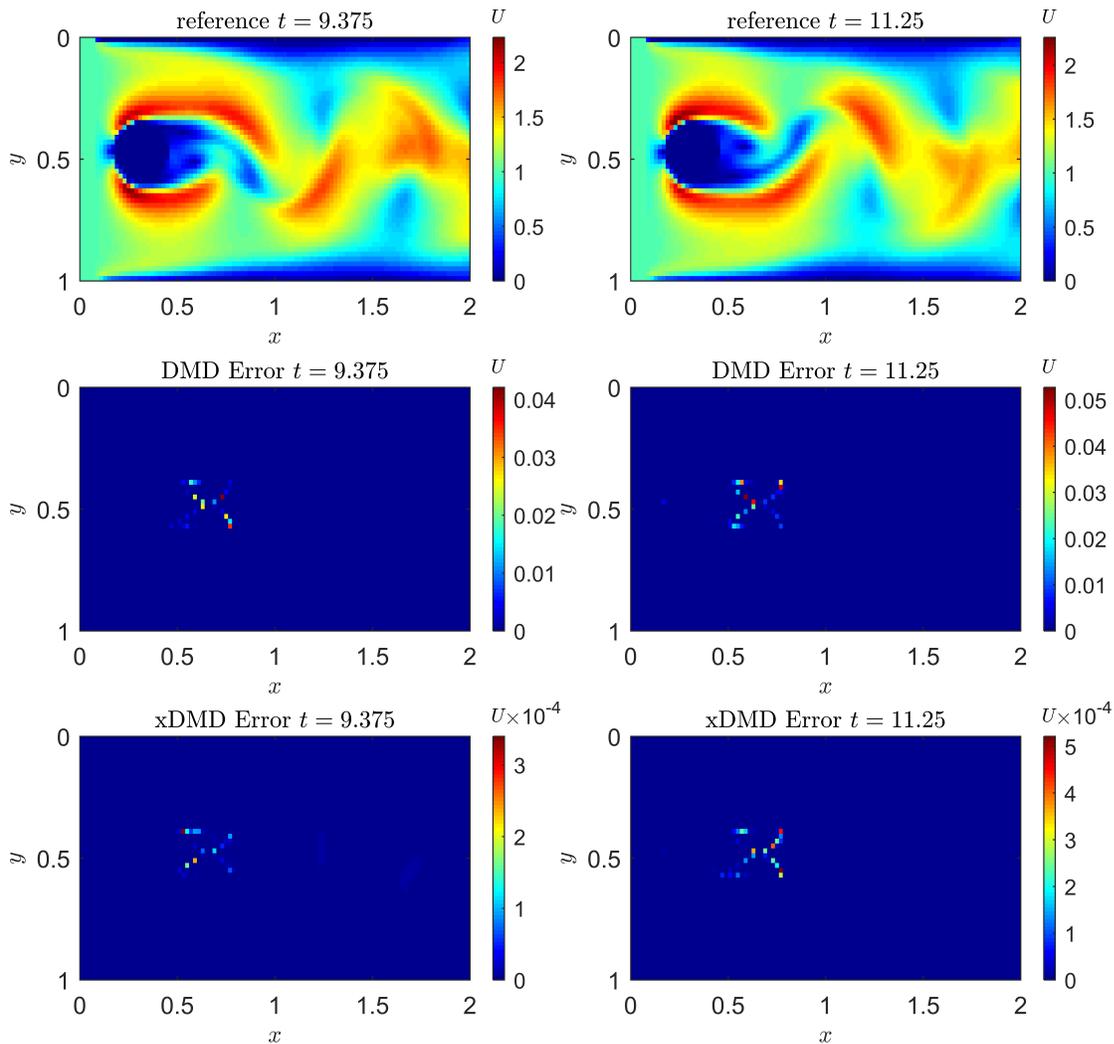


Figure 5.7: Velocity magnitude  $U = \sqrt{u^2 + v^2}$  of incompressible flow with the Reynolds number  $\text{Re} = 60$  around an impermeable circle predicted by solving numerically the two-dimensional Navier-Stokes equations (5.33) (top row) and by using the DMD and xDMD models in the representation mode. The representation errors (5.30) for these two approximations are displayed in the second and third rows, respectively.

a finite-difference scheme on the staggered grid with  $\Delta x = \Delta y = 0.02$  and  $\Delta t = 0.0015$ . Our observable (quantity of interest) is the magnitude of the flow velocity,  $U(x, y, t) = \sqrt{u^2 + v^2}$ . Visual examination of the solution  $U(x, y, t)$  reveals it to be periodic from  $t = 7.5$  to  $t = 15$  (the simulation horizon), i.e., the solution is confined in a fixed subspace  $\mathcal{M}$ . We collect  $M = 2500$  snapshots of  $U$  from  $t = 7.5$  to  $t = 11.25$  into a training dataset, from which DMD and xDMD learn the nonlinear dynamics. The discrepancy between the reference solution and its fitting with the DMD and xDMD models is the representation error.

The first row of Figure 5.7 depicts the spatial distribution of the flow speed  $U$ , at times  $t = 9.38$  and  $t = 11.25$ , computed with the (reference) solution of the Navier-Stokes equations (5.33). The DMD and xDMD models have a reduced rank of 75. Both DMD and xDMD fit the nonlinear flow data using a linear approximation with satisfactory accuracy (the last two rows of Figure 5.7). The errors are confined to the circle's wake, with xDMD being two orders of magnitude more accurate than DMD.

Next, we use the learned DMD and xDMD models in the extrapolation mode, i.e., to predict  $U(x, y, t)$  within the time interval from  $t = 11.25$  to  $t = 15$ . As shown in Figure 5.8, both DMD and xDMD yield accurate extrapolation, which should be expected due to the periodic behavior of the solution. Although the accuracy in extrapolation is diminished for both methods, xDMD remains more accurate than DMD at different extrapolation times.

Finally, Figure 5.9 exhibits the log relative error of the two methods as function of time. In the representation mode, both DMD and xDMD have nearly steady small fitting error, fluctuating about  $10^{-10}$  for xDMD and  $10^{-6}$  for DMD. The observation of xDMD's higher accuracy in fitting the data is consistent with Theorem 5.1. Similarly, the extrapolation error of DMD and xDMD validates Theorem 5.2. Although both extrapolation errors grow slowly, the xDMD error exhibits a periodic pattern (consistent with the periodic pattern of the solution  $U$ ), indicating that the xDMD linear model is able to capture the detailed periodic feature of the true flow better. Once an accurate linear representation of the nonlinear flow is available, one can conduct spatiotemporal mode analysis, reduced-order modeling and accelerated simulations. Table 5.1 collates computational times of simulating the reference solution

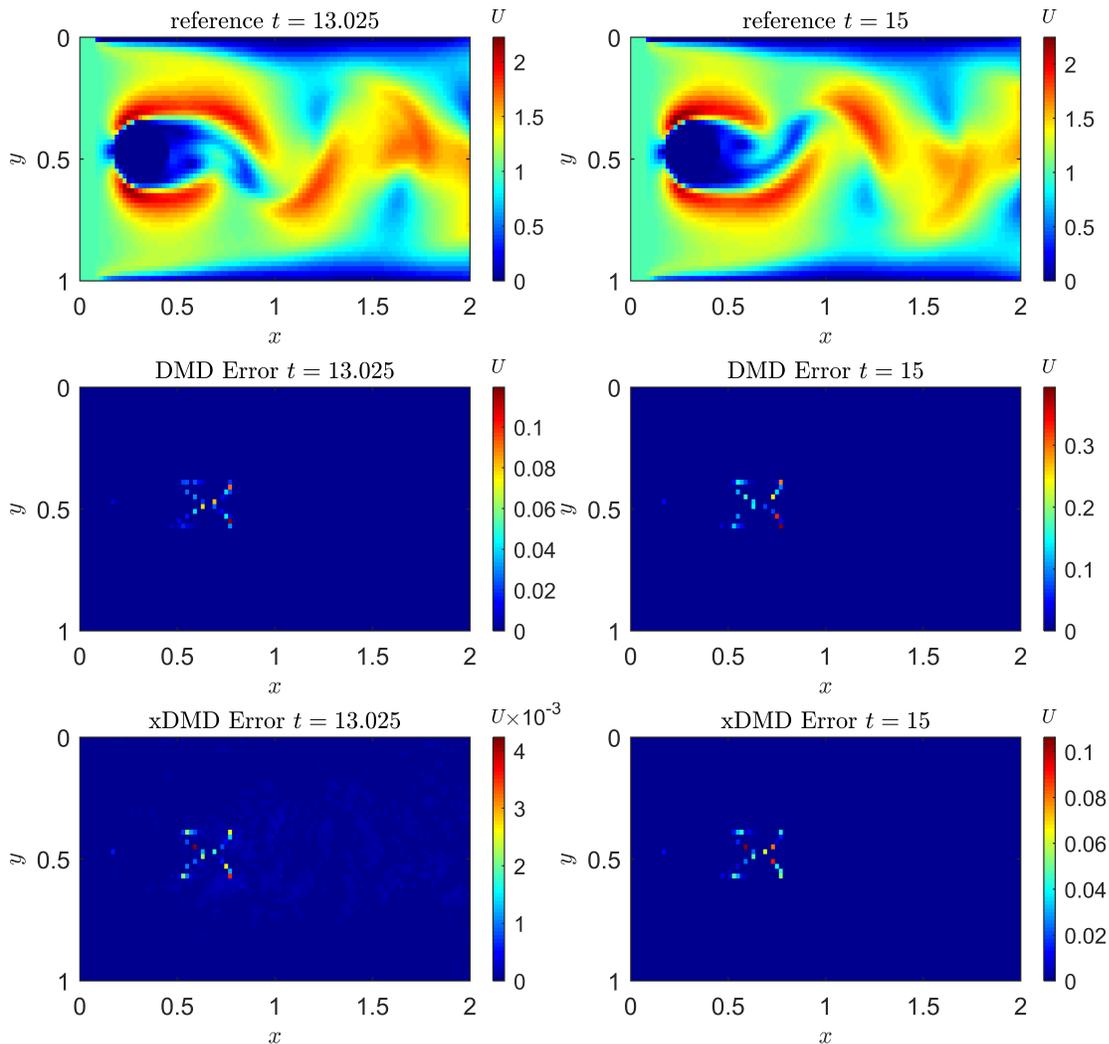


Figure 5.8: Velocity magnitude  $U = \sqrt{u^2 + v^2}$  of incompressible flow with the Reynolds number  $\text{Re} = 60$  around an impermeable circle predicted by solving numerically the two-dimensional Navier-Stokes equations (5.33) (top row) and by using the DMD and xDMD models in the extrapolation mode. The extrapolation errors (5.30) for these two approximations are displayed in the second and third rows, respectively.

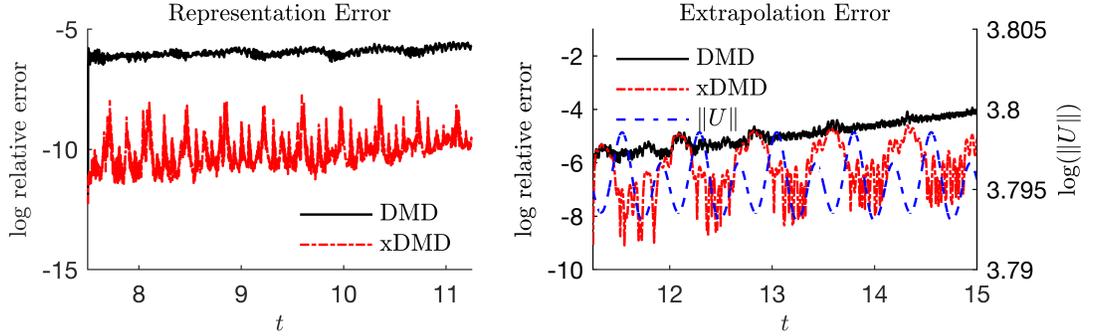


Figure 5.9: Temporal evolution of log relative error of the DMD and xDMD models in the representation and extrapolation modes.

	Simulation	DMD	xDMD
Computational time	29.0776	2.1352	2.1654
Relative error	–	$2.0515 \times 10^{-5}$	$3.1193 \times 10^{-6}$

Table 5.1: Computational time (sec) and relative error for the reference solution and the DMD and xDMD models.

and the linear approximation models. Further reduction in computation cost can be achieved by constructing reduced-order models using eigen-decomposition in DMD and xDMD.

### 5.3.4 Generalizability to New Inputs

Generalizability refers to a model’s ability to adapt properly to new, previously unseen data, drawn from the same distribution as the one used to create the model. With validated generalizability, a DMD or xDMD model can be employed as a surrogate to accelerate, e.g., expensive Markov Chain Monte Carlo (MCMC) sampling used in inverse problems. A typical setting for this type of problems is solute transport in groundwater flow, whose steady-state Darcy velocity (flux)  $\mathbf{q}(\mathbf{x}) = -K_c \nabla h$  is computed from the groundwater flow equation

$$\nabla \cdot (K_c \nabla h) = 0. \quad (5.34)$$

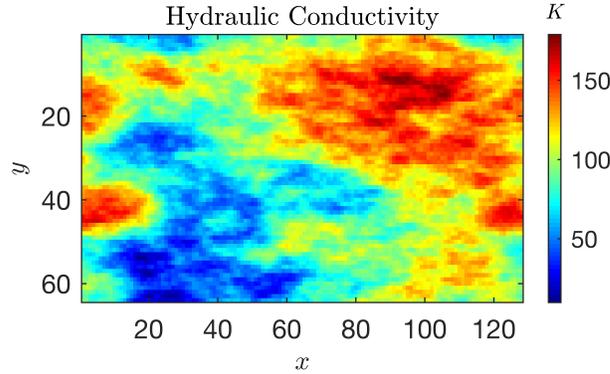


Figure 5.10: Spatial distribution of hydraulic conductivity  $K_c(\mathbf{x})$  used in our simulations.

Here  $h(\mathbf{x})$  is the hydraulic head, and  $K_c(\mathbf{x})$  is the hydraulic conductivity of a heterogeneous subsurface environment; in our simulations we use a rectangular simulation domain  $\mathcal{D} = \{\mathbf{x} = (x, y)^\top : (x, y) \in [0, 128] \times [0, 64]\}$  and the  $K_c(\mathbf{x})$  field in Figure 5.10 (these and other quantities are expressed in consistent units). The boundary conditions are  $h(x = 0, y) = 1$ ,  $h(x = 128, y) = 0$  and impermeable on  $y = 0, y = 64$ .

