REDUCED ORDER MODELING OF CONVECTION-DOMINATED FLOWS, DIMENSIONALITY REDUCTION AND STABILIZATION

BY

RAMBOD MOJGANI

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Aerospace Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2020

Urbana, Illinois

Doctoral Committee:

Professor Daniel J. Bodony, Chair
Dr. Maciej Balajewicz, Director of Research
Associate Professor Marco Panesi
Assistant Professor Andres Goza
Assistant Professor Mattia Gazzola

Abstract

We present methodologies for reduced order modeling of convection dominated flows. Accordingly, three main problems are addressed.

Firstly, an optimal manifold is realized to enhance reducibility of convection dominated flows. We design a low-rank auto-encoder to specifically reduce the dimensionality of solution arising from convection-dominated nonlinear physical systems. Although existing nonlinear manifold learning methods seem to be compelling tools to reduce the dimensionality of data characterized by large Kolmogorov n-width, they typically lack a straightforward mapping from the latent space to the high-dimensional physical space. Also, considering that the latent variables are often hard to interpret, many of these methods are dismissed in the reduced order modeling of dynamical systems governed by partial differential equations (PDEs). This deficiency is of importance to the extent that linear methods, such as principle component analysis (PCA) and Koopman operators, are still prevalent. Accordingly, we propose an interpretable nonlinear dimensionality reduction algorithm. An unsupervised learning problem is constructed that learns a diffeomorphic spatio-temporal grid which registers the output sequence of the PDEs on a non-uniform time-varying grid. The Kolmogorov n-width of the mapped data on the learned grid is minimized.

Secondly, the reduced order models are constructed on the realized manifolds. We project the high fidelity models on the learned manifold, leading to a time-varying system of equations. Moreover, as a data-driven model free architecture, recurrent neural networks on the learned manifold are trained, showing versatility of the proposed framework.

Finally, a stabilization method is developed to maintain stability and accuracy of the projection based ROMs on the learned manifold *a posteriori*. We extend the eigenvalue reassignment method of stabilization of linear time-invariant ROMs, to the more general case of linear time-varying systems. Through a post-processing step, the ROMs are controlled using a constrained nonlinear lease-square

minimization problem. The controller and the input signals are defined at the algebraic level, using left and right singular vectors of the reduced system matrices. The proposed stabilization method is general and applicable to a large variety of linear time-varying ROMs.

Acknowledgments

I would like to thank Dr. Balajewicz for his mentorship, continual support and the scientific contributions that lead to the development of many of the ideas and methods in this dissertation.

I am also grateful to other members of the committee: Professor Dutton, Associate Professor Panesi, Assistant Professor Goza, Assistant Professor Gazzola, and Professor Bodony for their invaluable feedback. Specifically, I'd like to thank Prof. Dutton and Prof. Panesi for their guidance that helped me with clarification of the research direction, and for the experience I gained as their teaching assistant. I'm grateful to Prof. Gazzola for the series of discussions that helped me to explore the method in more details and encouraged me to present my ideas for the future directions of the research. I'm thankful to Prof. Goza and Prof. Bodony for the discussions and comments leading to clearer presentation, notations, schematics of the proposed methods and some ideas for the future directions of the research.

This material is based upon the work supported by the Grainger College of Engineering, Computational Science and Engineering fellowship (2017-2018), the National Science Foundation under Grant No. NSF-CMMI-14-35474, and the Air Force Office of Scientific Research under Grant No. FA9550-17-10203.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AE	Auto-encoder
ALE	Arbitrary Lagrangian-Eulerian
ANN	Artificial Neural Networks
DMD	Dynamic Mode Decomposition
GRU	Gated Recurrent Unit
HFM	High Fidelity Model
LTI	Linear Time-Invariant
LTV	Linear Time-Varying
LSTM	Long Short-Term Memory
MOR	Model Order Reduction
NN	Neural Networks
PCA	Principle Component Analysis
PDE	Partial Differential Equation
POD	Proper Orthogonal Decomposition
RNN	Recurrent Neural Network
ROM	Reduced Order Model
SD	Standard deviation
sPOD	shifted POD
SVD	Singular Value Decomposition
UQ	Uncertainty Quantification

List of Symbols

A	Nozzle area
$\boldsymbol{A}$	System (/state) matrix
$\boldsymbol{A}_r$	Reduced system (/state) matrix
$\hat{\boldsymbol{A}}_r$	Stabilized reduced system (/state) matrix
$\boldsymbol{B}$	Input matrix
$\boldsymbol{B}_c$	Controller matrix
$\boldsymbol{B}_r$	Reduced input matrix
$\boldsymbol{b}_d$	Bias vector in decoder layer(s)
$\boldsymbol{b}_e$	Bias vector in encoder layer(s)
$\boldsymbol{C}$	Output matrix
$\boldsymbol{C}_r$	Reduced output matrix
d	Damping coefficient (in the mechanical vibration problem)
$d_n\,()$	Kolmogorov n-width
$\boldsymbol{f}\,()$	Nonlinear function representing a dynamical system
$f_{\text{RNN}}\,()$	RNN cell trained on the reduced latent space
f_{enc}	Activation function in encoder layer(s)
$\mathcal{G}\,(.)$	Mapping from the the constant grid to the time-varying grid
$\mathcal{G}^{-1}\,(.)$	Mapping from the parameter/time-varying grid to the constant grid
f_{dec}	Activation function of decoder layer(s)
$\boldsymbol{w}_r$	Vector of latent variables
$\boldsymbol{K}$	Feedback (gain) matrix
k	Rank of $\widetilde{\boldsymbol{M}}$ on the POD subspace
k_n	n^{th} spring coefficient (in the mechanical vibration problem)
k_r	Rank of $\widetilde{\boldsymbol{M}}$ on the identified manifold
$\mathcal{L}$	Loss function
$\boldsymbol{M}$	Snapshot matrix
$\mathcal{M}$	Normed linear space
$\widetilde{\boldsymbol{M}}$	Low-rank approximation of $\boldsymbol{M}$(stated on the constant grid in POD, and stated on the time-varying grid in the proposed approach)
$\widetilde{\boldsymbol{\mathcal{M}}}_n$	n-dimensional subspace on $\mathcal{M}$
m	Mass (in the mechanical vibration problem)
$\mathcal{N}\,(\mu, SD)$	Normal distribution with mean value of μ and standard deviation of SD
N_t	Number of time-levels in temporal discretization
N_x	Number of (spatial) grid points
p	Pressure (in fluid flow problems)

p	Period (in periodic systems)
$\boldsymbol{q}$	Flux vector
t	Time
$\boldsymbol{U}$	Matrix of spatial bases
$\boldsymbol{U}_{ru}$	Weight matrix acting on input variables on the latent manifold
$\boldsymbol{U}_x$	Matrix of spatial bases of the parameter/time-varying grid
u	Flow velocity in x-direction
$\boldsymbol{u}_c$	Vector of controller input
$\boldsymbol{u}_n$	n^{th} column of $\boldsymbol{U}$
$\boldsymbol{V}_x$	Matrix of temporal bases of the parameter/time-varying grid
v	Flow velocity in y-direction
$\boldsymbol{v}$	Vector of grid volume (equivalent to vector of grid spacing in one-dimensional space)
$v_{\min}$	Minimum admissible grid volume (equivalent to minimum admissible grid spacing in one-dimensional space)
$\boldsymbol{v}_n$	n^{th} column of $\boldsymbol{V}$
$\boldsymbol{v}_{xn}$	n^{th} column of $\boldsymbol{V}_x$
$\boldsymbol{W}_d$	Weight matrix acting on state variables in the decoder layer
$\boldsymbol{W}_e$	Weight matrix acting on state variables in the encoder layer
$\boldsymbol{W}_{rr}$	Weight matrix acting on state variables on the latent manifold
$\boldsymbol{W}_{wr}$	Weight matrix acting on state variables in the decoder
$\boldsymbol{w}$	State parameter
$\boldsymbol{w}_r$	State parameter on the reduced space
$\boldsymbol{X}$	Matrix of constant grid point locations
$\widetilde{\boldsymbol{X}}$	Matrix of time-varying grid point locations
$\boldsymbol{x}$	Vector of grid point locations
$\widetilde{\boldsymbol{x}}_i$	Vector of grid point locations corresponding to the i^{th} parameter/time level of time-varying grid
$\boldsymbol{y}$	System output
$\hat{\boldsymbol{y}}_r$	Output of the stabilized system
α	System parameter(in the mechanical vibration problem)
Δn	Intermittency of the controller (activation in time)
$\boldsymbol{\Gamma}_1$	Tikhonov matrix (corresponding to the spatial bases)
$\boldsymbol{\Gamma}_2$	Tikhonov matrix (corresponding to the parameter/temporal bases)
γ	Specific heat ratio
ε	Error
θ	Set of training parameters
λ	Eigenvalue of a matrix
μ	System parameter (in the parameter-varying Nozzle problem)
$\boldsymbol{\Pi}$	Monodromy matrix
ρ	Density
$\boldsymbol{\Sigma}$	Matrix of singular value
$\boldsymbol{\Sigma}_c[n]$	Controller matrix (of singular values)at n^{th} time-level
$\boldsymbol{\Sigma}_{\boldsymbol{l}}[n]$	Truncated matrix of singular values (l leading singular values) at n^{th} time-level
$\boldsymbol{\Sigma}^{\boldsymbol{\Delta}}$	Singular values matrix of the stabilized system