The resulting macroscopic velocity  $\mathbf{v}(\mathbf{x}) = \mathbf{q}/\omega$ , with  $\omega$  denoting the porosity, is then used in the advection-dispersion equation to calculate the contaminant concentration  $u(x, y, t)$ :

$$\frac{\partial u}{\partial t} + \mathbf{v} \cdot \nabla u = \nabla \cdot (\mathbf{D} \nabla u), \quad \mathbf{x} \in \mathcal{D}, \quad t \in (0, T], \quad (5.35)$$

with  $T = 80$ . In general, the dispersion coefficient  $\mathbf{D}$  is a second-rank semi-positive definite tensor, whose components depend on the magnitude of the flow velocity,  $|\mathbf{u}|$ . Here, for illustrative purposes, we treat it as the identity matrix,  $\mathbf{D} = \mathbf{I}$ . The boundary conditions for (5.35) are  $u(0, y, t) = 0.2$  and  $\partial_x u(128, y, t) = \partial_y u(x, 0, t) = \partial_y u(x, 64, t) = 0$ . The training is done for the initial condition  $u(x, y, 0) = u_{\text{in}}(x, y)$  with

$$u_{\text{in}} = s \exp[-(x - x_s)^2 - (y - y_s)^2], \quad (5.36)$$

where  $s = 100$  and the coordinates of the plume's center of mass,  $(x_s, y_s)$  are treated as independent random variables with uniform distributions,  $x_s \sim \mathcal{U}[0, 25]$  and  $y_s \sim \mathcal{U}[0, 64]$ . An example of the training dynamics is presented in Figure 5.11.

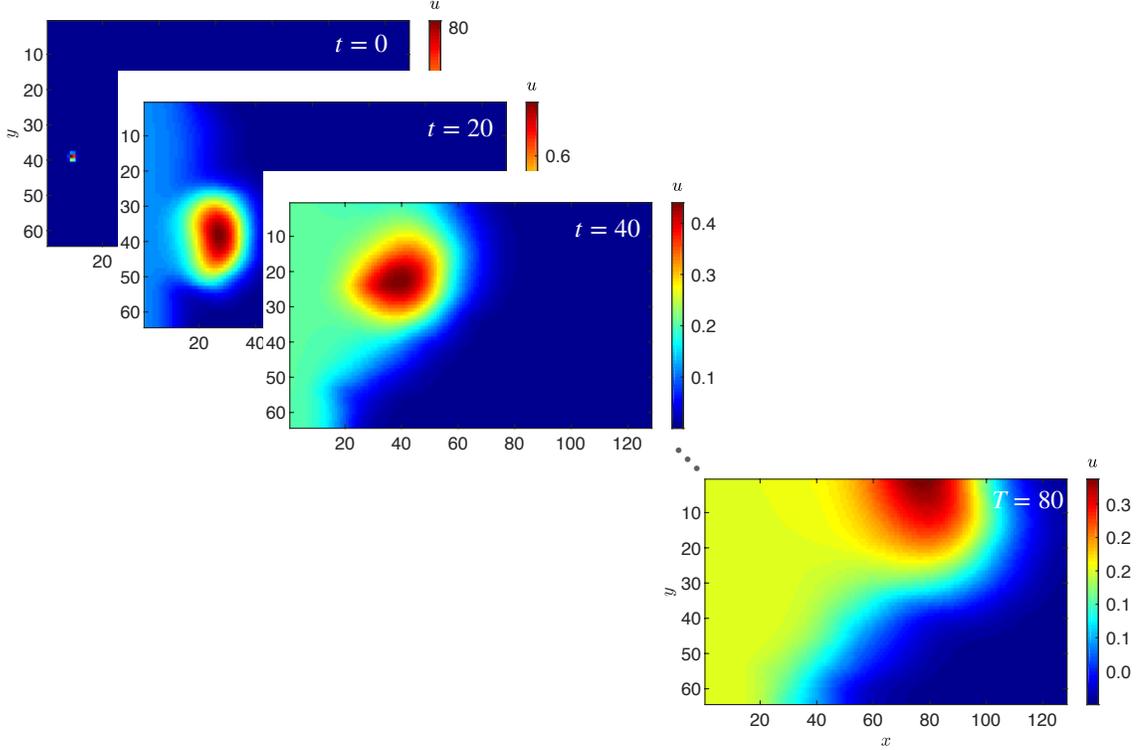


Figure 5.11: Representative snapshots of solute concentration,  $u(\mathbf{x}, t)$ , in the training dataset with initial condition (5.36).

We generate  $N_{\text{MC}}$  realizations of the pairs  $(x_s, y_s)$  and evaluate the corresponding initial conditions  $u_{\text{in}}^{(n)}(\mathbf{x})$  for  $n = 1, \dots, N_{\text{MC}}$ ,  $N_{\text{MC}} = 2000$ . For each of these realizations, (5.35) is solved<sup>1</sup> to compute our quantity of interest, the concentration field  $u_T^{(n)}(\mathbf{x}) \equiv u^{(n)}(\mathbf{x}, T)$ . The matrix pairs  $\{u_{\text{in}}^{(n)}, u_T^{(n)}\}_{n=1}^{N_{\text{MC}}}$  are arranged into

$$\mathbf{X}_1 = \begin{bmatrix} | & | & & | \\ \mathbf{u}_{\text{in}}^{(1)} & \mathbf{u}_{\text{in}}^{(2)} & \dots & \mathbf{u}_{\text{in}}^{(N_{\text{MC}})} \\ | & | & & | \end{bmatrix} \quad \text{and} \quad \mathbf{X}_2 = \begin{bmatrix} | & | & & | \\ \mathbf{u}_T^{(1)} & \mathbf{u}_T^{(2)} & \dots & \mathbf{u}_T^{(N_{\text{MC}})} \\ | & | & & | \end{bmatrix} \quad (5.37)$$

<sup>1</sup>The reference solutions are obtained with the groundwater flow simulator MODFLOW and the solute transport simulator MT3DMS, both ran on a uniform mesh  $\Delta x = \Delta y = 1$ .

where  $\mathbf{u}_{\text{in}}^n$  is vectorized  $u_{\text{in}}^{(n)}$  and  $\mathbf{u}_T^n$  is vectorized  $u_T^{(n)}$ . Finally, the DMD and xDMD models are deployed to learn the flow map  $\mathcal{F}_{\Delta t}$  with the time lag  $\Delta t = T$ .

Our goal here is to test the ability of these models to predict  $u(\mathbf{x}, T)$  for other initial conditions, such as the line source

$$u_{\text{in}} = \begin{cases} 80 & x = 10, y \in [20, 40], \\ 0 & \text{otherwise.} \end{cases} \quad (5.38)$$

Figure 5.12 exhibits the reference dynamics generated from this line source. The quantity of interest is the solute concentration map at  $T = 80$ .

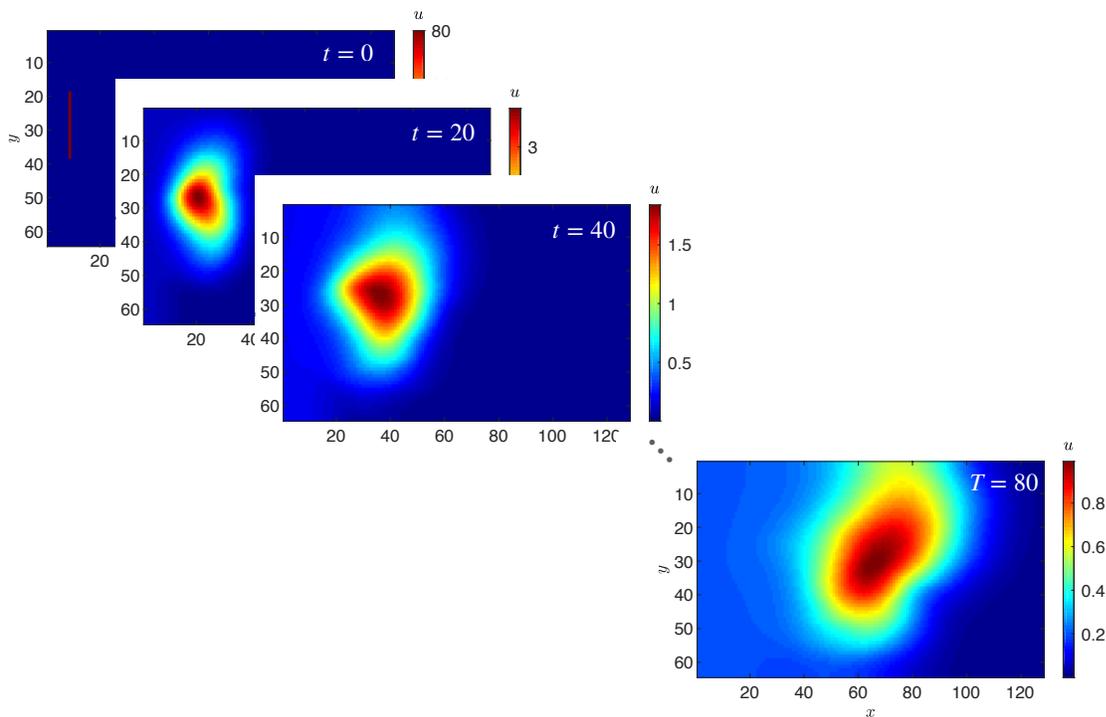


Figure 5.12: Representative snapshots of solute concentration,  $u(\mathbf{x}, t)$ , in the test case with initial condition (5.38).

In Figure 5.13, we compare the ability of DMD and xDMD to predict a quantity of interest, i.e.,  $u(\mathbf{x}, T)$ , for an initial condition that is qualitatively different from

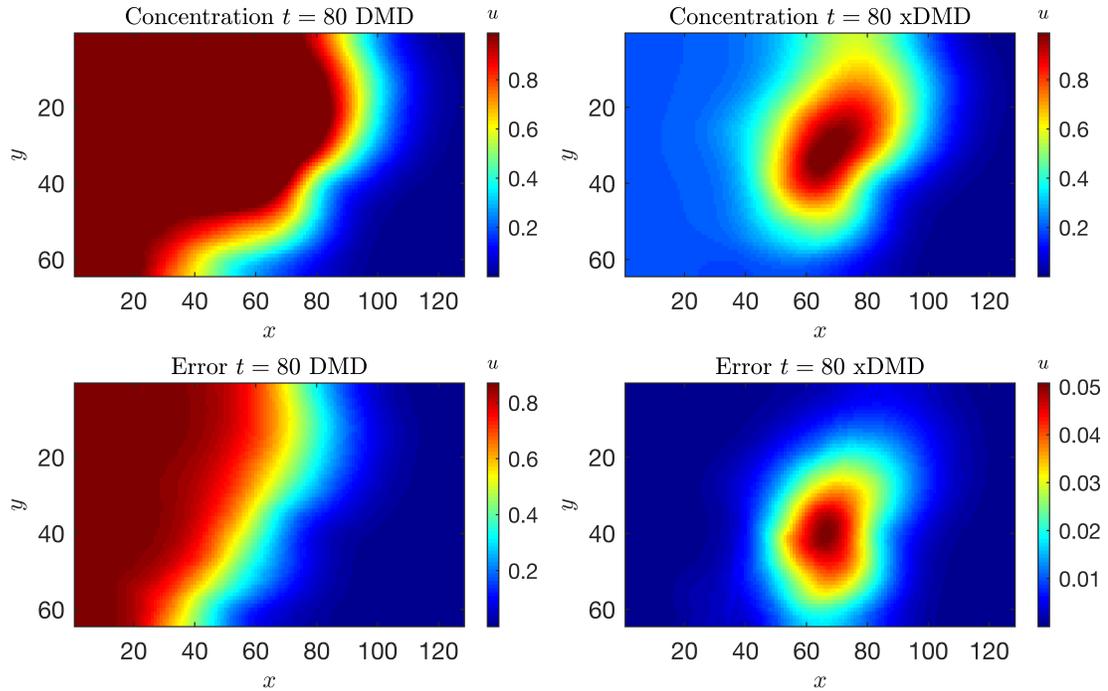


Figure 5.13: Solute concentration predicted with the DMD and xDMD models for the initial condition not seen during training. Also shown are the absolute errors of DMD and xDMD.

that for which they were trained<sup>2</sup>. The DMD and xDMD models have the reduced rank of 1648. While xDMD performs well in this generalizability test, DMD yields a wrong output concentration map because of its failure to handle inhomogeneity. The prediction error is largest in the vicinity of the left boundary, along which the inhomogeneous Dirichlet boundary condition is prescribed.

Figure 5.14 demonstrates the DMD and xDMD performance for the same task as before but when noisy data are used for training. The training data are corrupted by addition of zero-mean white noise whose strength is 0.1% of the nominal value. Although the predictions from both models are disturbed by the white noise, xDMD still captures the features of the concentration map. In contrast to Figure 5.6, the correction effects from the bias term in xDMD are more dominant than over-fitting the

<sup>2</sup>The results for the initial condition given by a linear combination of two Gaussians are presented in Appendix C.

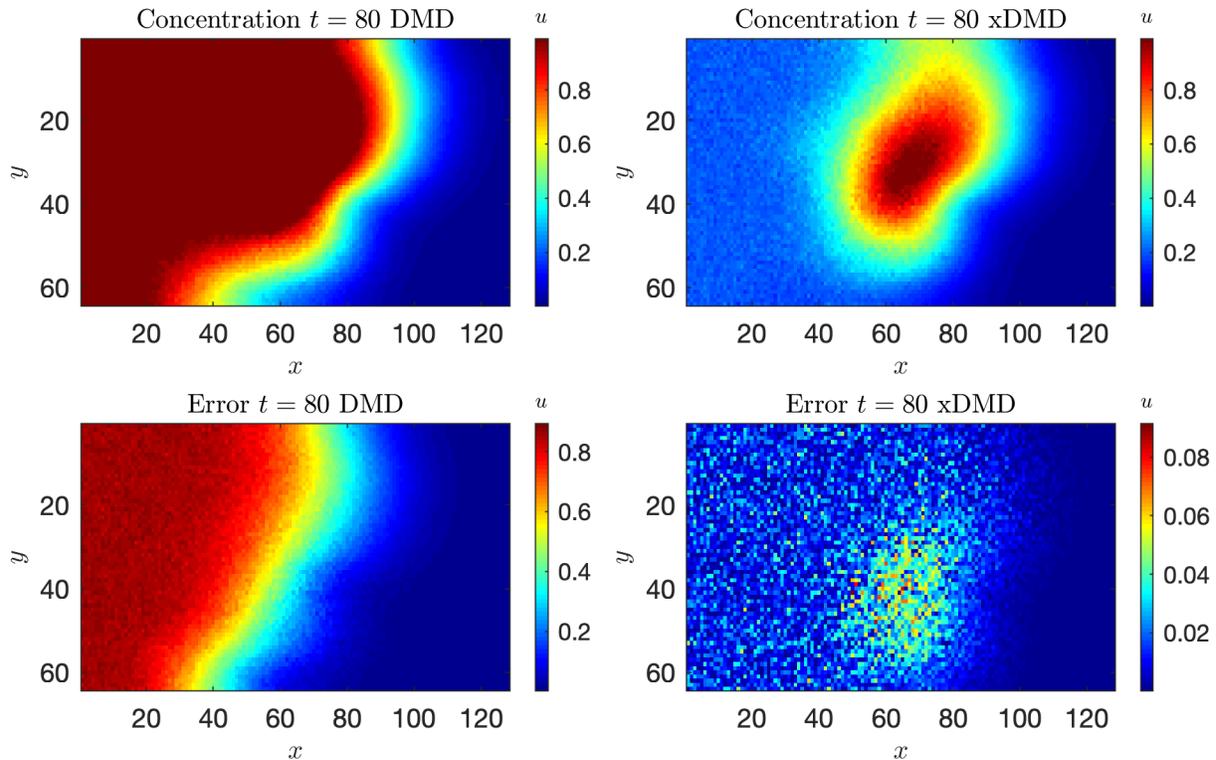


Figure 5.14: Solute concentration predicted with the DMD and xDMD models using 0.1% noisy data for the initial condition not seen during training. Also shown are the absolute errors of DMD and xDMD.

noise. The prediction error of DMD is still largest in the vicinity of the left boundary due to its failure to handle the inhomogeneous Dirichlet boundary condition.

## 5.4 Summary

We presented an extended DMD (xDMD) framework for representation of (linear or nonlinear) inhomogeneous PDEs. Our xDMD borrows from residual learning and bias identification ideas, which originated in the deep neural networks community. It shows high accuracy in learning the underlying dynamics, especially in inhomogeneous systems for which standard DMD fails. The inhomogeneous source can be accurately learned from the bias term at no extra computational cost. We conducted a number of numerical experiments to demonstrate that xDMD is an effective data-driven

modeling tool and offers better accuracy than the standard DMD.

Although xDMD provides an optimal linear approximation of the unknown dynamics, data-driven modeling for highly nonlinear PDE in general remains a challenging task. Judiciously chosen observables are needed in order to approximate the corresponding Koopman operator, which requires either prior knowledge about the dynamics or dictionary learning. Developments and experiences from deep learning may again bring potential solutions and vice versa.

In the follow-up work, we plan to use xDMD to construct surrogates, e.g., for Markov Chain Monte Carlo solutions of inverse problems and for uncertainty quantifications. The verified generalizability will allow us to replace the expensive simulation with xDMD surrogates in each Monte Carlo run. Further model reduction can be carried out to improve efficiency as well.

# Chapter 6

## Conclusion and Future Work

In this dissertation, we studied reduced-order models (ROMs) for transport phenomena - particularly, how to construct trustworthy and efficient surrogates of reduced dimension for subsurface simulations. Specifically, we introduced a physics-aware dynamic mode decomposition (DMD) framework, which combines the popular data-driven tool DMD with physics-aware ingredients. It ameliorates the following difficulties in conventional ROMs for flow and transport problems:

- We derived a theoretical error estimator for DMD extrapolation of numerical solutions, which allows one to monitor and control the errors associated with DMD-based ROMs approximating the physics-based partial differential equation (PDE) models. The global error is accumulated from a local truncation error that is based on computing the difference between the spectrum of the DMD approximation and the spectrum of the dynamical operator. Our analysis demonstrates the importance of a proper selection of observables, as predicted by the Koopman operator theory. That, in turn, facilitates the design of efficient algorithms for multi-scale/multi-physics simulations, e.g., by using ROMs as a surrogate to accelerate expensive Markov Chain Monte Carlo sampling used in inverse problems.
- Conventional DMD methods fail in the so-called *translational problems*, where wave-like behaviors play a dominant role in the dynamical systems. We first

proposed a Lagrangian-based DMD method to overcome this issue for smooth solutions. For hyperbolic conservation laws with shocks, where the Lagrangian framework is not valid anymore, we proposed a shock-preserving DMD based on a nonlinear Hodograph transformation that relies on the conservation law at hand to recover a low-rank structure. Our approaches exemplify the spirit of physics-aware DMD since they account for the evolution of characteristic lines and the information about rarefactions/shocks.

- Conventional DMD usually fails to provide accurate ROMs for inhomogeneous PDEs, where the true solutions can be driven out of the training data subspace learned by DMD models due to the inhomogeneity. To cope with the inhomogeneity in the dynamic system arising from different boundary conditions and time-independent sources, we propose an extended DMD (xDMD) approach. Motivated by similar ideas in deep neural networks (DNN), we equip our xDMD with two new features. First, it has a bias term, which serves as the physics-aware ingredient that accounts for inhomogeneity. Second, instead of learning a flow map, xDMD learns the residual increment by subtracting the identity operator. We also present a theoretical error analysis to demonstrate the improved accuracy of xDMD relative to standard DMD.

The resulting ROMs from physics-aware DMD framework are capable of capturing the key features of the underlying dynamics with higher-order accuracy than conventional DMD. Meanwhile, it takes a small fraction of the computational time of other iteration-based methods (e.g., proper orthogonal decomposition), which explains its rapid adoption by engineers in fast predictions for geo-potential fields [33], real-time control for robotic systems [34] and for wind farms [35], modeling for pulsatile blood flow [36].

Future research efforts along this line of study may include the following:

- The current framework belongs to the active research area of scientific machine learning, where data-driven modeling is enhanced by physical mechanisms and scientific knowledge. It is widely applicable to applications such

as real-time control, optimal design, data assimilation and uncertainty quantifications, etc. For a wider range of practical problems, preserving physics, e.g., preserving symplectic structure of Hamiltonian systems and preserving mass/momentum/energy of conservation laws, always plays an essential role in constructing robust and accurate models from data. It is also critical to balance the trade-off between the expressiveness of finite-dimensional functional spaces and the accuracy of the surrogate approximations on them. Closely related to this dissertation, advances in this direction will contribute to the development of machine learning interpretability.

- Many complex systems are designed and analyzed by their dependency on the parameters or random coefficients, which account for variations in shape, material, loading, and boundary and initial conditions, e.g., PDE-based models of flow and transport in (randomly) heterogeneous porous media. On one hand, DMD predictions with quantitative error bounds might provide a means for accelerating computationally expensive Monte Carlo and multi-scale simulations. On the other hand, DMD can be adapted to the model reduction strategies in a parametric setting so that the resulting ROMs for these parametric complex systems (known as PROMs) are robust with respect to the variations in parameters. Some ongoing work can be found on *arXiv* [184].
- One of the most interesting open problems for Koopman operator theory is that how to identify a principled way of determine the observable functions. We want to establish the connection between Koopman operator theory and other techniques in order to enlighten possible ways of identifying the observable space. Method of distribution, kinetic reformulation, level-set method and support vector machine have different ways to obtain linearity via dimension augmentation/feature mapping. On the other hand, kinetic equation and PDF equations, like many other high dimensional equations, need model reduction techniques to be implemented efficiently. Further understanding of the connection will lead us to efficient solvers for high dimensional PDEs as well.

# Appendix A

## Level-set DMD for Hyperbolic Conservation Laws

The level-set approach [185] provides another way to interpret conservation laws. Supposed that a state variable  $u(x, t)$  satisfies the one-dimensional conservation law

$$\frac{\partial u}{\partial t} + c(u) \frac{\partial u}{\partial x} = 0 \quad (\text{A.1})$$

with  $c(u) \geq 0$ . Its corresponding level-set formulation,

$$\frac{\partial s}{\partial t} + c(y) \frac{\partial s}{\partial x} = 0, \quad (\text{A.2})$$

is a linear two-dimensional transport equation for the dependent variable  $s(x, y, t) : \mathbb{R}^2 \times \mathbb{R}^+ \rightarrow \mathbb{R}$ . Together with a Lipschitz-continuous initial function  $s_0$ , which embeds the initial data  $u_0$  (see the example below), the zeroth-level set of  $s(x, y, t)$ , i.e., the  $x - y$  contour of the solution to  $s(x, y, t) = 0$ , gives the solution to the conservation law (A.1),  $u(x, t)$ .

By way of example, we consider the inviscid Burgers equation (3.14) with initial condition (3.15). Its level-set formulation is

$$\frac{\partial s}{\partial t} + y \frac{\partial s}{\partial x} = 0, \quad s(x, y, 0) = s_0(x, y) \equiv y - u_0. \quad (\text{A.3})$$

We apply Lagrangian DMD to construct a ROM for this two-dimensional linear advection equation from  $M = 250$  snapshots. Figure A.1 demonstrates that the ROM with the  $r = 3$  rank truncation approximates the HFM solution  $u(x, t)$  with satisfactory accuracy. Although solving a two-dimensional linear problem takes more computational time than solving the nonlinear one-dimensional problem in this case, the level-set DMD provides another venue for investigation of physics-aware DMD that might have efficient applications to other problems.

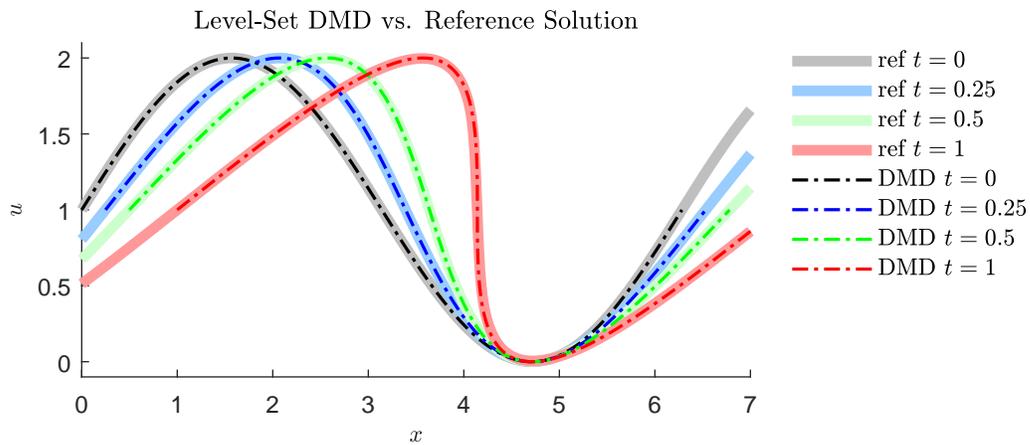


Figure A.1: Solutions of the inviscid Burgers equation,  $u(x, t)$ , alternatively obtained with the numerical method (3.2) and the ROM constructed via the level-set DMD.

# Appendix B

## Scalar Conservation Laws with Convex Fluxes

Burgers' equation has a monotonically increasing flux function. Here, we extend our analysis to smooth, strictly convex flux functions  $C(u)$ . We consider a hyperbolic conservation law (4.1) defined for  $(x, t) \in \mathbb{R} \times [0, T]$ . It is subject to the initial condition  $u(x, 0) = u_0(x)$ , where the initial data  $u_0(x)$  satisfy the following assumption.

**Assumption B.1.** *The real-valued function  $u_0(x)$  is such that*

- $\lim_{x \rightarrow \pm\infty} u_0(x) = \mp 1$ , and
- $u_0(x)$  is non-increasing and, therefore, the inverse function  $x(u_0)$  is well-defined on  $-1 \leq u_0 \leq 1$ .

**Remark B.1.** *The domain of definition,  $x \in \mathbb{R}$ , can be generalized to a finite-length interval  $(u_R, u_L)$ . The derivation is similar.*

### B.1 Solution Before Shock Formation

Similar to 4.2.1, hodograph transformation yields an equation for  $x(t, u)$ :

$$\frac{dx}{dt}(t, u) = c(u), \quad x(0, u) = x_0(u); \quad u \in (-1, 1). \quad (\text{B.1})$$

The convexity of  $C(u)$  ensures that its derivative  $c(u)$  is an increasing function. Let  $G$  denote the inverse function of  $c$ :

$$G[c(u)] = c[G(u)] = u. \quad (\text{B.2})$$

Then, defining  $y(t, u) = x[t, G(u)]$ , (B.1) becomes

$$\frac{dy}{dt}(t, u) = u, \quad y(0, u) = y_0(u) = x_0[G(u)]; \quad u \in (-1, 1). \quad (\text{B.3})$$

Differentiating both sides of this equation with respect to  $u$ ,

$$\frac{\partial^2 y}{\partial t \partial u} = 1, \quad (\text{B.4})$$

which gives

$$\frac{du}{dt}(t, u) = y'_0(u) + t. \quad (\text{B.5})$$

Therefore the shock formation time is determined by

$$t^* = -\min_u y'_0(u) = -y'_0[c(u^*)] \quad (\text{B.6})$$

## B.2 Solution After Shock Formation

The shock speed  $s$  is given by the Rankine-Hugoniot condition,

$$s = \frac{C(u_1) - C(u_2)}{u_1 - u_2}, \quad (\text{B.7})$$

where  $u_1(t)$  and  $u_2(t)$  are defined as the limits of  $u(t)$  from the top and bottom of the shock, respectively. Since  $s = dx^*/dt$ , this gives an equation for the shock trajectory  $x^*(t)$ ,

$$\frac{dx^*}{dt} = \frac{C(u_1) - C(u_2)}{u_1 - u_2}. \quad (\text{B.8})$$

A system of coupled ODEs for  $u_1(t)$  and  $u_2(t)$  is derived in [151],

$$\begin{aligned} \frac{du_1}{dt} &= C_1(u_1, u_2) \equiv \frac{1}{g(u_1) - c'(u_1)t} \left[ c(u_1) - \frac{C(u_1) - C(u_2)}{u_1 - u_2} \right] \\ \frac{du_2}{dt} &= C_2(u_1, u_2) \equiv \frac{1}{g(u_2) - c'(u_2)t} \left[ c(u_2) - \frac{C(u_1) - C(u_2)}{u_1 - u_2} \right]. \end{aligned} \quad (\text{B.9})$$

where  $g(u) = -x'_0(u)$ . These ODEs are subject to initial conditions  $u_1(t^*) = u^*$  and  $u_2(t^*) = u^*$ .

### B.3 Summary of Hodograph Solution

In summary, the reformulation for general scalar conservation law with convex flux is

$$\begin{cases} t < t^* : & \text{Equation (B.1)} \\ t > t^* : & \begin{cases} \text{Equation (B.1)} & \text{for } u \in (u_R, u_2) \cup (u_1, u_L), \\ \text{Equation (B.8)} & \text{for } u \in (u_2, u_1). \end{cases} \end{cases} \quad (\text{B.10})$$

where  $t^* = -x'_0(u^*)$ .

**Remark B.2.** *One can show that  $u_1(t)$  is monotonically increasing in time and  $u_2(t)$  is monotonically decreasing, so that*

$$u_1 \geq u^*, \quad u_2 \leq u^*, \quad x'_0(u_1) + t \leq 0, \quad x'_0(u_2) + t \leq 0. \quad (\text{B.11})$$

*In many cases of interests, and in our numerical experiment, either  $u_2 = u_R$  and  $u_1 = u_L$  or  $|u_2 - u_1| \ll \Delta t$  (so that  $u_2 \approx u_R$  and  $u_1 \approx u_L$ ). This allows one to focus on shock propagation, i.e., on (B.8), without having to solve (B.9).*

**Remark B.3.** *For more general initial condition  $u_0$ , one needs to decompose  $u_0(x)$  into regions of monotonicity. Each monotonic piece of  $u_0$  would have a unique inverse function  $x_0(u_0)$ . Then, based on the generalized entropy condition, one constructs the convex hull for the flux function  $C(u)$ , providing a way to decompose the initial data. Shock propagating initial data and rarefaction propagating initial data are determined*

*afterwards. Then, the full solution is the combination of the rarefaction pieces and the shock pieces.*

# Appendix C

## Additional Numerical Tests of Extended DMD

We provide a few additional tests used to demonstrate the relative performance of DMD and xDMD.