σ	Singular value
$\boldsymbol{\phi_A}[n, n_0]$	State transition matrix between n and n_0
Ω	Computational domain
$\partial\Omega$	Boundary of the computational domain

Chapter 1

Introduction

Numerical simulations of nonlinear fluid flow systems often require very large computational resources. This can be especially true in the case of high-Reynolds number flows, compressible flows, and combustion, where very fine spatial and temporal discretization are required to adequately resolve and propagate the flow states. These computational costs often hinder the use of high-fidelity computer simulations for applications where repeated realizations of the system are required [3–5]. These applications include, for example, interdisciplinary design and shape optimization [6–11], uncertainty quantification (UQ) [12–15], and real-time control [16, 17]. There are, therefore, significant engineering and scientific benefits to developing and studying model order reduction (MOR) approaches that seek to reduce the size and cost of the computational models while minimizing the loss of physical fidelity.

Projection-based or artificial neural network-based model order reduction approaches are of the two generally accepted approaches to overcome this challenge. The goal is to leverage the abundance of computational resources in the off-line stage in order to develop low-cost models capable of delivering real-time/low-cost solutions in the on-line stage. The investment in the off-line stage can often be justified by frequent realizations or limited resources in the on-line stage.

Reduced Order Models (ROMs), by definition, are particularly efficient in representing systems with identifiable low-rank coherent structures [18], where the system states can be reproduced on a manifold using a relatively low number of bases. In the context of linear time-invariant (LTI) systems, the rank of the system is described precisely by the Hankel singular values of the system [4]. Similarly, for nonlinear and/or parameterized systems the rate of decay of the singular values of a matrix of snapshots of the solution, is usually used as a measure for the projection-based ROM efficiency [4, 19–26]. The fast decay of singular values corresponding to elliptic and parabolic partial differential equations (PDEs) are well studied [27–29]. In the field of fluid dynamics, examples of

such problems are low-Reynolds number flow, where the diffusion term dominates the dynamics. Contrarily, in a large class of engineering problems, such as convection dominated flows featuring traveling wave-like solutions with moving sharp gradients, such as shocks and interfaces, the rate of decay of the singular values is substantially slower, and unfortunately the first premise of an efficient ROM cannot be met. Therefore, to construct efficient ROMs of high-Reynolds number flows, supersonic flows, acoustic waves, and multi-phase flows with distinct moving interfaces, the more fundamental problem of reducibility of the system has to be resolved. Similar deficiencies also prevail outside the field of fluid dynamics, examples include reduction of models in computational finance [30] and spread of infectious diseases [31], as well as different applications in image processing such as object tracking [32]. Formally, the Kolmogorov n-width of the solution space indicates how close the n-dimensional subspaces can approximate the state-space.

Even in the most modern MOR approaches the state variables are approximated on linear subspaces [4, 33–40]. Typically, these subspaces are identified via Proper Orthogonal Decomposition (POD) [41–47], Dynamic Mode Decomposition (DMD) [48–55], balanced POD [56, 57] (based on balanced truncation [58]), and reduced basis methods [59]. Many of these linear low-dimensional subspaces can be replicated in an artificial neural networks; however, the cost of training of such architectures are considerably higher. The appeal of neural networks is their capability to identify the nonlinear manifolds simply by using nonlinear activation functions, and therefore enhancing reducibility. However, these universal approximators are developed without considering the physics of the problems and therefore lack interpretability.

Over the years, a large variety of MOR methodologies have been proposed to address reduction of convection dominated flow. These method are mostly centered on the traditional projection-based models. However, there is a more recent growing attention on the artificial neural network-based (ANN) models. The attempts to address reduction of convection-dominated flows can be summarized in three main categories.

In the first family of methods, symmetries or self-similarities concealed in the system or data are exploited via different forms of transformations (interpolation, scaling, rotation, or translation) [22, 45, 60–65]. Rowley and Marsden [60] apply the Karhunen-Loéve decomposition to nonlinear one-dimensional equations with continuous translational symmetry by introducing a method to find reconstruction equations for such problems. The amount of shifts in the reconstruction equations

are a function of time and are determined by template fitting [66] or centering [67]. The method is extended using Lie algebra to accommodate rescaling in time [61]. Mowlavi and Sapsis [64] extend the approach to stochastic systems. Rim et al. [22] propose an iterative algorithm called transport reversal that decomposes the snapshot matrix into multiple shifting profiles. The transport reversal matrix dictates the required shift of the profile to minimize the distance between the initial condition and the solution in a later time. This method is extensible to multidimensional problems via the Radon transform [68]. Another set of closely related approaches incorporate the transformations in the definition of the bases, e.g. Lagrangian DMD ($\mathcal{L}$DMD) [21] or shifted POD (sPOD) [23, 69–71]. In the latest efforts, Mendible et al. [26] develop an unsupervised machine learning algorithm based on a library of candidate wave-speeds to generalize the sPOD modes with time-varying speeds. In another series of approaches, a spatio-temporal domain using time-varying maps are learned, these maps are trained to track the traveling features of the solution [72, 73].

In the second family of methods, global bases are replaced with local bases [74–76] or split bases [77]. Amsallem et al. [74] formulate the criteria to partition the domain (time or variable-space) into sub-regions in which each sub-region is assigned with corresponding reduced-order bases. The sub-region selection procedure scales with the dimension of the low-order model, rather than the high-fidelity model, to maintain the computational efficiency. Carlberg [77] enriches the basis by splitting a given basis vector into several vectors which minimizes the error for a quantity of interest. Lucia [75] decomposes the domain into overlapping subdomains to isolate the region containing translating shocks. The solution is ensured to be smooth using constraints in the optimization-based solver.

Finally, in the third family of methods, traditional MOR is generalized by adding some form of temporal dependency in the spatial bases [20, 26, 63, 78]. In Iollo and Lombardi [20], transport and diffusion phenomena are decoupled and treated independently. Transport is modeled by solving an optimal transport problem while diffusion is expressed via traditional, stationary global modes. Finally, in Gerbeau and Lombardi [63], the spatial bases are evolved in time using the eigen-bases of the governing PDEs.

In this thesis, we address the problem of irreducibility regardless of the type of the reduced order model, i.e. we propose a low-rank registration-based manifold to decrease the Kolmogorov n-width of the solution space.