### C.1 Boundary Conditions and Noisy Data

We study the non-homogeneity driven by boundary conditions. Consider a one-dimensional diffusion equation,

$$\partial_t u = D \partial_{xx} u, \quad x \in [0, 1], \quad t > 0, \quad (\text{C.1})$$

with  $D = 0.1$ . Three different cases are tested to compare DMD and gDMD:

- Case 1: Dirichlet boundary conditions,

$$\begin{cases} u(x, 0) = 1, \\ u(0, t) = 3, \end{cases} \quad u(1, t) = 2. \quad (\text{C.2})$$

- Case 2: Neumann boundary conditions,

$$\begin{cases} u(x, 0) = e^{-20(x-0.5)^2}, \\ u_x(0, t) = 0, \quad u_x(1, t) = 0. \end{cases} \quad (\text{C.3})$$

- Case 3: Contaminant training data. The initial and boundary conditions are the same as in Case 1. The training data are solution to (C.1) with 0.1% measurement noise.

The same spatiotemporal discretization as in Test 5.3.1 is used, with the same number of training data. The solution behavior is trivial and, thus, not shown here. The relative accuracy of DMD and xDMD is compared in Figure C.1 in terms of representation, extrapolation and interpolation.

In Case 1, xDMD exhibits the higher-order accuracy than DMD in all three regimes of representation, extrapolation and interpolation. DMD captures the overall solution behavior because the diffusion effect dominates the dynamics in comparison with the non-homogeneity driven by the boundaries. The DMD error is mostly distributed near the two boundaries, and this error accumulates with time. On the other hand, xDMD has a flat error distribution in the physical domain with the much smaller error magnitude.

In Case 2, which is a homogeneous case, the accuracy of xDMD than that of DMD. The improvements are mostly due to the modification in rDMD, but also indicate that no sacrifice of accuracy is made by adding the bias. This test guarantees better performance of xDMD without knowledge of homogeneity.

In Case 3, both DMD and xDMD lose several orders of accuracy and behave almost the same in the presence of noise. In the interpolation test, xDMD is even less accurate than DMD. This behavior is reminiscent of the over-fitting issue in machine learning: models with more parameters fit too closely to the limited number of contaminant data and therefore fail to fit additional data or reliably predict future observations. Obviously, xDMD has more parameters than DMD due to the bias term.

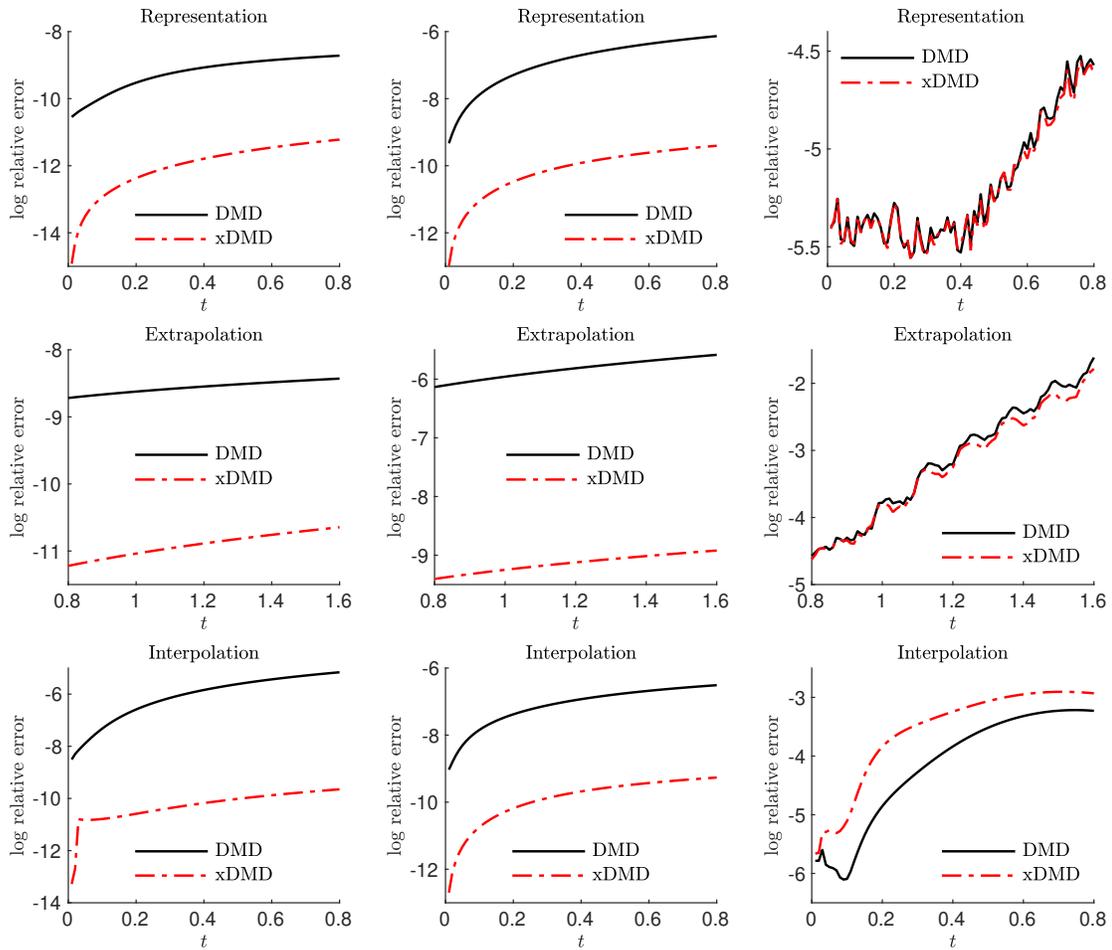


Figure C.1: Solution accuracy in Cases 1 (left column), 2 (middle column) and 3 (right column) in the representation (top row), extrapolation (middle row) and interpolation (bottom row) regimes.

## C.2 Two-dimensional Viscous Burgers' Equation

Consider the two-dimensional viscous Burgers' equation,

$$\begin{cases} \partial_t u + u\partial_x u + v\partial_y u = \nu(\partial_{xx}u + \partial_{yy}u) \\ \partial_t v + u\partial_x v + v\partial_y v = \nu(\partial_{xx}v + \partial_{yy}v). \end{cases} \quad (\text{C.4})$$

This equation, with  $\nu = 0.05$ , is defined for  $(x, y) \in [0, 2] \times [0, 2]$  and  $t \in [0, 2]$ ; and is subject to no-flux boundary conditions and the initial condition

$$u(x, y, 0) = v(x, y, 0) = \begin{cases} 1 & (x, y) \in [0.5, 1] \times [0.5, 1], \\ 0 & \text{otherwise.} \end{cases} \quad (\text{C.5})$$

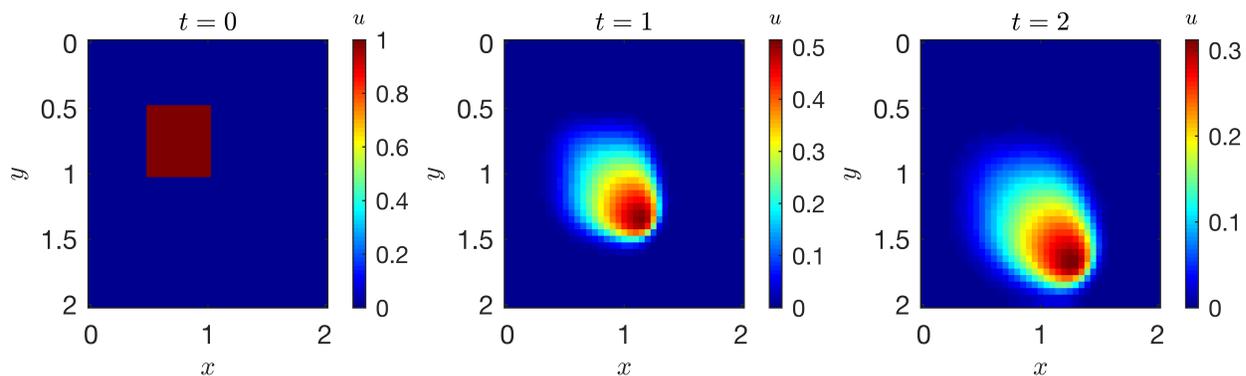


Figure C.2: Temporal snapshots of the solution,  $u(\mathbf{x}, t) = v(\mathbf{x}, t)$ , to the 2D viscous Burgers' equation.

The reference solution is computed via a finite-difference scheme on a uniform mesh with  $\Delta x = \Delta y = 0.05$  and  $\Delta t = 0.001$ . The snapshot solution needs to be reshaped into a vectorized form. We randomly select 500 snapshots out of the 2000 reference solutions to form the training data. Due to the viscosity  $\nu$ , the solution exhibits weak nonlinearity and smooth diffusive profiles (Figure C.2). Both DMD and xDMD capture the solution with satisfactory accuracy. We plot the relative error of xDMD is two orders of magnitude smaller than DMD (Figure C.3).

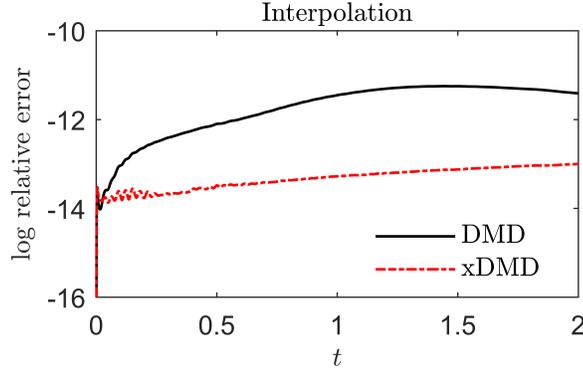


Figure C.3: Temporal evolution of the log relative errors of the DMD and xDMD solutions,  $\mathbf{u}(\mathbf{x}, t) = (u, v)^\top$ , to the viscous Burgers' equation in the interpolation regime.

### C.3 One-dimensional Advection-Diffusion Equation

Consider a one-dimensional advection-diffusion equation with a time-invariant source,

$$\begin{cases} \partial_t u + v \partial_x u = D \partial_{xx} u + S(x), & x \in [-4, 4], & t \in [0, 4], \\ S(x) = \exp(-x^2/0.2). \end{cases} \quad (\text{C.6})$$

We set  $v = 1$  and  $D = 0.1$ . The training is conducted using the initial and boundary conditions

$$\begin{cases} u(x, 0) = \exp(-(x + 2)^2/0.1), \\ u_x(-4, t) = 0, \quad u_x(4, t) = 0. \end{cases} \quad (\text{C.7})$$

The initial condition mimics a localized source at point  $x = -2$  with strength 1 and width  $\sqrt{0.1}$ .

The training data should be carefully chosen such that its traveling wave can cover the whole domain of interest and the training time should be sufficiently long. In this case, one should choose a training dataset with active pulses all over the domain  $[-4, 4]$ . Otherwise, the data-driven modeling will receive no signal in parts of the domain and, thus, fail to learn the global dynamics. This issue has been discussed in [30] for advection-dominant phenomena.

The training data are collected from reference solutions using a finite-difference scheme with  $\Delta x = 0.04$  and  $\Delta t = 0.04$ . Figure C.4 shows that both DMD and xDMD represent the training data with satisfactory accuracy. As in the previous tests, xDMD achieves higher-order accuracy than DMD.

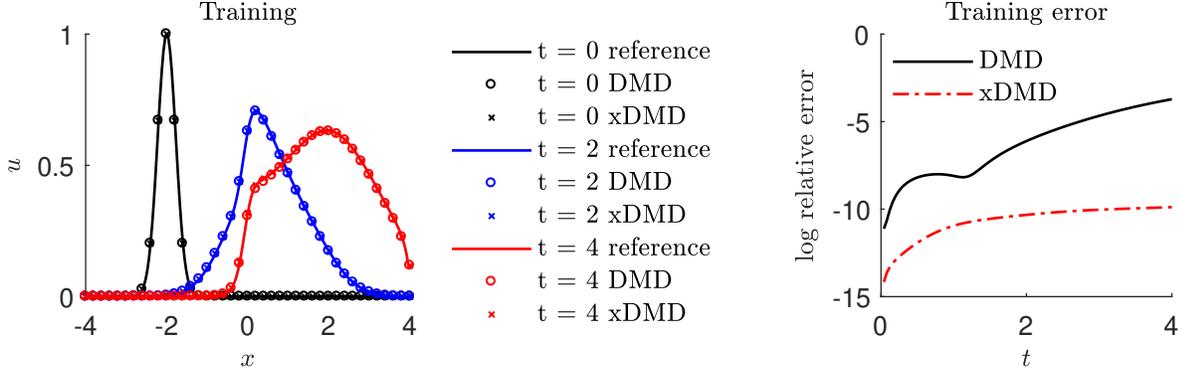


Figure C.4: DMD and xDMD solutions to (C.6) on the training data. Left: the DMD and xDMD solution profiles compared with the reference solution at different times; Right: temporal evolution of the log relative error.

Essentially, we want the DMD and xDMD models to learn, from the training data, the advection-diffusion operator with a fixed source. If the models are accurate, then for a different input (e.g., a point source with different strength, location and width), one can use the DMD and xDMD approximations to obtain solutions directly, without solving the governing equation. We test this generalizability on two types of inputs. In Test 1, the input data are generated for a single point source  $u(x, 0) = s \exp(-(x - x^0)^2/\sigma^2)$ , where  $s \sim \mathcal{U}[1, 11]$ ,  $x^0 \sim \mathcal{U}[-2, 1]$ , and  $\sigma^2 \sim \mathcal{U}[1/15, 1/10]$ . In Test 2, the input data are generated from a two-point source  $u(x, 0) = s_1 \exp(-(x - x_1^0)^2/\sigma_1^2) + s_2 \exp(-(x - x_2^0)^2/\sigma_2^2)$ , where  $s_1, s_2 \sim \mathcal{U}[1, 11]$ ,  $x_1^0, x_2^0 \sim \mathcal{U}[-2, 1]$ , and  $\sigma_1^2, \sigma_2^2 \sim \mathcal{U}[1/15, 1/10]$ .