1.1 Motivations

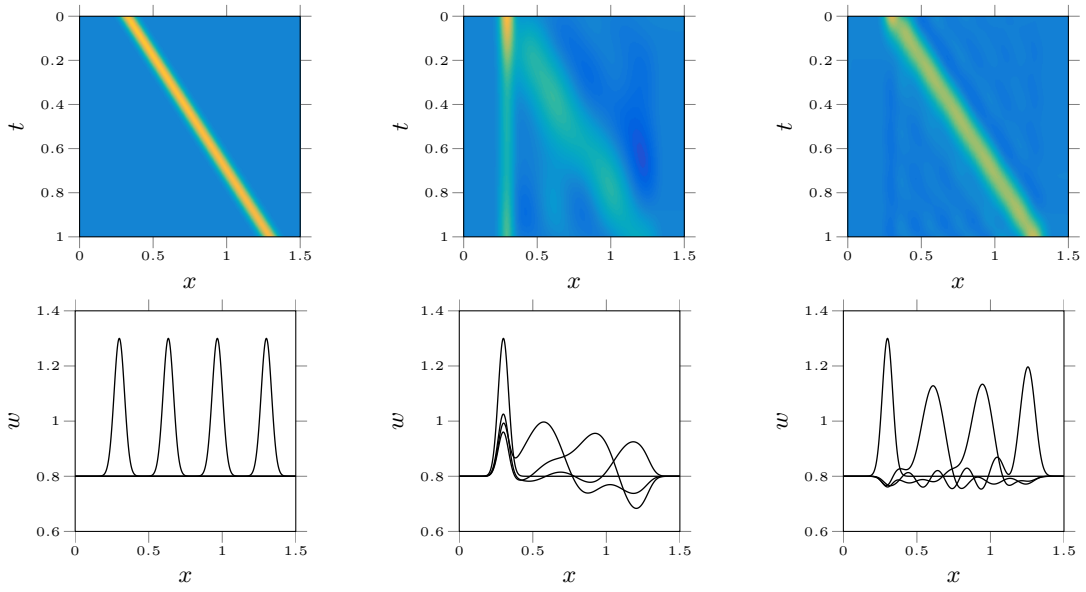
Despite the efficacy and success of tools such as POD, it is well known that linear dimensionality reduction techniques often miss important convection-dominated structures in the data. Consider, for example, the scalar linear convection equation

$$\begin{cases} \dfrac{\partial w}{\partial t} + \dfrac{\partial w}{\partial x} &= 0, \quad (x,t) \in [0,\infty) \times [0,1], \\ w(x,0) &= 0.8 + 0.5\ e^{-(x-0.3)^2/0.05^2}, \\ w(0,t) &= 0.8. \end{cases} \tag{1.1}$$

This initial value problem models a flow in a one-dimensional domain and has the exact solution $w(x,t) = w(x-t,0)$. The exact solution to this equation, and the corresponding POD approximations using $k = 5$ and $k = 10$ POD modes are given in Fig. 1.1a, Fig. 1.1b, and Fig. 1.1c, respectively. The contour plots illustrate the solution in the $x - t$ plane while the line plots show the solution at $t \in \{0, 1/3, 2/3, 1\}$. Despite the relative simplicity of the governing equation, this particular solution clearly violates the assumption that the solution can be accurately approximated using a small linear subspace. Moreover, because the POD decomposition is optimal, no other subspace can be expected to outperform the POD solutions illustrated in Fig. 1.1. The solutions are fundamentally high-rank and therefore, cannot be efficiently compressed via linear dimensionality reduction techniques such as POD.

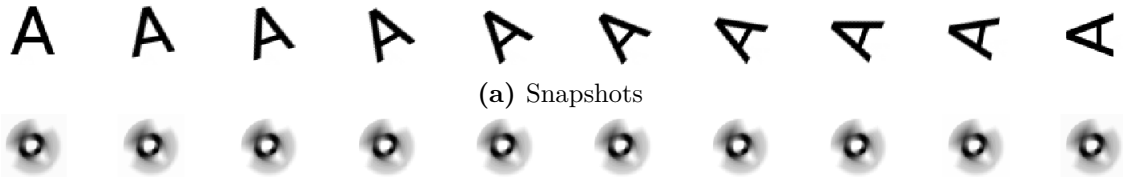
It must also be emphasized that the presence of a convective term is not a necessary nor sufficient condition for this bottleneck. Firstly, it is not a sufficient condition because not all solutions of the convection equation are high-rank. For example, the solution for initial conditions $w_0 = \sin(x)$ in the domain $(x,t) \in [0, 2\pi] \times [0, T]$ with periodic boundary conditions is exactly rank two, and therefore, would be reproduced exactly - without error - using exactly two POD modes. Secondly, it is not a necessary condition because high-rank solutions often arise in raw data-sets. For example, consider the well known benchmark problem in machine learning and computer vision – a data-set comprised of a rotated character "A". The character "A" is stored in a 50×50 matrix and is rotated a total of 90 degrees with 3 degrees increments resulting in a snapshot matrix of dimension 2500×31. A representative sample of the snapshots is shown in Fig. 5.1a while a single POD mode reconstruction is illustrated in Fig. 5.1b. As expected, a linear dimensionality approach such as POD is not capable

of extracting the intrinsic low-dimensional characteristics of this problem.



(a) Exact solution **(b)** $k = 5$ POD **(c)** $k = 10$ POD

Figure 1.1: Traditional POD approximations of a solution of the scalar convection equation.



(a) Snapshots

(b) $k = 1$ POD approximation of the snapshots

Figure 1.2: 90 degrees rotation of character "A"

1.2 Contributions

The objective of this thesis is to develop an accurate and stable low-rank reduced order model for convection dominated nonlinear flows. The major contribution of this work are as follows:

1. **An efficient and low-rank registration-based auto-encoder and reduced order models**

   We address the fundamental limitation of common approaches used for MOR of convection-dominated nonlinear flows. The traditional POD low-rank optimization problem is generalized to learn a nonlinear manifold. In practice, a registration process is defined that minimizes the Kolmogorov n-width of the mapped snapshots. From a classical physics viewpoint, it can be seen as a change of frame of reference to a parameter/time-varying spatial grid, i.e. an arbitrary Lagrangian-Eulerian (ALE) framework. The registration-based optimization problem yields two set of independent global bases: the first approximates the state variables of the system while the second approximates the location of the temporally evolving computational grid. This proposed new approach offers several advantages over previous methods. First, because the proposed approach can be interpreted as a data-driven generalization of Rowley and Marsden [60], is not limited by any of the difficulties associated with solving auxiliary reconstruction equations [22, 64]. Second, due to this data-driven formulation, the proposed method is also, in principle, extendable to arbitrary physical and parameter dimensions. This is in contrast to other methods [60, 64] where extensions beyond a single dimension are infeasible. Finally, the proposed approach is implemented in both projection-based ROMs and neural network-based ROMs. Therefore, the burden of high training costs associated with universal approximators such as [24, 79, 80], can be avoided in presence of the governing PDEs. Otherwise, it enhances both the accuracy and the training costs of the universal approximators.

2. **Enabling the predictive capabilities in ROMs of convection-dominated flows**

   We extend the proposed ROM formulation to enable the predictive simulation beyond the range of the training snapshots, a known challenge for both projection-based and neural network-based ROMs. In the proposed method, the identified manifold is evolved in time and

is extrapolated beyond the training regime. This task has become feasible in the proposed framework, since the time-varying grid follows a low-rank representation of characteristics underlying the convection dominated problem.

3. **Stabilization of linear time-varying reduced order models**

   We propose a new approach to stabilize discrete linear time-varying projection-based reduced order models, since the ROMs on the identified manifolds are time-varying dynamical systems and lack an *a priori* stability guarantee, a common construct of projection-based ROMs. We have extended the optimization-based eigenvalue reassignment method for LTIs [81], to linear time-varying ROMs through an optimization-based singular value calibration method, an *a posteriori* stabilization method. The approach calibrates the largest singular values while maintaining the singular vector and minimizing the ROM error. Since the energy decay/growth of the state parameters in LTVs is bounded by products of the largest singular values of the sequence of system matrices, the singular values, instead of the eigenvalues, are calibrated. As a byproduct of this choice, the eigenvalues of the system matrices are reassigned without requiring any further assumption on their type, i.e. real versus complex-conjugate pairs, something that must be specifically treated in the original method of eigenvalues reassignment [81].

1.3 Thesis outline

This thesis is organized as follows. In §1, we propose a manifold on which convection dominated PDEs can be optimally reduced by introducing an unsupervised learning problem. Subsequently, in §3 we discuss the procedure to construct the projection-based and neural network-based reduced order models on the learned manifolds. Projection of the PDEs on the learned manifold leads to time-varying ROMs, therefore in §4, a feedback controller is designed to algebraically stabilize the corresponding linear time-varying ROMs. In §5 the proposed approach is applied to several representative problems. Finally, in §6, the contributions of this thesis and the future prospects are laid out.

Chapter 2

Identifying an optimal manifold

Any of the model order reduction approaches has two essential elements. First, identifying a subspace or manifold on which the solution can be represented. Secondly, constructing the reduced order dynamical system on the identified subspace/manifold. In this section, we focus on the first element of building any ROMs, which we demonstrated to be the bottle-neck of ROMs for convection-dominated flows (§1.1). The ROMs are constructed in §3.