Figure C.5 shows that xDMD has superior performance in generalizing the learned model to new, previously unseen inputs. The modeling errors in the two tests are well controlled under reasonable magnitude. On the other hand, DMD has poor performance in generalization due to the lack of source term identification. The nature of (C.6) implies that a good model should consist of two parts: one part accounts for the advection-diffusion operator, which is sensitive to the variation of

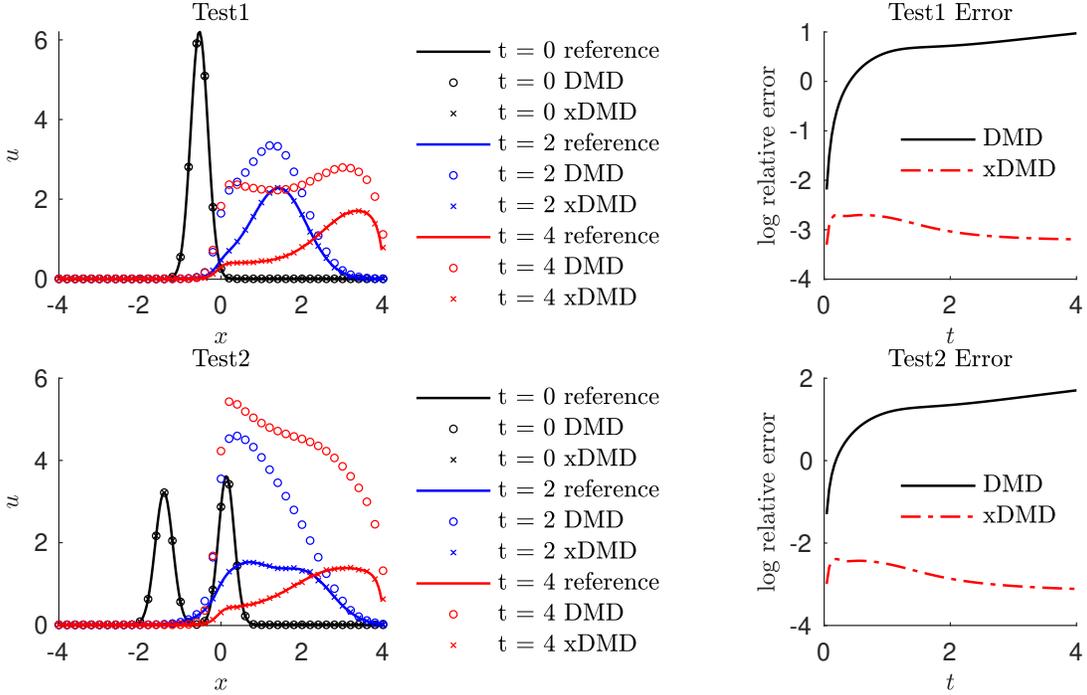


Figure C.5: DMD and xDMD solutions to (C.6) on the test data. Left: the DMD and xDMD solution profiles compared with the reference solution at different times; Right: temporal evolution of the log relative error.

the initial inputs; the other part accounts for the inhomogeneous source term, which is invariant to the initial inputs. This intuition is explicitly accounted for in the xDMD framework.

## C.4 Two-dimensional Advection-Diffusion Equation

Next, we consider a two-dimensional advection-diffusion equation

$$\begin{cases} \partial_t u + \mathbf{v} \cdot \nabla u = \nabla \cdot (\mathbf{D} \nabla u) + S(x, y), & (x, y) \in [0, 20] \times [0, 10], \quad t \in [0, T], \\ S(x, y) = s \exp \left[ -\frac{(x - 5)^2 + (y - 5)^2}{2\sigma^2} \right]. \end{cases} \quad (\text{C.8})$$

We set

$$\mathbf{v} = \begin{pmatrix} -2.75 \\ 0.0 \end{pmatrix}, \quad \mathbf{D} = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix}, \quad s = 100.0, \quad \sigma^2 = 0.05.$$

This equation is subject to no-flux boundary conditions and the initial condition  $u(\mathbf{x}, 0) = u_{\text{in}}(\mathbf{x})$ . The problem describes, e.g., the spatiotemporal evolution of the concentration  $u(\mathbf{x}, t)$  of a groundwater contaminant that is advected by flow velocity  $\mathbf{v}$ , while undergoing hydrodynamic dispersion. The training data are generated for

$$u_{\text{in}}(\mathbf{x}) \equiv s \exp \left[ -\frac{(x - x_s)^2 + (y - y_s)^2}{2\sigma^2} \right], \quad (\text{C.9})$$

wherein the coordinates of the plume's center of mass,  $(x_s, y_s)$  are treated as independent random variables with uniform distributions,  $x_s \sim \mathcal{U}[0, 10]$  and  $y_s \sim \mathcal{U}[0, 10]$ . We generate  $N_{\text{MC}} = 4000$  realizations of the pairs  $(x_s, y_s)$  and evaluate the corresponding initial conditions  $u_{\text{in}}^{(n)}(\mathbf{x})$  for  $n = 1, \dots, N_{\text{MC}}$ . For each of these realizations, (C.8) is solved to compute  $u_T^{(n)}(\mathbf{x}) \equiv u^{(n)}(\mathbf{x}, T)$  with  $T = 4$  using a finite-difference scheme with  $\Delta x = \Delta y = 0.25$ . The matrix pairs  $\{u_{\text{in}}^{(n)}, u_T^{(n)}\}_{n=1}^{N_{\text{MC}}}$  are arranged into data matrices  $\mathbf{X}_1$  and  $\mathbf{X}_2$ , as in (1.18). Finally, the DMD and xDMD models are deployed to learn the flow map  $\mathcal{F}_{\Delta t}$  with the time lag  $\Delta t = T$ .

We test the generalizability of the DMD and xDMD models by considering the following three tests.

- Test 1: The initial input is the same as in (C.9) but the single point source is now allowed to have different strength and width:  $s \sim \mathcal{U}(50, 100)$  and  $\sigma^2 \sim \mathcal{U}(0.02, 0.1)$ .
- Test 2: The initial input is a two-point source with different strengths, locations and widths:

$$u_{\text{in}} = s_1 \exp \left[ -\frac{(x - x_{s_1})^2 + (y - y_{s_1})^2}{2\sigma_1^2} \right] + s_2 \exp \left[ -\frac{(x - x_{s_2})^2 + (y - y_{s_2})^2}{2\sigma_2^2} \right], \quad (\text{C.10})$$

where  $s_1, s_2 = \mathcal{U}(50, 100)$ ,  $\sigma_1^2, \sigma_2^2 = \mathcal{U}(0.02, 0.1)$ ,  $x_{s_1}, x_{s_2} \sim \mathcal{U}[0, 10]$ , and  $y_{s_1}, y_{s_2} \sim \mathcal{U}[0, 10]$ .

- Test 3: The initial input is a fixed-strength line source,

$$u_{\text{in}} = \begin{cases} 75 & x = 5, \quad y \in [3, 6], \\ 0 & \text{otherwise.} \end{cases} \quad (\text{C.11})$$

For Test 1, Figure C.6 demonstrates the xDMD model's ability to accurately predict the solution for an initial condition not represented in the training set. At the same time, the DMD model fails this relatively weak generalization test due to the reason given in section C.3. The DMD error map has a peak centered at  $(5, 5)$ , which is the location of the source  $S(\mathbf{x})$  in (C.8). This further verifies that the loss of accuracy is caused by the shortcoming of DMD in identifying the inhomogeneous source term.

Figure C.7 reveals a similar performance of DMD and xDMD in the more challenging Test 2. As before, the xDMD model accurately predicts the solution at  $T = 4$  corresponding to the two-point initial input not seen during the training. The right corner concentration tail is mostly caused by the advection-diffusion effect on the north-east point source. This pure advection-diffusion dynamic is well captured by DMD, as shown by the flat low error concentration in the DMD error map. The error peak is at  $(5, 5)$  again, showing the significant effect of identifying the source.

Figure C.8 demonstrates a similar behavior of the DMD and xDMD solutions for a line source initial input (Test 3). Although the solutions to (C.8) with the single-point source, two-point source and line source exhibit quite different features, all of them can be thought of as a linear superposition of the training single-point sources. Therefore, all the three types of the initial input can be regarded as drawn from the same distribution. As before, xDMD again achieves satisfactory accuracy in this generalizability test and DMD appears similar error map pattern centered at  $(5, 5)$ .

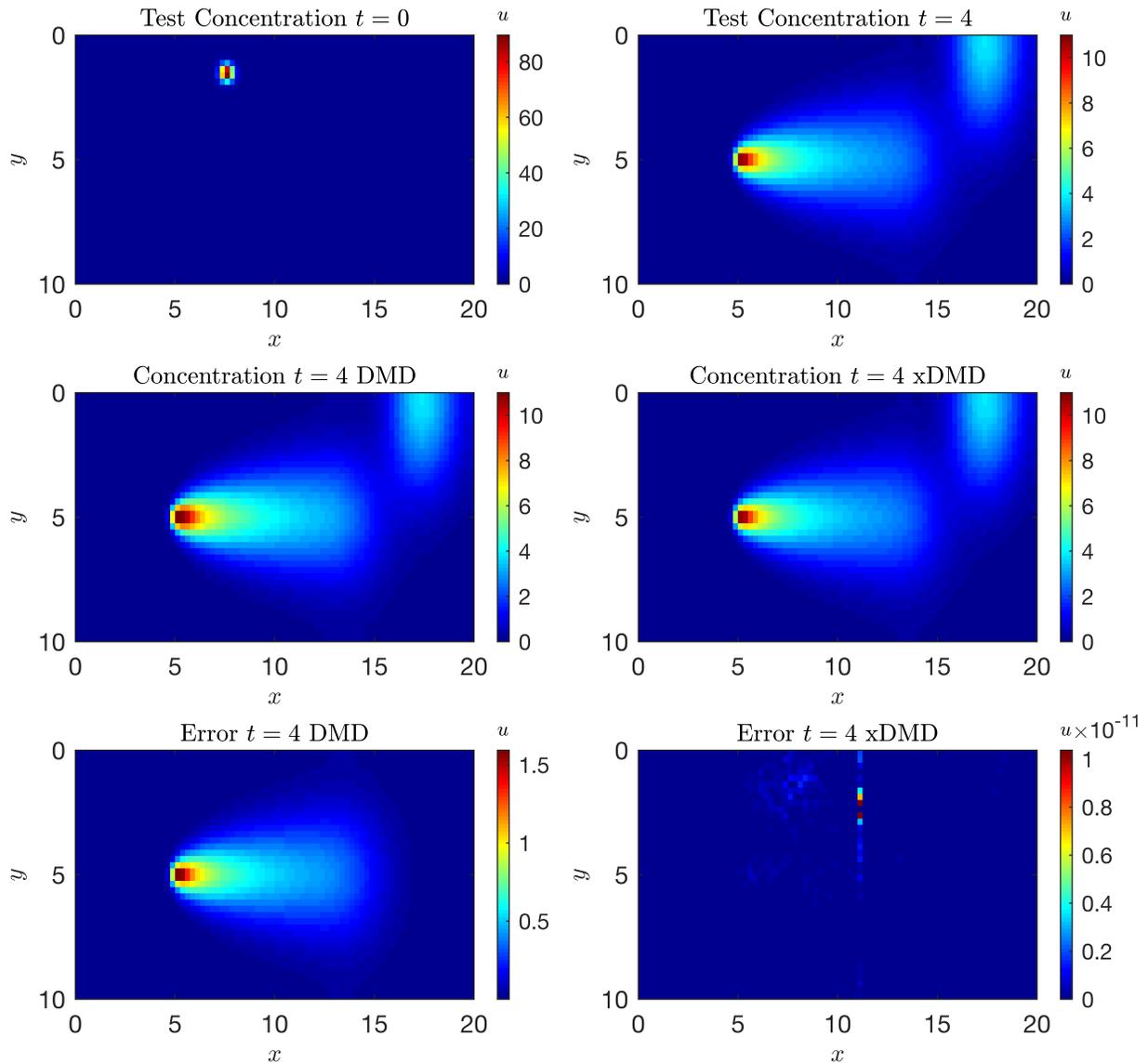


Figure C.6: Test 1: The ground truth at the end of simulation time (top row) and its DMD and xDMD approximations (middle row), accompanied by the corresponding absolute error maps (bottom row).

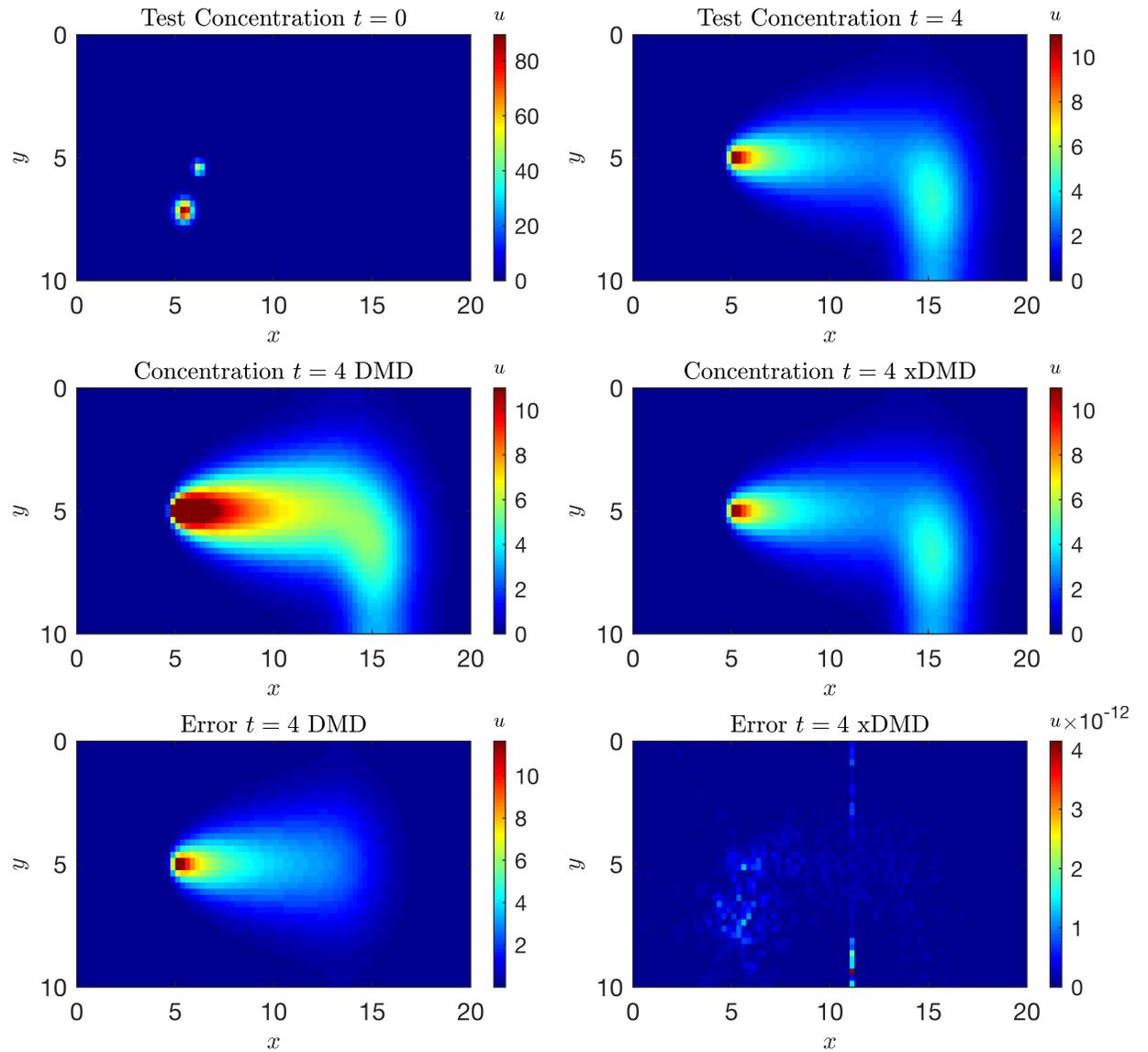


Figure C.7: Test 2: The ground truth at the end of simulation time (top row) and its DMD and xDMD approximations (middle row), accompanied by the corresponding absolute error maps (bottom row).