In section §2.1, the concepts and definitions corresponding to manifold learning and reducibility of ROMs are discussed. The unsupervised learning problem to reduce dimensionality of the convection-dominated flows is posed and discussed in §2.2. The reducibility of different snapshots using the proposed framework is demonstrated in §5.1.

2.1 Preliminaries

The goal of reduced order modeling is to leverage the vast amount of data generated from high accuracy simulations to learn a low-dimensional model that can accurately and efficiently approximate the underlying dynamical system. This is especially a challenging task for convection dominated PDEs, where the Kolmogorov n-width of the snapshots of the solution is relatively large, i.e. the solution cannot be effectively reduced on a linear subspace. Such problems emerge frequently in a broad range of applications, from Navier-Stokes equations (fluid dynamics) to Schrödinger equation (quantum-mechanical systems) [26]. In the machine learning community, the recognition of similar challenge dates back to 1990s and attempts in classification of handwritten digits [82], where presence of simple transformations such as translations and rotations in the data-set is well known to dramatically deteriorate the accuracy of linear methods such as principle component analysis (PCA). Fundamentally, other linear manifolds (subspaces) suffer from similar drawbacks, examples include

proper orthogonal decomposition (POD), multidimensional scaling (MDS) [83], factor analysis [84] and independent component analysis (ICA) [84]. Therefore, the high dimensionality of the data on any of these linear manifolds has incentivized a slew of nonlinear manifold learning approaches, such as Iso-map [85], kernel PCA [86], locally linear embedding (LLE) [87], Laplacian eigenmaps (LEM) [88], semi-definite embedding (SDE) [89], auto-encoders [90], t-SNE [91], and diffeomorphic dimensionality reduction [92].

Although many of the aforementioned nonlinear methods provide the sought after low-dimensional manifold, only a few provide the mapping from the learned low-dimensional to the high-dimensional manifold, for a survey [see 93]. This is especially important in reduced order modeling of PDEs, since the models are to be evolved in the parameters space or time on the low-dimensional manifold, i.e. evolving of the latent variables, and subsequently the latent variables have to be mapped to the physical high-dimensional manifold. Auto-encoders (AE), specifically convolutional auto-encoders (CAEs) [94] and deep convolutional generative adversarial networks (DCGANs) [95], are amongst the successful methods used in dimensionality reduction of PDEs [93, 96]. However, linear manifolds such as proper orthogonal decomposition (POD) and dynamic mode decomposition (DMD) are still often extensively preferred to these nonlinear approaches, since they provide an interpretable framework for analysis of the system, as well as controlling of the reduced system. POD reveals the coherent structures in fluid flows [46, 47], and DMD obtains a finite-dimensional, matrix approximations of the Koopman operator, which opens the possibility of taking advantage of estimation and control theories developed for the linear systems [97]. In a more recent effort, it is shown that deep AE architectures can be trained to transform nonlinear PDEs into linear PDEs, by learning the eigen-function of the Koopman operator [98]. In this approach, although the transformation is nonlinear, the latent variables lie on a linear subspace. Finally, a similar approach that prioritizes the optimal reducibility by learning a nonlinear manifold leads to a low-dimensional latent space. Therefore, by definition, such an approach results in a more efficient reduced order model.

In this section, we develop an auto-encoder to learn a manifold on which the reduced order models can be efficiently constructed. To this end, we pose an unsupervised learning problem, that learns a spatio-temporal grid on which the low-rank linear decomposition of the solution of the PDE is optimal. The method can be interpreted as learning a map that registers the output sequence of a convection-dominated PDE to a low-rank reconstruction of the solution. The method is in spirit

of registration based manifold learning approaches, e.g. [92, 99].

2.1.1 Reducibility and linear manifolds

In approximation theory, Kolmogorov n-width is used to measure how well the n-dimensional subspaces can approximate the solution manifold, $\mathcal{M}$ [100]. In other words, the "success" of linear dimension reduction is tied to availability of an accurate enough approximation for the solution manifold [101] and the decay of the n-width describes the best achievable error by an n-term approximation [102]. The following definitions and theorems from [103] briefly explains this measure and its connection to convection-dominated flows.

Definition: Let $\mathcal{M}$ be a normed linear space and $\widetilde{\boldsymbol{\mathcal{M}}}_n$ any n-dimensional subspace of $\mathcal{M}$. For each $x \in \mathcal{M}$, $\delta\left(x, \widetilde{\boldsymbol{\mathcal{M}}}_n\right)$ shall denote the distance of the n-dimensional subspace $\widetilde{\boldsymbol{\mathcal{M}}}_n$ from x, defined by

$$\delta\left(x; \widetilde{\boldsymbol{\mathcal{M}}}_n\right) = \inf\left\{\|x - y\|_X : y \in \widetilde{\boldsymbol{\mathcal{M}}}_n\right\}. \tag{2.1}$$

If there exist a $y^* \in \widetilde{\boldsymbol{\mathcal{M}}}_n$ for which $\delta\left(x, \widetilde{\boldsymbol{\mathcal{M}}}_n\right) = \|x - y^*\|$, then y^* is the best approximation of x from $\widetilde{\boldsymbol{\mathcal{M}}}_n$. Extending the concept from a single element of x to $\mathcal{S}$, a given subset of $\mathcal{M}$, the deviation of $\mathcal{S}$ from $\widetilde{\boldsymbol{\mathcal{M}}}_n$ is defined as

$$\delta\left(\mathcal{S}; \widetilde{\boldsymbol{\mathcal{M}}}_n\right) = \sup_{x \in \mathcal{S}} \inf_{y \in \widetilde{\boldsymbol{\mathcal{M}}}_n} \|x - y\|, \tag{2.2}$$

representing the worst element of $x \in \mathcal{S}$ approximated in $\widetilde{\boldsymbol{\mathcal{M}}}_n$.

Definition: Kolmogorov n-width of $\mathcal{M}$, $d_n(\mathcal{M})$, is defined as

$$d_n(\mathcal{S}; \mathcal{M}) := \inf_{\widetilde{\boldsymbol{\mathcal{M}}}_n} \delta\left(\mathcal{S}; \widetilde{\boldsymbol{\mathcal{M}}}_n\right), \tag{2.3}$$

where the infimum is taken over all n-dimensional subspaces ($\widetilde{\boldsymbol{\mathcal{M}}}_n$) of the state space, $\mathcal{M}$.

In the context of Petrov–Galerkin projection schemes, n-width identifies the best achievable rate of convergence for a given set of input data [104]. The connection between SVD of the Hankel operator and the Kolmogorov n-width is rigorously established [105, 106]. More specifically, it can be shown by appropriately defining the subspaces minimizing the Kolmogorov n-width, it is then equivalent to $(n+1)^{st}$ Hankel singular value [107]. This connection combines the concepts and

literature in regard to rate of decay of singular values, Kolmogorov n-width and acceptable accuracy and feasibility of a low-rank reduced order model.

2.1.2 POD-based subspace learning

In the context of spectral methods in computational fluid dynamics (CFD), the spatial bases are usually analytical functions, e.g. trigonometric functions or Chebyshev polynomials. In the context of MOR, the spatial bases are derived *a posteriori* from a snapshot of a solution dataset. Over the years, a large variety of approaches for generating bases from snapshots have been developed. We confine our attention to bases generated via the proper orthogonal decomposition (POD) [41–47].[1]

Constructing the bases from the snapshots in the spirit of the POD method can be formulated mathematically as a low-rank matrix approximation problem as follows:

For a given snapshot matrix $\boldsymbol{M} \in \mathbb{R}^{N\times K}$*, find a lower rank matrix* $\widetilde{\boldsymbol{M}} \in \mathbb{R}^{N\times K}$ *that solves the minimization problem*

$$\underset{\text{rank}(\widetilde{\boldsymbol{M}})=k}{\text{minimize}} \quad \left\|\boldsymbol{M}-\widetilde{\boldsymbol{M}}\right\|_F, \tag{2.4}$$

where $k \ll N$. A snapshot matrix is defined here as a matrix whose columns contain the states of the system of interest. More specifically, each column corresponds to the state of the system for some particular value of the system parameters, time, or the boundary/initial conditions. Hence, $\boldsymbol{M} = [\boldsymbol{w}_1, \cdots, \boldsymbol{w}_K]$, where $\boldsymbol{w}_i \in \mathbb{R}^N$ is the state at the i^{th} parameter/time step. In problem (2.4), the rank constraint can be taken care of by representing the unknown matrix as $\widetilde{\boldsymbol{M}} = \boldsymbol{U}\boldsymbol{V}$, where $\boldsymbol{U} \in \mathbb{R}^{N\times k}$ and $\boldsymbol{V} \in \mathbb{R}^{k\times K}$, so that problem (2.4) becomes

$$\underset{\boldsymbol{U},\,\boldsymbol{V}}{\text{minimize}} \quad \|\boldsymbol{M}-\boldsymbol{U}\boldsymbol{V}\|_F. \tag{2.5}$$

It is well known that the solution of the above low-rank approximation problem is given by the singular value decomposition (SVD) of $\boldsymbol{M}$. Specifically, $\boldsymbol{U} = [\boldsymbol{u}_1, \cdots, \boldsymbol{u}_k] \in \mathbb{R}^{N\times k}$ and $\boldsymbol{V} = \boldsymbol{\Sigma}\,[\boldsymbol{v}_1, \cdots, \boldsymbol{v}_k] \in \mathbb{R}^{k\times K}$, where $\boldsymbol{M} = \boldsymbol{U}^*\boldsymbol{\Sigma}^*\boldsymbol{V}^{*\mathrm{T}}$, $\boldsymbol{U}^* = [\boldsymbol{u}_1, \cdots, \boldsymbol{u}_k, \boldsymbol{u}_{k+1}, \cdots, \boldsymbol{u}_N]$, $\boldsymbol{V}^{*T} = [\boldsymbol{v}_1, \cdots, \boldsymbol{v}_k, \boldsymbol{v}_{k+1}, \cdots, \boldsymbol{v}_K]$, and $\boldsymbol{\Sigma}^* = \text{diag}\,(\sigma_1, \sigma_2, \cdots, \sigma_r)$ is a diagonal rank-r matrix of singular values, where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r$, and $\boldsymbol{\Sigma} = \text{diag}\,(\sigma_1, \sigma_2, \cdots, \sigma_k)$. This decomposition has a very

[1] It is emphasized however that the same principle is applicable to any basis generation method; such as, for example, the dynamic mode decomposition (DMD) and Koopman modes [48–55].

close connotation to factor analysis [84] and can be reproduced by artificial neural networks with linear activations [108]. These methods are discussed in more details in §2.1.3. Although the linearity of the learned manifold leads to inefficiencies in convection-dominated PDEs with large Kolmogorov n-width, the existence of a closed form solution as well as the abundance of computationally efficient approaches, such as [109], has made the method predominately be utilized in the field. Our goal is to extend the norm minimization problem of (2.5) to an interpretable and efficient nonlinear manifold learning problem.

2.1.3 Artificial neural network-based manifold learning

The growing investment in machine learning infrastructure has lead to abundance of efficient open-source software and task specific hardware designed to leverage universality of neural network in different applications. In the context of model order reduction of dynamical systems, neural networks are used to identify nonlinear manifolds [93] or even to construct an end-to-end representation of the dynamical systems [79, 80, 110].

In neural network architectures auto-encoders are often used to provide the dimensionality reduction step [79]. In other words, an encoder reduces the input state from an n-dimensional space to a k-dimensional space, a recurrent neural network (RNN) evolves the hidden state parameter on an k-dimensional space and then the hidden states can be decoded to the n-dimensional physical space to extract the quantity of interest. All the components of the architecture are trained simultaneously.

In the context of machine learning, auto-encoders are feature-extracting functions parameterized in a closed form [111]. The *encoder* function, f_{enc}, computes the *code* or the latent variable, $\boldsymbol{w}_r \in \mathbb{R}^k$, from the *input* vector, $\boldsymbol{w} \in \mathbb{R}^N$ given a set of weights and biases $\boldsymbol{\theta}_{\text{enc}}$, i.e. $\boldsymbol{w}_r = f_{\text{enc}}(\boldsymbol{w}; \boldsymbol{\theta}_{\text{enc}})$. Another closed-form parameterized function maps with the corresponding set of weights and biases $\boldsymbol{\theta}_{\text{dec}}$, *decodes* the latent variable onto the input space, $\widetilde{\boldsymbol{w}} = f_{\text{dec}}(\boldsymbol{w}_r; \boldsymbol{\theta}_{\text{dec}})$, resulting in a reconstruction. Conventionally, auto-encoders are developed as multi-layer perceptrons (MLPs) with nonlinear activation functions, i.e.

$$\boldsymbol{w}_r = f_{\text{enc}}(\boldsymbol{w}; \boldsymbol{\theta}_{\text{enc}}) = \sigma_{\text{enc}}(\boldsymbol{W}_e \boldsymbol{w} + \boldsymbol{b}_e), \tag{2.6a}$$

$$\widetilde{\boldsymbol{w}} = f_{\text{dec}}(\boldsymbol{w}_r; \boldsymbol{\theta}_{\text{dec}}) = \sigma_{\text{dec}}(\boldsymbol{W}_d \boldsymbol{w}_r + \boldsymbol{b}_d), \tag{2.6b}$$

where σ_{enc} and σ_{dec} are encoder and decoder activation functions (typically, the element-wise sigmoid or hyperbolic tangent non-linearity, or an identify function). The task of training an auto-encoder is to seek the set of weights and biases in

$$\boldsymbol{\theta}_{\text{enc}} = \left\{ \boldsymbol{W}_e \in \mathbb{R}^{k \times N}, \boldsymbol{b}_e \in \mathbb{R}^k \right\},$$

and

$$\boldsymbol{\theta}_{\text{dec}} = \left\{ \boldsymbol{W}_d \in \mathbb{R}^{N \times k}, \boldsymbol{b}_d \in \mathbb{R}^N \right\},$$

that minimizes the reconstruction error/loss, $\mathcal{L}$, i.e.

$$\underset{\boldsymbol{\theta}_{\text{enc}},\, \boldsymbol{\theta}_{\text{dec}}}{\text{minimize}} \quad \sum_{n=1}^{N_t} \mathcal{L}\left(\boldsymbol{w}[n], \widetilde{\boldsymbol{w}}[n]\right), \tag{2.7}$$

where $\widetilde{\boldsymbol{w}}[n] = f_{\text{dec}}\left(f_{\text{enc}}\left(\boldsymbol{w}[n]; \boldsymbol{\theta}_{\text{enc}}\right); \boldsymbol{\theta}_{\text{dec}}\right)$. A special case of auto-encoders with identity activation functions, no biases and minimizing a squared Euclidean reconstruction error, i.e.

$$\underset{\boldsymbol{W}_d,\, \boldsymbol{W}_e}{\text{minimize}} \quad \sum_{n=1}^{N_t} \left\| \boldsymbol{w}[n] - \boldsymbol{W}_d \boldsymbol{W}_e \boldsymbol{w}[n] \right\|_F^2, \tag{2.8}$$

given $\boldsymbol{W}_d = {\boldsymbol{W}_e}^T$, learns the same subspace as principal component analysis (PCA) [112], and therefore identifies the direction of the greatest variance in the data-set [90]. The PCA bases, or the POD bases as described in (2.5), can be recovered from the identified weights [113]. Poor performance of these approaches has been identified in machine learning community, e.g. in text classification problem in presence of even simple transformations such as translation and rotation [82]. In principle, a very similar challenge persists in the reduction of the convection dominated flows.