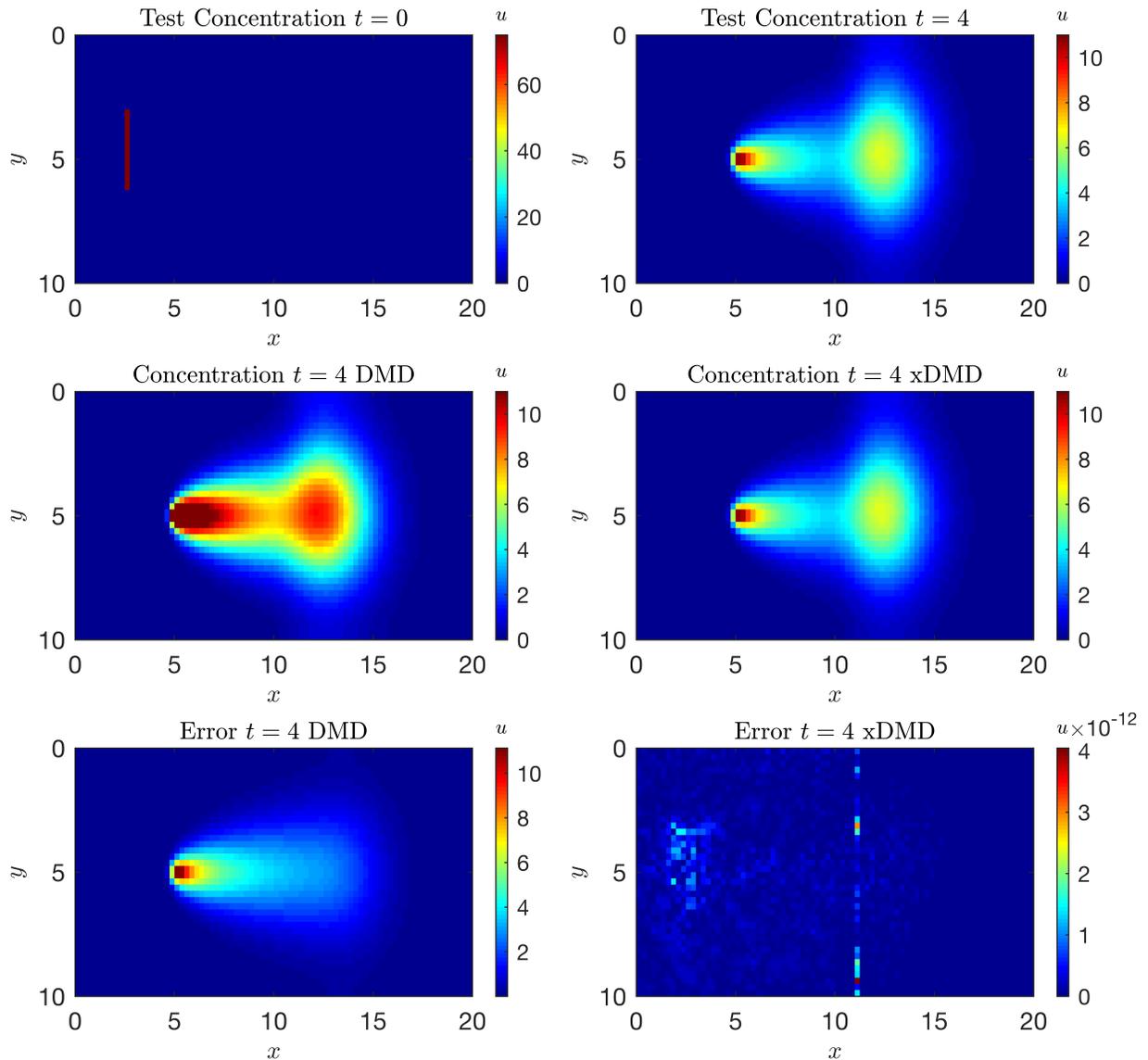


Figure C.8: Test 3: The ground truth at the end of simulation time (top row) and its DMD and xDMD approximations (middle row), accompanied by the corresponding absolute error maps (bottom row).

## C.5 Transport in Heterogeneous Media: Generalizability to New Inputs

The setting is identical to that in Section 5.3.4. Our goal here is to test the ability of DMD and xDMD models to predict  $u(\mathbf{x}, T)$  for initial conditions not seen in training, such as a two-point source with different strength and locations:

$$u_{\text{in}}(\mathbf{x}) = s_1 \exp(-(x - x_{s_1})^2 + (y - y_{s_1})^2) + s_2 \exp(-(x - x_{s_2})^2 + (y - y_{s_2})^2), \quad (\text{C.12})$$

where  $s_1 = 50$ ,  $s_2 = 80$ ,  $(x_{s_1}, y_{s_1}) = (10, 40)$ ,  $(x_{s_2}, y_{s_2}) = (20, 20)$ .

Figure C.9 shows the success of xDMD in learning the solution for an initial data not seen the training data. The xDMD error map has very small magnitude, indicating the high accuracy of xDMD in this generalized test. On the other hand, DMD predicts a very different concentration map, failing the generalization test. As before, the DMD error is highest close to the left boundary, where the Dirichlet boundary condition is imposed. This visualization again addresses the significant role of the bias term added in the new xDMD framework.

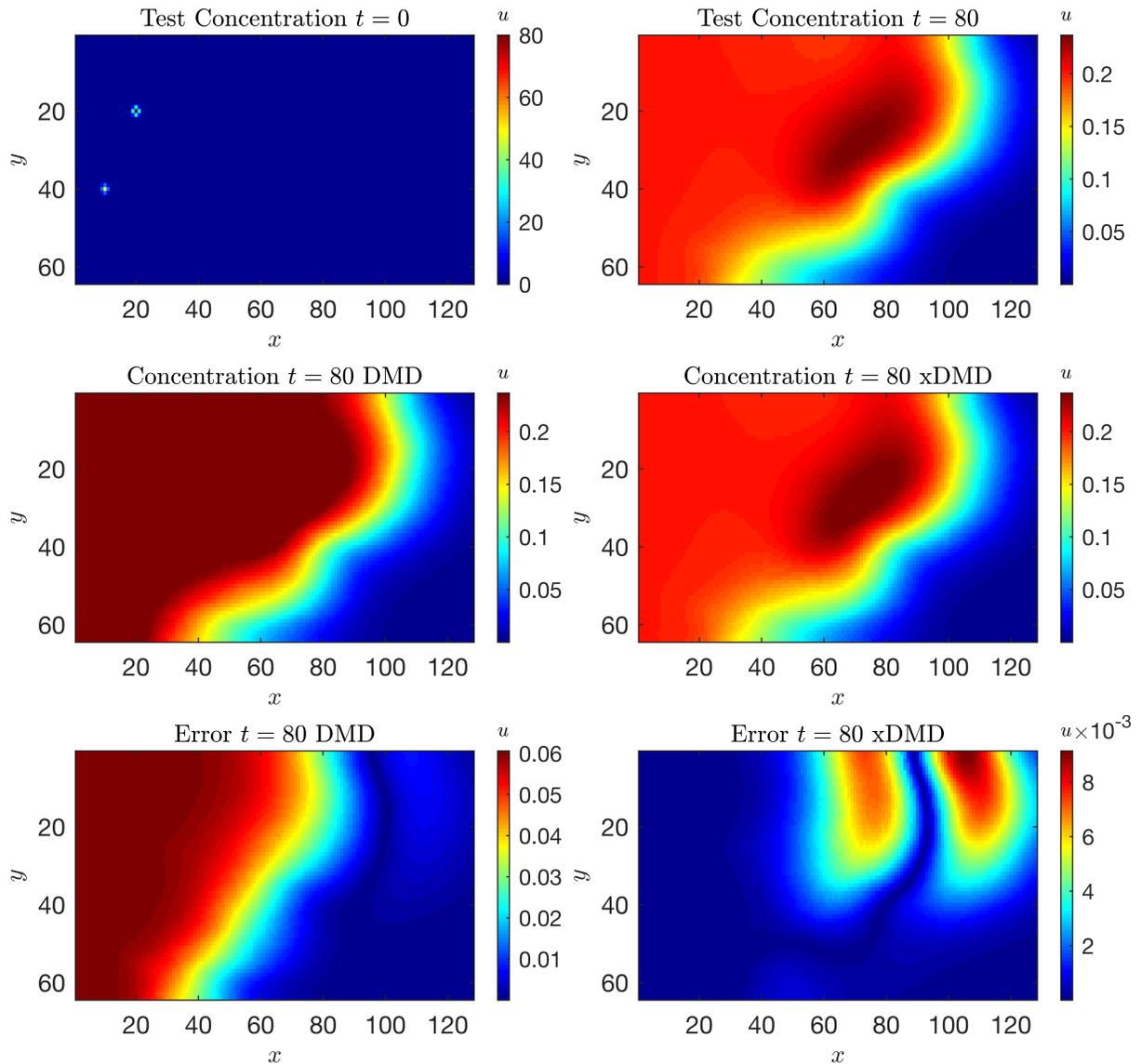


Figure C.9: Section 4.4: The ground truth at the end of simulation time (top row) and its DMD and xDMD approximations (middle row), accompanied by the corresponding absolute error maps (bottom row).

# Bibliography

- [1] Steven L Brunton, Bingni W Brunton, Joshua L Proctor, and J Nathan Kutz. Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control. *PloS one*, 11(2):e0150171, 2016.
- [2] Hannah Lu and Daniel M Tartakovsky. Prediction accuracy of dynamic mode decomposition. *SIAM Journal on Scientific Computing*, 42(3):A1639–A1662, 2020.
- [3] Marco A Cardoso and Louis J Durlofsky. Linearized reduced-order models for subsurface flow simulation. *Journal of Computational Physics*, 229(3):681–700, 2010.
- [4] Jincong He, Jonsat Sætrom, and Louis J Durlofsky. Enhanced linearized reduced-order models for subsurface flow simulation. *Journal of Computational Physics*, 230(23):8313–8341, 2011.
- [5] Jincong He and Louis J Durlofsky. Reduced-order modeling for compositional simulation by use of trajectory piecewise linearization. *SPE Journal*, 19(05):858–872, 2014.
- [6] Zhaoyang Larry Jin and Louis J Durlofsky. Reduced-order modeling of CO<sub>2</sub> storage operations. *International Journal of Greenhouse Gas Control*, 68:49–67, 2018.
- [7] Pascal Audigane, Irina Gaus, Isabelle Czernichowski-Lauriol, Karsten Pruess, and Tianfu Xu. Two-dimensional reactive transport modeling of CO<sub>2</sub> injection

- in a saline aquifer at the Sleipner site, North Sea. *American Journal of Science*, 307(7):974–1008, 2007.
- [8] Yi Xiong, Perapon Fakcharoenphol, Philip Winterfeld, Ronglei Zhang, Yu-Shu Wu, et al. Coupled geomechanical and reactive geochemical model for fluid and heat flow: application for enhanced geothermal reservoir. In *SPE Reservoir Characterization and Simulation Conference and Exhibition*. Society of Petroleum Engineers, 2013.
- [9] Sergio A Bea, Haruko Wainwright, Nicolas Spycher, Boris Faybishenko, Susan S Hubbard, and Miles E Denham. Identifying key controls on the behavior of an acidic-U (VI) plume in the Savannah River Site using reactive transport modeling. *Journal of Contaminant Hydrology*, 151:34–54, 2013.
- [10] Hannah Lu, Dinara Ermakova, Haruko Murakami Wainwright, Liange Zheng, and Daniel M Tartakovsky. Data-informed emulators for multi-physics simulations. *Journal of Machine Learning for Modeling and Computing*, 2(2), 2021.
- [11] Carl I Steefel et al. Reactive transport codes for subsurface environmental simulation. *Computational Geosciences*, 19(3):445–478, 2015.
- [12] L A Schmit Jr and B Farshi. Some approximation concepts for structural synthesis. *AIAA Journal*, 12(5):692–699, 1974.
- [13] J-F M Barthelemy and Raphael T Haftka. Approximation concepts for optimum structural design—a review. *Structural Optimization*, 5(3):129–144, 1993.
- [14] Timothy W Simpson, JD Poplinski, Patrick N Koch, and Janet K Allen. Meta-models for computer-based engineering design: survey and recommendations. *Engineering with Computers*, 17(2):129–150, 2001.
- [15] David J Lucia, Philip S Beran, and Walter A Silva. Reduced-order modeling: new approaches for computational physics. *Progress in Aerospace Sciences*, 40(1-2):51–117, 2004.

- [16] Kostas M Saridakis and Argiris J Dentsoras. Soft computing in engineering design—A review. *Advanced Engineering Informatics*, 22(2):202–221, 2008.
- [17] Alexander I J Forrester and Andy J Keane. Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences*, 45(1-3):50–79, 2009.
- [18] Saman Razavi, Bryan A Tolson, and Donald H Burn. Review of surrogate modeling in water resources. *Water Resources Research*, 48(7), 2012.
- [19] Gaetan Kerschen, Jean-claude Golinval, Alexander F Vakakis, and Lawrence A Bergman. The method of proper orthogonal decomposition for dynamical characterization and order reduction of mechanical systems: an overview. *Nonlinear Dynamics*, 41(1-3):147–169, 2005.
- [20] Clarence W Rowley. Model reduction for fluids, using balanced proper orthogonal decomposition. *International Journal of Bifurcation and Chaos*, 15(03):997–1013, 2005.
- [21] Peter Benner, Serkan Gugercin, and Karen Willcox. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM review*, 57(4):483–531, 2015.
- [22] Peter J Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, 656:5–28, 2010.
- [23] J Nathan Kutz, Steven L Brunton, Bingni W Brunton, and Joshua L Proctor. *Dynamic mode decomposition: data-driven modeling of complex systems*. SIAM, 2016.
- [24] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer School on Machine Learning*, pages 63–71. Springer, 2003.
- [25] George Shu Heng Pau, Yingqi Zhang, and Stefan Finsterle. Reduced order models for many-query subsurface flow applications. *Computational Geosciences*, 17(4):705–721, 2013.