2.2 Proposed approach: Low-rank registration-based auto-encoder

We generalize the linear manifold learning problem of (2.4), as a nonlinear manifold learning as follows: For a given data-set lying on the high-dimensional state space, learn a manifold, and the corresponding mapping, on which the solution can be efficiently expressed as a linear low-rank

decomposition. The map from the high-dimensional physical manifold to the identified manifold is denoted by $\mathcal{G}(\boldsymbol{M}) : \mathbb{R}^{N\times K} \to \mathbb{R}^{N\times K}$ and its reverse by $\mathcal{G}^{-1}\left(\widetilde{\boldsymbol{M}}\right) : \mathbb{R}^{N\times K} \to \mathbb{R}^{N\times K}$.[2]

The proposed manifold learning minimization problem is in the following form:

$$\underset{\mathcal{G}^{-1}(.),\ \widetilde{\boldsymbol{M}}}{\text{minimize}} \quad \left\|\boldsymbol{M} - \mathcal{G}^{-1}\left(\widetilde{\boldsymbol{M}}\right)\right\|_F, \tag{2.9}$$

where $\widetilde{\boldsymbol{M}}$ is the mapped data on the learned manifold. In this case, we assume $\widetilde{\boldsymbol{M}}$ is low-rank and $\widetilde{\boldsymbol{M}} = \boldsymbol{U}\boldsymbol{V}$ is the rank-k_r linear decomposition of the mapped data, where $\boldsymbol{U} \in \mathbb{R}^{N\times k_r}$ and $\boldsymbol{V} \in \mathbb{R}^{k_r\times K}$. The map and the corresponding operators are illustrated in Fig. 2.1.

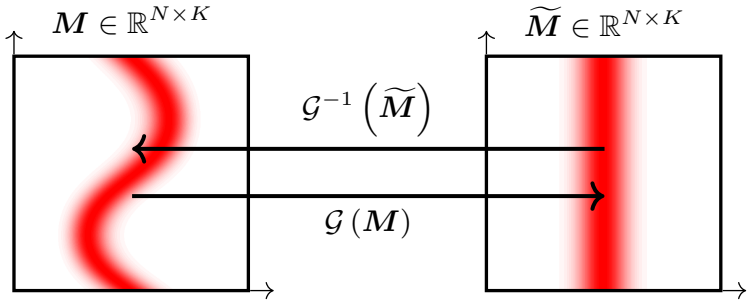


Figure 2.1: Illustration of the snapshots, $\boldsymbol{M}$, and the low-rank snapshots on the identified manifold, $\widetilde{\boldsymbol{M}}$, and the corresponding maps between the manifolds.

For a given map, $\mathcal{G}^{-1}(.)$, the rank-k_r decomposition of the mapped data, i.e.

$$\underset{\boldsymbol{U},\ \boldsymbol{V}}{\text{minimize}} \quad \|\mathcal{G}(\boldsymbol{M}) - \boldsymbol{U}\boldsymbol{V}\|_F, \tag{2.10}$$

can simply be computed by the SVD of $\mathcal{G}(\boldsymbol{M})$, instead of $\boldsymbol{M}$, as described in §2.1.2. In principle, the compression of the data on the learned manifold is lossless if $k_r \geq \text{rank}(\mathcal{G}(\boldsymbol{M}))$. However, aliasing errors lead to a limitation on the reconstruction accuracy on a finite-dimensional space (see §5.1.3). The proposed compression in (2.9) outperforms the traditional POD of (2.4), if and only if $\text{rank}(\mathcal{G}(\boldsymbol{M})) < \text{rank}(\boldsymbol{M})$.

2.2.1 Interpolation and Diffeomorphism

[2] In this section, we limit the discussion of the maps to an abstract level and the numerical details of these operators are postponed to §2.2.1.

In this section, we tie the idea of the identified manifold to a time/parameter-varying grid. The matrix of the constant grid point positions is $\boldsymbol{X} = \boldsymbol{x}_0 \odot \mathbf{1} = [\boldsymbol{x}_0, \boldsymbol{x}_0, \cdots, \boldsymbol{x}_0] \in \mathbb{R}^{N\times K}$, where $\odot$ denotes the Hadamard product and $\boldsymbol{x}_0 \in \mathbb{R}^N$ is the vector of grid positions, where its i^{th} row corresponds to position of the i^{th} grid. Accordingly, $\widetilde{\boldsymbol{X}} = [\widetilde{\boldsymbol{x}}_1, \widetilde{\boldsymbol{x}}_2, \cdots, \widetilde{\boldsymbol{x}}_K] \in \mathbb{R}^{N\times K}$ is a matrix of the parameter/time-varying grid point positions, and similarly $\widetilde{\boldsymbol{x}}_i \in \mathbb{R}^N$ is the vector of grid positions at the i^{th} time-level, where

$$\widetilde{\boldsymbol{x}}_i := \boldsymbol{x}_0 + \Delta\widetilde{\boldsymbol{x}}_i \in \mathbb{R}^N, \tag{2.11}$$

where $\boldsymbol{x}_0$ is the vector of constant grid positions, and $\Delta\widetilde{\boldsymbol{x}}_i$ is the distance between the constant grid position and the time-varying grid at the i^{th} time-level. Equation (2.11) in the matrix notations is rewritten as,

$$\widetilde{\boldsymbol{X}} := \boldsymbol{X} + \boldsymbol{\Delta}\widetilde{\boldsymbol{X}} \in \mathbb{R}^{N\times K}, \tag{2.12}$$

and is illustrated in Fig. 2.2.

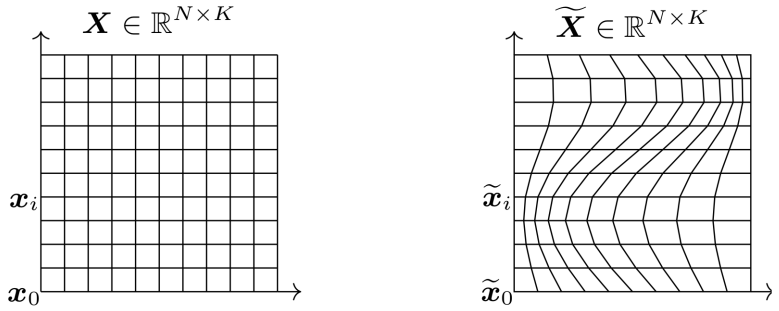


Figure 2.2: Illustration of the time-constant grid, $\boldsymbol{X}$, and the time-varying grid, $\widetilde{\boldsymbol{X}}$.

Moreover, $\boldsymbol{M} = [\boldsymbol{m}_1, \boldsymbol{m}_2, \cdots, \boldsymbol{m}_K] \in \mathbb{R}^{N\times K}$ and $\widetilde{\boldsymbol{M}} = [\widetilde{\boldsymbol{m}}_1, \widetilde{\boldsymbol{m}}_2, \cdots, \widetilde{\boldsymbol{m}}_K] \in \mathbb{R}^{N\times K}$ are the snapshots of the state parameters on $\boldsymbol{X}$ and $\widetilde{\boldsymbol{X}}$, respectively. Consequently, the map between the state space and the identified manifold is represented by an interpolation scheme between these two sets of grids. The map $\mathcal{G}(.)$, interpolates the snapshots stated on the constant grid onto the parameter/time-varying grid. Similarly, the map $\mathcal{G}^{-1}(.)$, interpolates the snapshots stated on the parameter/time-varying grid onto the constant grid. These operators/maps are presented in Alg. 1 and Alg. 2. To summarize:

- $\mathcal{G}(.)$ interpolates $\boldsymbol{M}$ stated on $\boldsymbol{X}$ onto $\widetilde{\boldsymbol{X}}$ and results in $\mathcal{G}(\boldsymbol{M})$, and

- $\mathcal{G}^{-1}(.)$ interpolates $\widetilde{\boldsymbol{M}}$ stated on $\widetilde{\boldsymbol{X}}$ on to $\boldsymbol{X}$ and results in $\mathcal{G}^{-1}\left(\widetilde{\boldsymbol{M}}\right)$.