- [26] DJ Booker and RA Woods. Comparing and combining physically-based and empirically-based approaches for estimating the hydrology of ungauged catchments. *Journal of Hydrology*, 508:227–239, 2014.
- [27] Seyed Amir Naghibi, Hamid Reza Pourghasemi, and Barnali Dixon. GIS-based groundwater potential mapping using boosted regression tree, classification and regression tree, and random forest machine learning models in Iran. *Environmental Monitoring and Assessment*, 188(1):44, 2016.
- [28] Jan S Hesthaven and Stefano Ubbiali. Non-intrusive reduced order modeling of nonlinear problems using neural networks. *Journal of Computational Physics*, 363:55–78, 2018.
- [29] Tong Qin, Kailiang Wu, and Dongbin Xiu. Data driven governing equations approximation using deep neural networks. *Journal of Computational Physics*, 395:620–635, 2019.
- [30] Hannah Lu and Daniel M Tartakovsky. Lagrangian dynamic mode decomposition for construction of reduced-order models of advection-dominated phenomena. *Journal of Computational Physics*, 407:109229, 2020.
- [31] Hannah Lu and Daniel M Tartakovsky. Dynamic mode decomposition for construction of reduced-order models of hyperbolic problems with shocks. *Journal of Machine Learning for Modeling and Computing*, 2(1), 2021.
- [32] Hannah Lu and Daniel M Tartakovsky. Extended dynamic mode decomposition for inhomogeneous problems. *Journal of Computational Physics*, 444:110550, 2021.
- [33] Shady E Ahmed, Omer San, Diana A Bistrrian, and Ionel M Navon. Sampling and resolution characteristics in reduced order models of shallow water equations: Intrusive vs nonintrusive. *International Journal for Numerical Methods in Fluids*, 92(8):992–1036, 2020.

- [34] Giorgos Mamakoukas, Maria L Castano, Xiaobo Tan, and Todd D Murphey. Derivative-based koopman operators for real-time control of robotic systems. *IEEE Transactions on Robotics*, 37(6):2173–2192, 2021.
- [35] Nassir Cassamo and Jan-Willem van Wingerden. On the potential of reduced order models for wind farm control: a koopman dynamic mode decomposition approach. *Energies*, 13(24):6513, 2020.
- [36] Milad Habibi, Scott Dawson, and Amirhossein Arzani. Data-driven pulsatile blood flow physics with dynamic mode decomposition. *Fluids*, 5(3):111, 2020.
- [37] Saifon Chaturantabut and Danny C Sorensen. Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing*, 32(5):2737–2764, 2010.
- [38] Maxime Barrault, Yvon Maday, Ngoc Cuong Nguyen, and Anthony T Patera. An ‘empirical interpolation’ method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathematique*, 339(9):667–672, 2004.
- [39] Bernard O Koopman. Hamiltonian systems and transformation in hilbert space. *Proceedings of the national academy of sciences of the united states of america*, 17(5):315, 1931.
- [40] I. Mezić. Spectral properties of dynamical systems, model reduction and decompositions. *Nonlin. Dyn.*, 41(1-3):309–325, 2005.
- [41] I. Mezić. Analysis of fluid flows via spectral properties of the Koopman operator. *Annu. Rev. Fluid Mech.*, 45:357–378, 2013.
- [42] I. Mezić and A. Banaszuk. Comparison of systems with complex behavior. *Physica D*, 197(1-2):101–133, 2004.
- [43] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. S. Henningson. Spectral analysis of nonlinear flows. *J. Fluid Mech.*, 641:115–127, 2009.

- [44] S. L. Brunton, J. L. Proctor, and J. N. Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proc. Natl. Acad. Sci. U.S.A.*, 113(15):3932–3937, 2016.
- [45] M. Schmidt and H. Lipson. Distilling free-form natural laws from experimental data. *Science*, 324(5923):81–85, 2009.
- [46] AS Gurvich, BM Koprov, LP Tsvang, NL Byzova, and IM Bobyleva. Atmospheric turbulence and radio wave propagation (selected articles). Technical report, Foreign Technology Div Wright-Patterson AFB OHIO, 1969.
- [47] Denis V Voskov and Hamdi A Tchelepi. Compositional space parameterization: theory and application for immiscible displacements. *SPE Journal*, 14(03):431–440, 2009.
- [48] N Castelletto, Hadi Hajibeygi, and HA Tchelepi. Hybrid multiscale formulation for coupled flow and geomechanics. In *ECMOR XV-15th European Conference on the Mathematics of Oil Recovery*, pages cp–494. European Association of Geoscientists & Engineers, 2016.
- [49] Gene Golub, Virginia Klema, and Gilbert W Stewart. Rank degeneracy and least squares problems. Technical report, Stanford University, Department of Computer Science, Stanford, CA, 1976.
- [50] Fangxin Fang, Christopher C Pain, IM Navon, Ahmed H Elsheikh, Juan Du, and Dunhui Xiao. Non-linear petrov–galerkin methods for reduced order hyperbolic equations and discontinuous finite element methods. *Journal of Computational Physics*, 234:540–559, 2013.
- [51] Kevin Carlberg, Matthew Barone, and Harbir Antil. Galerkin v. least-squares petrov–galerkin projection in nonlinear model reduction. *Journal of Computational Physics*, 330:693–734, 2017.
- [52] Jincong He and Louis J Durlofsky. Constraint reduction procedures for reduced-order subsurface flow models based on pod–tpwl. *International Journal for Numerical Methods in Engineering*, 103(1):1–30, 2015.

- [53] Matthieu AH Rousset, Chung K Huang, Hector Klie, and Louis J Durlofsky. Reduced-order modeling for thermal recovery processes. *Computational Geosciences*, 18(3):401–415, 2014.
- [54] Kevin Carlberg, Charbel Bou-Mosleh, and Charbel Farhat. Efficient non-linear model reduction via a least-squares petrov–galerkin projection and compressive tensor approximations. *International Journal for numerical methods in engineering*, 86(2):155–181, 2011.
- [55] Eduardo Gildin, Mohammadreza Ghasemi, Anastasiya Romanovskay, and Yalchin Efendiev. Nonlinear complexity reduction for fast simulation of flow in heterogeneous porous media. In *SPE Reservoir Simulation Symposium*. OnePetro, 2013.
- [56] Mehdi Ghommem, Eduardo Gildin, and Mohammadreza Ghasemi. Complexity reduction of multiphase flows in heterogeneous porous media. *SPE Journal*, 21(01):144–151, 2016.
- [57] Rui Jiang and Louis J Durlofsky. Implementation and detailed assessment of a gnat reduced-order model for subsurface flow simulation. *Journal of Computational Physics*, 379:192–213, 2019.
- [58] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor. *Dynamic mode decomposition: data-driven modeling of complex systems*, volume 149. SIAM, Philadelphia, PA, 2016.
- [59] B. W. Brunton, L. A. Johnson, J. G. Ojemann, and J. N. Kutz. Extracting spatial-temporal coherent patterns in large-scale neural recordings using dynamic mode decomposition. *J. Neurosci. Meth.*, 258:1–15, 2016.
- [60] J. Mann and J. N. Kutz. Dynamic mode decomposition for financial trading strategies. *Quant. Finance*, 16(11):1643–1655, 2016.
- [61] J. L. Proctor and P. A. Eckhoff. Discovering dynamic patterns from infectious disease data using dynamic mode decomposition. *Int. Health*, 7(2):139–145, 2015.

- [62] J. L. Proctor, S. L. Brunton, and J. N. Kutz. Dynamic mode decomposition with control. *SIAM J. Appl. Dyn. Syst.*, 15(1):142–161, 2016.
- [63] Q. Li, F. Dietrich, E. M. Bollt, and I. G. Kevrekidis. Extended dynamic mode decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the Koopman operator. *Chaos*, 27(10):103111, 2017.
- [64] S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz. Data-driven discovery of partial differential equations. *Science Adv.*, 3(4):e1602614, 2017.
- [65] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley. A data-driven approximation of the Koopman operator: Extending dynamic mode decomposition. *J. Nonlinear Sci.*, 25(6):1307–1346, 2015.
- [66] I. G. Kevrekidis, C. W. Gear, J. M. Hyman, P. G. Kevrekidid, O. Runborg, C. Theodoropoulos, et al. Equation-free, coarse-grained multiscale computation: Enabling microscopic simulators to perform system-level analysis. *Comm. Math. Sci.*, 1(4):715–762, 2003.
- [67] W.-X. Wang, R. Yang, Y.-C. Lai, V. Kovanis, and C. Grebogi. Predicting catastrophes in nonlinear dynamical systems by compressive sensing. *Phys. Rev. Lett.*, 106(15):154101, 2011.
- [68] S. T. M. Dawson, M. S. Hemati, M. O. Williams, and C. W. Rowley. Characterizing and correcting for the effect of sensor noise in the dynamic mode decomposition. *Exper. Fluids*, 57(3):42, 2016.
- [69] Z. Drmac, I. Mezić, and R. Mohr. Data driven modal decompositions: analysis and enhancements. *SIAM J. Sci. Comput.*, 40(4):A2253–A2285, 2018.
- [70] S. L. Brunton, J. L. Proctor, J. H. Tu, and J. N. Kutz. Compressed sensing and dynamic mode decomposition. *J. Comput. Dyn.*, 2(165):2158–2491, 2015.
- [71] J. H. Tu, C. W. Rowley, J. N. Kutz, and J. K. Shang. Spectral analysis of fluid flows using sub-Nyquist-rate PIV data. *Exper. Fluids*, 55(9):1805, 2014.

- [72] J. N. Kutz, X. Fu, and S. L. Brunton. Multiresolution dynamic mode decomposition. *SIAM J. Appl. Dyn. Syst.*, 15(2):713–735, 2016.
- [73] C. W. Rowley and J. E. Marsden. Reconstruction equations and the Karhunen-Loève expansion for systems with symmetry. *Physica D*, 142(1-2):1–19, 2000.
- [74] William E Boyce, Richard C DiPrima, and Douglas B Meade. *Elementary differential equations and boundary value problems*. John Wiley & Sons, 2021.
- [75] Jonathan H Tu. *Dynamic mode decomposition: Theory and applications*. PhD thesis, Princeton University, 2013.
- [76] Philip Holmes, John L Lumley, Gahl Berkooz, and Clarence W Rowley. *Turbulence, coherent structures, dynamical systems and symmetry*. Cambridge university press, 2012.
- [77] Mario Ohlberger and Stephan Rave. Nonlinear reduced basis approximation of parameterized evolution equations via the method of freezing. *Comptes Rendus Mathématique*, 351(23-24):901–906, 2013.
- [78] Wolfgang Dahmen, Christian Plesken, and Gerrit Welper. Double greedy algorithms: Reduced basis methods for transport dominated problems. *ESAIM: Mathematical Modelling and Numerical Analysis*, 48(3):623–663, 2014.
- [79] Tommaso Taddei, Simona Perotto, and Alfio Quarteroni. Reduced basis techniques for nonlinear conservation laws. *ESAIM: Mathematical Modelling and Numerical Analysis*, 49(3):787–814, 2015.
- [80] Angelo Iollo and Damiano Lombardi. Advection modes by optimal mass transfer. *Physical Review E*, 89(2):022923, 2014.
- [81] Julius Reiss, Philipp Schulze, Jörn Sesterhenn, and Volker Mehrmann. The shifted proper orthogonal decomposition: A mode decomposition for multiple transport phenomena. *SIAM Journal on Scientific Computing*, 40(3):A1322–A1344, 2018.

- [82] Constantin Greif and Karsten Urban. Decay of the kolmogorov n-width for wave problems. *Applied Mathematics Letters*, 96:216–222, 2019.
- [83] Zhu Wang, Imran Akhtar, Jeff Borggaard, and Traian Iliescu. Proper orthogonal decomposition closure models for turbulent flows: a numerical comparison. *Computer Methods in Applied Mechanics and Engineering*, 237:10–26, 2012.
- [84] Shaowu Pan and Karthik Duraisamy. Data-driven discovery of closure models. *SIAM Journal on Applied Dynamical Systems*, 17(4):2381–2413, 2018.
- [85] Eric J Parish and Karthik Duraisamy. A paradigm for data-driven predictive modeling using field inversion and machine learning. *Journal of Computational Physics*, 305:758–774, 2016.
- [86] Angelo Iollo, Stéphane Lanteri, and J-A Désidéri. Stability properties of pod–galerkin approximations for the compressible navier–stokes equations. *Theoretical and Computational Fluid Dynamics*, 13(6):377–396, 2000.
- [87] Clarence W Rowley, Tim Colonius, and Richard M Murray. Model reduction for compressible flows using pod and galerkin projection. *Physica D: Nonlinear Phenomena*, 189(1-2):115–129, 2004.
- [88] Randall J LeVeque, Knut Waagan, Frank I González, Donsub Rim, and Guang Lin. Generating random earthquake events for probabilistic tsunami hazard assessment. In *Global Tsunami Science: Past and Future, Volume I*, pages 3671–3692. Springer, 2016.
- [89] Donsub Rim and Kyle T Mandli. Displacement interpolation using monotone rearrangement. *SIAM/ASA Journal on Uncertainty Quantification*, 6(4):1503–1531, 2018.
- [90] Gerrit Welper. Interpolation of functions with parameter dependent jumps by transformed snapshots. *SIAM Journal on Scientific Computing*, 39(4):A1225–A1250, 2017.

- [91] Benjamin Peherstorfer and Karen Willcox. Online adaptive model reduction for nonlinear systems via low-rank updates. *SIAM Journal on Scientific Computing*, 37(4):A2123–A2150, 2015.
- [92] Ralf Zimmermann, Benjamin Peherstorfer, and Karen Willcox. Geometric subspace updates with applications to online adaptive nonlinear model reduction. *SIAM Journal on Matrix Analysis and Applications*, 39(1):234–261, 2018.
- [93] Othmar Koch and Christian Lubich. Dynamical low-rank approximation. *SIAM Journal on Matrix Analysis and Applications*, 29(2):434–454, 2007.
- [94] Themistoklis P Sapsis and Pierre FJ Lermusiaux. Dynamically orthogonal field equations for continuous stochastic dynamical systems. *Physica D: Nonlinear Phenomena*, 238(23-24):2347–2360, 2009.
- [95] Eleonora Musharbash, Fabio Nobile, and Tao Zhou. Error analysis of the dynamically orthogonal approximation of time dependent random pdes. *SIAM Journal on Scientific Computing*, 37(2):A776–A810, 2015.
- [96] Eleonora Musharbash and Fabio Nobile. Dual dynamically orthogonal approximation of incompressible navier stokes equations with random boundary conditions. *Journal of Computational Physics*, 354:135–162, 2018.
- [97] Benjamin Peherstorfer. Model reduction for transport-dominated problems via online adaptive bases and adaptive sampling. *SIAM Journal on Scientific Computing*, 42(5):A2803–A2836, 2020.
- [98] Marzieh Alireza Mirhoseini and Matthew J Zahr. Model reduction of convection-dominated partial differential equations via optimization-based implicit feature tracking. *arXiv preprint arXiv:2109.14694*, 2021.
- [99] Davide Papapicco, Nicola Demo, Michele Girfoglio, Giovanni Stabile, and Gianluigi Rozza. The neural network shifted-proper orthogonal decomposition: A machine learning approach for non-linear reduction of hyperbolic equations. *Computer Methods in Applied Mechanics and Engineering*, 392:114687, 2022.

- [100] Ariana Mendible, Steven L Brunton, Aleksandr Y Aravkin, Wes Lowrie, and J Nathan Kutz. Dimensionality reduction and reduced-order modeling for traveling wave physics. *Theoretical and Computational Fluid Dynamics*, 34(4):385–400, 2020.
- [101] Opal Issan and Boris Kramer. Predicting solar wind streams from the inner-heliosphere to earth via shifted operator inference. *arXiv preprint arXiv:2203.13372*, 2022.
- [102] Joshua B Tenenbaum, Vin de Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.
- [103] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.
- [104] David L Donoho and Carrie Grimes. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Sciences*, 100(10):5591–5596, 2003.
- [105] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.
- [106] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.
- [107] Teuvo Kohonen. Self-organized formation of topologically correct feature maps. *Biological cybernetics*, 43(1):59–69, 1982.
- [108] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.
- [109] Kookjin Lee and Kevin T Carlberg. Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. *Journal of Computational Physics*, 404:108973, 2020.

- [110] Zhen Chen and Dongbin Xiu. On generalized residual network for deep learning of unknown dynamical systems. *Journal of Computational Physics*, 438:110362, 2021.
- [111] Rambod Mojjani and Maciej Balajewicz. Lagrangian basis method for dimensionality reduction of convection dominated nonlinear flows. *arXiv preprint arXiv:1701.04343*, 2017.
- [112] Alessandro Alla and J Nathan Kutz. Nonlinear model order reduction via dynamic mode decomposition. *SIAM J. Sci. Comput.*, 39(5):B778–B796, 2017.
- [113] Matthew O Williams, Peter J Schmid, and J Nathan Kutz. Hybrid reduced-order integration with proper orthogonal decomposition and dynamic mode decomposition. *Multiscale Model. Simul.*, 11(2):522–544, 2013.
- [114] D. Duke, J. Soria, and D. Honnery. An error analysis of the dynamic mode decomposition. *Exper. Fluids*, 52(2):529–542, 2012.
- [115] M. Korda and I. Mezić. On convergence of extended dynamic mode decomposition to the Koopman operator. *J. Nonlinear Sci.*, 28(2):687–710, 2018.
- [116] Hassan Arbabi and Igor Mezic. Ergodic theory, dynamic mode decomposition, and computation of spectral properties of the koopman operator. *SIAM Journal on Applied Dynamical Systems*, 16(4):2096–2126, 2017.
- [117] Daniel M Tartakovsky, Alberto Guadagnini, and Monica Riva. Stochastic averaging of nonlinear flows in heterogeneous porous media. *Journal of Fluid Mechanics*, 492:47–62, 2003.
- [118] Daniel M Tartakovsky, Shlomo P Neuman, and Zhiming Lu. Conditional stochastic averaging of steady state unsaturated flow by means of kirchhoff transformation. *Water Resources Research*, 35(3):731–745, 1999.
- [119] D. M. Tartakovsky and M. Dentz. Diffusion in porous media: Phenomena and mechanisms. *Transp. Porous Media*, 130(1):105–127, 2019.

- [120] A. Quarteroni, G. Rozza, et al. *Reduced order methods for modeling and computational reduction*, volume 9. Springer, 2014.
- [121] Swagato Acharjee and Nicholas Zabarlas. A concurrent model reduction approach on spatial and random domains for the solution of stochastic PDEs. *Int. J. Num. Meth. Engrg.*, 66(12):1934–1954, 2006.
- [122] D. Amsallem, M. J. Zahr, and C. Farhat. Nonlinear model order reduction based on local reduced-order bases. *Int. J. Num. Meth. Engrg.*, 92(10):891–916, 2012.
- [123] D. J. Lucia. Reduced order modeling for high speed flows with moving shocks. Technical report, Air Force Inst. of Tech., Wright-Patterson Air Force Base, OH, 2001.
- [124] K. Carlberg. Adaptive h-refinement for reduced-order models. *Int. J. Num. Meth. Engrg.*, 102(5):1192–1210, 2015.
- [125] J.-F. Gerbeau and D. Lombardi. Approximated Lax pairs for the reduced order integration of nonlinear evolution equations. *J. Comput. Phys.*, 265:246–269, 2014.
- [126] M. E. Kavousanakis, R. Erban, A. G. Boudouvis, C. W. Gear, and I. G. Kevrekidis. Projective and coarse projective integration for problems with continuous symmetries. *J. Comput. Phys.*, 225(1):382–407, 2007.
- [127] J. L. Lumley. *Stochastic tools in turbulence*. Courier Corporation, 2007.
- [128] M.-L. Rapún and J. M. Vega. Reduced order models based on local POD plus Galerkin projection. *J. Comput. Phys.*, 229(8):3046–3063, 2010.
- [129] C. W. Rowley, I. G. Kevrekidis, J. E. Marsden, and K. Lust. Reduction and reconstruction for self-similar dynamical systems. *Nonlinearity*, 16(4):1257, 2003.
- [130] D. M. Tartakovsky, A. Guadagnini, and M. Riva. Stochastic averaging of nonlinear flows in heterogeneous porous media. *J. Fluid Mech.*, 492:47–62, 2003.

- [131] M. Dentz, D. M. Tartakovsky, E. Abarca, A. Guadagnini, X. Sanchez-Vila, and J. Carrera. Variable density flow in porous media. *J. Fluid Mech.*, 561:209–235, 2006.
- [132] J. Morton, A. Jameson, M. J. Kochenderfer, and F. Witherden. Deep dynamical modeling and control of unsteady fluid flows. In *Adv. Neural Inf. Process Syst.*, pages 9278–9288, 2018.
- [133] Jean-Luc Thiffeault. Advection-diffusion in Lagrangian coordinates. *Phys. Lett. A*, 309(5-6):415–422, 2003.
- [134] Mikhail Shashkov and Burton Wendroff. A composite scheme for gas dynamics in Lagrangian coordinates. *J. Comput. Phys.*, 150(2):502–517, 1999.
- [135] Simone E Hieber and Petros Koumoutsakos. A Lagrangian particle level set method. *J. Comput. Phys.*, 210(1):342–367, 2005.
- [136] Tung Chang and Ling Hsiao. The Riemann problem and interaction of waves in gas dynamics. Recon Tech. Report A 90, NASA STI, 1989.
- [137] Frank H Shu. *The Physics of Astrophysics: Gas Dynamics*, volume 2. University Science Books, New York, 1991.
- [138] Gerald Beresford Whitham. *Linear and Nonlinear Waves*, volume 42. John Wiley & Sons, New York, 2011.
- [139] Richard Courant and Kurt Otto Friedrichs. *Supersonic flow and shock waves*, volume 21. Springer Science & Business Media, Boston, 1999.
- [140] Jacob Bear. *Dynamics of fluids in porous media*. Courier Corporation, New York, 2013.
- [141] Olga Arsen'evna Oleinik. Discontinuous solutions of non-linear differential equations. *Uspekhi Matematicheskikh Nauk*, 12(3):3–73, 1957.
- [142] Joel Smoller. *Shock waves and reaction-diffusion equations*, volume 258. Springer Science & Business Media, Boston, 2012.

- [143] Peter Lax. Shock waves and entropy. In Eduardo H. Zarantonello, editor, *Contributions to nonlinear functional analysis*, pages 603–634. Elsevier, New York, 1971.
- [144] Amiram Harten. On the symmetric form of systems of conservation laws with entropy. *J. Comput. Phys.*, 49:151–164, 1983.
- [145] Randall J LeVeque. *Numerical methods for conservation laws*, volume 132. Springer, New York, 1992.
- [146] Stanley Osher and Sukumar Chakravarthy. High resolution schemes and the entropy condition. *SIAM J. Num. Anal.*, 21(5):955–984, 1984.
- [147] Andrew Majda and Stanley Osher. Numerical viscosity and the entropy condition. *Comm. Pure Appl. Math.*, 32(6):797–838, 1979.
- [148] William Rowan Hamilton. The hodograph, or a new method of expressing in symbolic language the newtonian law of attraction. *Proc. Roy. Irish Acad.*, 3:344–353, 1847.
- [149] P A Clarkson, A S Fokas, and M J Ablowitz. Hodograph transformations of linearizable partial differential equations. *SIAM J. Appl. Math.*, 49(4):1188–1209, 1989.
- [150] Nathan J Kutz, J L Proctor, and S L Brunton. Applied Koopman theory for partial differential equations and data-driven modeling of spatio-temporal systems. *Complexity*, 2018, 2018.
- [151] Qin Li, Jian-Guo Liu, and Ruiwen Shu. Sensitivity analysis of burgers’ equation with shocks. *SIAM/ASA Journal on Uncertainty Quantification*, 8(4):1493–1521, 2020.
- [152] Muhammad R Mohyuddin, AM Siddiqui, T Hayat, J Siddiqui, and S Asghar. Exact solutions of time-dependent navier-stokes equations by hodograph-legendre transformation method. *Tamsui Oxford Journal of Mathematical Sciences*, 24(3):257–268, 2008.

- [153] Jiequan Li and Yuxi Zheng. Interaction of rarefaction waves of the two-dimensional self-similar euler equations. *Archive for rational mechanics and analysis*, 193(3):623–657, 2009.
- [154] Hayden Schaeffer. Learning partial differential equations via data discovery and sparse optimization. *Proc. Roy. Soc. A*, 473(2197):20160446, 2017.
- [155] J. Bakarji and D. M. Tartakovsky. Data-driven discovery of coarse-grained equations. *J. Comput. Phys.*, 434:110219, 2021.
- [156] Hayden Schaeffer and Scott G McCalla. Sparse model selection via integral terms. *Phys. Rev. E*, 96(2):023302, 2017.
- [157] Giang Tran and Rachel Ward. Exact recovery of chaotic systems from highly corrupted data. *Multiscale Model. Simul.*, 15(3):1108–1129, 2017.
- [158] Hayden Schaeffer, Giang Tran, and Rachel Ward. Extracting sparse high-dimensional dynamics from limited data. *SIAM J. Appl. Math.*, 78(6):3279–3295, 2018.
- [159] Alexandre M Tartakovsky, Carlos Ortiz Marrero, Paris Perdikaris, Guzel D Tartakovsky, and David Barajas-Solano. Physics-informed deep neural networks for learning parameters and constitutive relationships in subsurface flow problems. *Water Resour. Res.*, 2020.
- [160] Nicholas Geneva and Nicholas Zabaras. Modeling the dynamics of PDE systems with physics-constrained deep auto-regressive networks. *J. Comput. Phys.*, 403:109056, 2020.
- [161] JONATHAN H Tu, CLARENCE W Rowley, DIRK M Luchtenburg, STEVEN L Brunton, and J NATHAN Kutz. On dynamic mode decomposition: Theory and applications. *J. Comput. Dyn.*, 1(2), 2014.
- [162] Steven L Brunton, Bingni W Brunton, Joshua L Proctor, Eurika Kaiser, and J Nathan Kutz. Chaos as an intermittently forced linear system. *Nature Comm.*, 8(1):1–9, 2017.

- [163] Samuel H Rudy, J Nathan Kutz, and Steven L Brunton. Deep learning of dynamics and signal-noise decomposition with time-stepping constraints. *J. Comput. Phys.*, 396:483–506, 2019.
- [164] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.*, 378:686–707, 2019.
- [165] Zichao Long, Yiping Lu, and Bin Dong. PDE-Net 2.0: Learning PDEs from data with a numeric-symbolic hybrid deep network. *J. Comput. Phys.*, 399:108925, 2019.
- [166] Kailiang Wu and Dongbin Xiu. Data-driven deep learning of partial differential equations in modal space. *J. Comput. Phys.*, page 109307, 2020.
- [167] Suraj Pawar, SM Rahman, H Vaddireddy, Omer San, Adil Rasheed, and Prakash Vedula. A deep learning enabler for nonintrusive reduced order modeling of fluid flows. *Phys. Fluids*, 31(8):085101, 2019.
- [168] Shaoxing Mo, Nicholas Zabararas, Xiaoqing Shi, and Jichun Wu. Deep autoregressive neural networks for high-dimensional inverse problems in groundwater contaminant source identification. *Water Resour. Res.*, 55(5):3856–3881, 2019.
- [169] Yimin Liu, Wenyue Sun, and Louis J Durlofsky. A deep-learning-based geological parameterization for history matching complex models. *Math. Geosci.*, 51(6):725–766, 2019.
- [170] Shing Chan and Ahmed H Elsheikh. A machine learning approach for efficient uncertainty quantification using multiscale methods. *J. Comput. Phys.*, 354:493–511, 2018.
- [171] Sharmila Karumuri, Rohit Tripathy, Ilias Bilonis, and Jitesh Panchal. Simulator-free solution of high-dimensional stochastic elliptic partial differential equations using deep neural networks. *J. Comput. Phys.*, 404:109120, 2020.

- [172] Rohit K Tripathy and Ilias Bilonis. Deep UQ: Learning deep neural network surrogate models for high dimensional uncertainty quantification. *J. Comput. Phys.*, 375:565–588, 2018.
- [173] Yin hao Zhu and Nicholas Zabaras. Bayesian deep convolutional encoder-decoder networks for surrogate modeling and uncertainty quantification. *J. Comput. Phys.*, 366:415–447, 2018.
- [174] Soledad Le Clainche and José M Vega. Higher order dynamic mode decomposition. *SIAM Journal on Applied Dynamical Systems*, 16(2):882–925, 2017.
- [175] Patrick Héas and Cédric Herzet. Low-rank dynamic mode decomposition: Optimal solution in polynomial-time. *arXiv preprint arXiv:1610.02962*, 2016.
- [176] Zhen Chen and Dongbin Xiu. On generalized residual network for deep learning of unknown dynamical systems. *Journal of Computational Physics*, 438:110362, 2021.
- [177] Moritz Hardt and Tengyu Ma. Identity matters in deep learning. *arXiv preprint arXiv: 1611.04231*, 2018.
- [178] Andrew Stuart and Anthony R Humphries. *Dynamical systems and numerical analysis*, volume 2. Cambridge University Press, 1998.
- [179] Clarence W Rowley and Scott TM Dawson. Model reduction for flow analysis and control. *Annual Review of Fluid Mechanics*, 49:387–417, 2017.
- [180] Milan Korda and Igor Mezić. Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control. *Automatica*, 93:149–160, 2018.
- [181] Joshua L Proctor, Steven L Brunton, and J Nathan Kutz. Generalizing koopman theory to allow for inputs and control. *SIAM Journal on Applied Dynamical Systems*, 17(1):909–930, 2018.

- [182] Sebastian Peitz, Samuel E Otto, and Clarence W Rowley. Data-driven model predictive control using interpolated koopman generators. *SIAM Journal on Applied Dynamical Systems*, 19(3):2162–2193, 2020.
- [183] Jamie Johns. A Matlab code for numerical solution of Navier-Stokes equations for two-dimensional incompressible flow (velocity-pressure formulation) along with ability for importing custom scenarios for the fluid flow. <https://github.com/JamieMJohns/Navier-stokes-2D-numerical-solve-incompressible-flow-with-custom-scenarios-MATLAB->, 2018.
- [184] Hannah Lu and Daniel Tartakovsky. Model reduction via dynamic mode decomposition. *arXiv preprint arXiv:2204.09590*, 2022.
- [185] Y.-H. Tsai, Y. Giga, and S. Osher. A level set approach for computing discontinuous solutions of Hamilton-Jacobi equations. *Math. Comput.*, 72(241):159–181, 2003.