Algorithm 1 The map from the constant grid to the parameter/time-varying grid, $\mathcal{G}(.)$

Input: The constant grid ($\boldsymbol{X} \in \mathbb{R}^{N\times K}$),
The parameter/time-varying grid ($\widetilde{\boldsymbol{X}} \in \mathbb{R}^{N\times K}$),
The snapshots of the state variables on the constant grid ($\boldsymbol{M} \in \mathbb{R}^{N\times K}$),
Output:The snapshots of the state variables on the parameter/time-varying grid ($\widetilde{\boldsymbol{M}} \in \mathbb{R}^{N\times K}$)

1: **for** $i = 1, 2, 3, \cdots, K$ **do**
2: $\quad\widetilde{\boldsymbol{m}}_i \leftarrow$ interpolate $\boldsymbol{m}_i$ stated on $\boldsymbol{x}_i$ to $\widetilde{\boldsymbol{x}}_i$ // Using the interpolation scheme of choice
3: **end for**

Algorithm 2 The map from the parameter/time-varying grid to the constant grid to , $\mathcal{G}^{-1}(.)$

Input: The constant grid ($\boldsymbol{X} \in \mathbb{R}^{N\times K}$),
The parameter/time-varying grid ($\widetilde{\boldsymbol{X}} \in \mathbb{R}^{N\times K}$),
The snapshots of the state variables on the parameter/time-varying grid ($\widetilde{\boldsymbol{M}} \in \mathbb{R}^{N\times K}$),
Output:The snapshots of the state variables on the constant grid ($\boldsymbol{M} \in \mathbb{R}^{N\times K}$)

1: **for** $i = 1, 2, 3, \cdots, K$ **do**
2: $\quad\boldsymbol{m}_i \leftarrow$ interpolate $\widetilde{\boldsymbol{m}}_i$ stated on $\widetilde{\boldsymbol{x}}_i$ to $\boldsymbol{x}_i$ // Using the interpolation scheme of choice
3: **end for**

Finally, the proposed method in (2.9) can be interpreted as a registration task, that minimizes the Kolmogorov n-width of the snapshots of the latent variables, $\mathcal{G}(\boldsymbol{M})$, on the learned parameter/time-varying grid.[3]

Remark 1 As an extension of Alg. 1 and Alg. 2, the loop/march in the column space can be substituted with multi-directional interpolation schemes, e.g. bi-linear or tri-linear.[4] Constructing a surrogate model and evaluating the model at the target grid points is another applicable and straightforward generalization. Considering the relative computational costs of these interpolation methods, any of the commonly used and off-the-shelf schemes can be utilized in the proposed framework.

We impose diffeomorphism as a condition on the mapping to and from the learned manifold: By definition, a map, $\mathcal{G}(.)$, is said to be diffeomorphic if $\mathcal{G}(.)$ and $\mathcal{G}^{-1}(.)$ are differentiable [114]. Bijectivity (i.e. one to oneness) and smoothness guarantee diffeomorphism. Therefore, by enforcing the map to be diffeomorphic, we ensure existence and uniqueness of $\widetilde{\boldsymbol{M}}$, given $\boldsymbol{M}$ and vice versa. Bijectivity is achieved by ensuring that volume of all the cells remain strictly positive as a constraint

[3]Image registration is the task of finding point-wise correspondences and their transformations between a set of images and a *template* or *atlas* image.

[4]Bi-linear for a one-dimensional in space and time and tri-linear for a two-dimensional in space and time.

in (2.9). A negative cell volume leads to the indeterminate derivative of the state parameter, which can be seen as a "tear" in an image. Smoothness of the grid is maintained by penalizing the abrupt changes of the grid volume, both in space and parameter/time.

2.2.2 Low-rank registration

In the this section, the construction of the grid in the parameter/time space is discussed. There are two general approaches to formulate the grid deformation in a registration problem. In the first class of approaches, the grid nodes are controlled as the solution of the a minimization problem. Diffeomorphism can then be achieved by enforcing a constraint on the determinant of the deformation gradient: be strictly positive for all grid cells. This approach leads to a high-dimensional optimization problem which its nonlinearity and ill-posedness makes it computationally challenging [115]. In the second class of approaches, the mapping is the solution of a transport equation, i.e. flow fields, as in diffeomorphic dimensionality reduction [92]. Interestingly, in some special cases, a similar transport equation arises where the frame of references is changed from the Eulerian to the Lagrangian viewpoint, i.e. by solving the hyperbolic PDEs on the corresponding characteristic lines. This change of the reference is proven to be efficient in reduced order modeling of convection dominated PDEs [39, 116, 117], also see appendix §A. In [39, 116, 117], it is shown that a *low-rank* grid can efficiently reduce the dimensionality of the snapshots of convection dominated flows featuring a large Kolmogorov n-width. We leverage this premise in a data-driven setting.

Recall (2.11) and (2.12) for the time-varying grid, and incorporate the assumption that the time-varying grid is low-rank, i.e.

$$\widetilde{\boldsymbol{X}} := \boldsymbol{X} + \boldsymbol{U}_x \boldsymbol{V}_x \in \mathbb{R}^{N \times K}, \tag{2.13}$$

where $\boldsymbol{U}_x \boldsymbol{V}_x \in \mathbb{R}^{N \times K}$ is a rank-r matrix, where $\boldsymbol{U}_x \in \mathbb{R}^{N \times r}$ and $\boldsymbol{V}_x \in \mathbb{R}^{r \times K}$. In its vector form $\widetilde{\boldsymbol{x}}_i \in \mathbb{R}^N$ is constructed as

$$\widetilde{\boldsymbol{x}}_i := \boldsymbol{x}_0 + \boldsymbol{U}_x \boldsymbol{v}_{xi}, \tag{2.14}$$

where $\boldsymbol{x}_0$ is the constant grid, $\boldsymbol{v}_{xi} \in \mathbb{R}^r$ is the i^{th} column of $\boldsymbol{V}_x = [\boldsymbol{v}_{x0}, \boldsymbol{v}_{x1}, \cdots, \boldsymbol{v}_{xK}]$. In summary, $\widetilde{\boldsymbol{X}} \in \mathbb{R}^{N \times K}$ defines the evolution of the parameter/time-varying grid on which the low-dimensional

latent variables lie and is illustrated in Fig. 2.3.

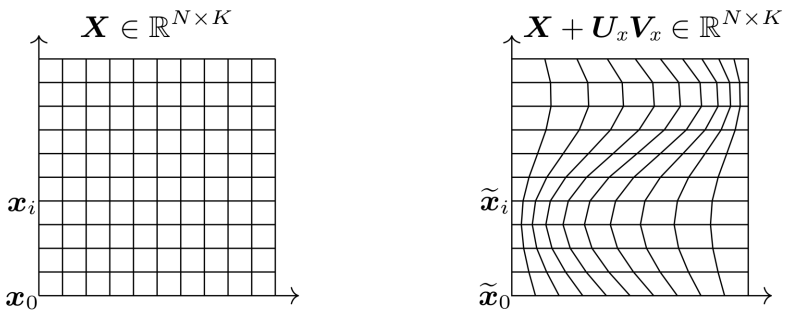


Figure 2.3: Illustration of the time-constant grid, $\boldsymbol{X}$, and the low-rank time-varying grid, $\widetilde{\boldsymbol{X}} := \boldsymbol{X} + \boldsymbol{U}_x\boldsymbol{V}_x$.

The rank-r grid is then the solution of the proposed manifold learning problem. The latent variables can be interpreted as the evolution of the state parameters on the low-rank approximation of the path on which the information travels, i.e. low-rank approximation of the characteristic lines of the hyperbolic PDEs. This is one of the key elements of the proposed method, which greatly reduces the size of the optimization compared to existing registration-based methods, such as [99]. More importantly, by incorporating the underlying physics of the nonlinearities of the hyperbolic PDEs, unprecedented predictive capabilities beyond the training range are achieved. The existence of a low-rank near-optimal grid for many of the convection-dominated PDEs are demonstrated in §5; However, the extension of this change of frame for any arbitrary and stochastic systems is not straightforward.

2.2.3 Implementation

In this section, we summarize the elements of the proposed algorithm and make some clarifications on implementation of the method. As illustrated in Fig. 2.4, the procedure is designed to identify a low-rank grid, $\widetilde{\boldsymbol{X}} = \boldsymbol{X} + \boldsymbol{U}_x\boldsymbol{V}_x$, on which the snapshot of the mapped data, $\widetilde{\boldsymbol{M}} = \mathcal{G}(\boldsymbol{M})$, is low-rank. The final minimization problem has the following form:

$$\begin{aligned} \underset{\boldsymbol{U},\,\boldsymbol{V},\,\boldsymbol{U}_x,\,\boldsymbol{V}_x}{\text{minimize}} \quad & \left\|\boldsymbol{M} - \mathcal{G}^{-1}(\boldsymbol{U}\boldsymbol{V})\right\|_F + \|\boldsymbol{\Gamma}_1\boldsymbol{U}_x\|_F + \left\|\boldsymbol{V}_x\,\boldsymbol{\Gamma}_2{}^T\right\|_F, \\ \text{subject to} \quad & \boldsymbol{v}_n \geq v_{\min}, \forall n \in \{1,\ldots,K\}, \\ & \widetilde{\boldsymbol{x}}_n|_{\partial\Omega} = \boldsymbol{x}|_{\partial\Omega}, \forall n \in \{1,\ldots,K\}, \end{aligned} \tag{2.15}$$

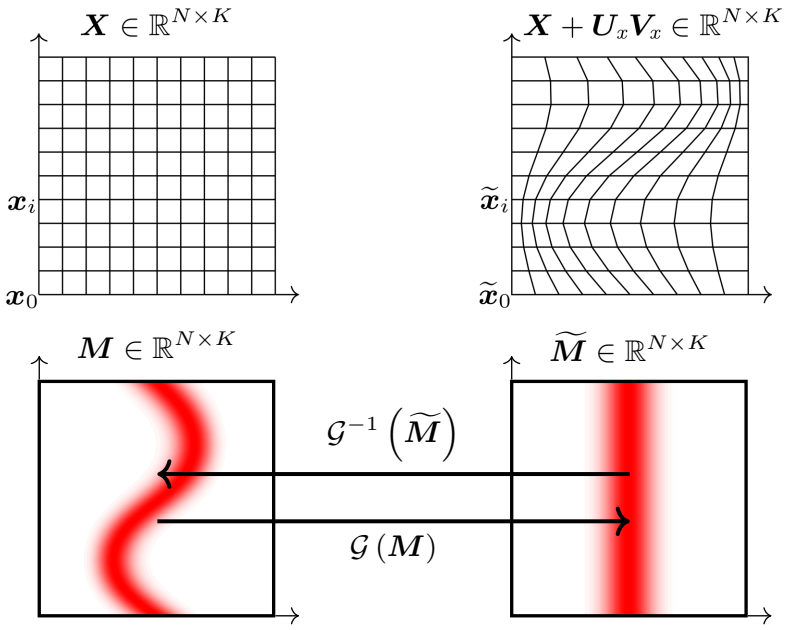


Figure 2.4: Illustration of proposed new nonlinear dimensionality reduction approach.

where $\mathcal{G}^{-1}(.)$ interpolates the low-rank mapped snapshots, $\boldsymbol{UV}$, stated on a low-rank parameter/time-varying grid, $\widetilde{\boldsymbol{X}}$, to the constant grid, $\boldsymbol{X}$, i.e. $\mathcal{G}^{-1}(.) : \widetilde{\boldsymbol{X}} := \boldsymbol{X} + \boldsymbol{U}_x\boldsymbol{V}_x \rightarrow \boldsymbol{X}$, and $\boldsymbol{\Gamma}_1 \in \mathbb{R}^{N\times N}$ and $\boldsymbol{\Gamma}_2 \in \mathbb{R}^{K\times K}$ are Tikhonov matrices designed to promote grid smoothness. Also $\boldsymbol{v}_n \in \mathbb{R}^{N-1}$ is a vector of cell volumes of the parameter/time-varying grid at the n^{th} parameter/time step, $v_{\min}$ is the minimum admissible cell volume, and $\widetilde{\boldsymbol{x}}_n|_{\partial\Omega}$ and $\boldsymbol{x}|_{\partial\Omega}$ are boundary points of the learned grid and the constant grid, respectively. The minimum cell volume is a hyper-parameter and is problem dependent.

The Tikhonov matrices are defined as $\boldsymbol{\Gamma}_1 = \gamma_x\boldsymbol{D}_{xx}$ and $\boldsymbol{\Gamma}_2 = \gamma_t\boldsymbol{D}_{tt}$, where $\boldsymbol{D}_{xx}$ and $\boldsymbol{D}_{tt}$ are the second derivative matrices in the spatial and parameter space, respectively. This choice penalizes any abrupt changes in the spacing of the grid in space and time/parameter space, respectively. The appropriate Tikhonov matrices are hyper-parameters and problem dependent. Constructing the reconstruction error versus norm of the Tikhonov terms and locating the L-shaped *corner* of the graph provides a systematic approach to choose γ_x and γ_t [118].

Moreover, for practical reasons, a weak constraint on the rank reduction is chosen. While (2.9) implies minimizing over the rank of $\widetilde{\boldsymbol{M}}$, in many cases, the solution of the minimization for a preset size of the decomposition is preferred. We assume $\widetilde{\boldsymbol{M}} = \boldsymbol{UV}$, where $\boldsymbol{U} \in \mathbb{R}^{N\times k_r}$, $\boldsymbol{V} \in \mathbb{R}^{k_r\times K}$, and $k_r \ll N$, $k_r \ll K$. Moreover, to reduce the size of the optimization problem, $\boldsymbol{U}_x$ and $\boldsymbol{V}_x$ are

uniformly down-sampled, however, the objective is evaluated on the fine grid.

To interpolate the snapshots between the two sets of grid, we simply utilize a p-degree polynomial interpolation scheme. This choice innately incorporates a sparsity pattern into the mapping. While the latent space representation of a data-point on the identified grid using a nearest-neighbor interpolation only requires one data-point on the constant grid, a p-degree polynomial interpolation requires $p-1$ entries of the input vector. This, in principle, leads to a great reduction in the size of the optimization problem compared to the traditional neural networks, where there is no *a priori* assumptions on the structure of the neurons' connectivity.

Algorithm 3 summarizes the proposed low-rank registration-based auto-encoder.

Algorithm 3 Training of the low-rank registration-based auto-encoder

Input: Hyper-parameters:
$\boldsymbol{\Gamma}_1$, $\boldsymbol{\Gamma}_2$,
Minimum admissible grid volume ($v_{\min}$),
Reduction parameters:
Rank of the parameter/time-varying grid (r),
Rank of the low-dimensional representation (k_r),
The snapshots matrix ($\boldsymbol{M} \in \mathbb{R}^{N\times K}$),
The constant grid ($\boldsymbol{X} = \boldsymbol{x}_0 \odot \boldsymbol{1} \in \mathbb{R}^{N\times K}$),
Maximum number of iterations ($j_{\max}$),

Output: Parameter/time-varying grid and its low-rank decomposition $\widetilde{\boldsymbol{X}}$,
The corresponding maps, i.e. $\mathcal{G}(.) : \boldsymbol{X} \to \widetilde{\boldsymbol{X}}$ and $\mathcal{G}^{-1}(.) : \widetilde{\boldsymbol{X}} \to \boldsymbol{X}$

Initialize the time-varying grid, i.e. $\widetilde{\boldsymbol{X}}^{(0)} = \boldsymbol{X} + \boldsymbol{U}_x^{(0)}\boldsymbol{V}_x^{(0)}$, with $\boldsymbol{U}_x^{(0)} \in \mathbb{R}^{N\times r}$ and $\boldsymbol{V}_x^{(0)} \in \mathbb{R}^{r\times K}$ using the SVD decomposition of the constant grid, $\boldsymbol{X}$, plus a small random perturbation
$j \leftarrow 0$
while $j \le j_{\max}$ **do**
  $\widetilde{\boldsymbol{M}} \leftarrow \mathcal{G}(\boldsymbol{M})$ // Interpolate the snapshots, $\boldsymbol{M}$, onto $\widetilde{\boldsymbol{X}}^{(j)}$
  $\boldsymbol{U}\boldsymbol{V} \approx \widetilde{\boldsymbol{M}}$ s.t. rank$(\boldsymbol{U}\boldsymbol{V}) = k_r$ // Approximate $\widetilde{\boldsymbol{M}}$ using its SVD as in (2.10)
  $\widetilde{\boldsymbol{M}} \leftarrow \mathcal{G}^{-1}(\boldsymbol{U}\boldsymbol{V})$ // Interpolate $\boldsymbol{U}\boldsymbol{V}$ onto the constant grid, $\boldsymbol{X}$
  $\mathcal{J} = \left\|\boldsymbol{M} - \widetilde{\boldsymbol{M}}\right\|_F + \left\|\boldsymbol{\Gamma}_1\boldsymbol{U}_x^{(j)}\right\|_F + \left\|\boldsymbol{V}_x^{(j)}{\boldsymbol{\Gamma}_2}^T\right\|_F$ // Evaluate the objective
  Update $\boldsymbol{U}_x^{(j)}$ and $\boldsymbol{V}_x^{(j)}$ minimizing $\mathcal{J}$ // Update the grid bases via the rule of the optimization
  $\widetilde{\boldsymbol{X}}^{(j+1)} = \boldsymbol{X}^{(j)} + \boldsymbol{U}_x^{(j)}\boldsymbol{V}_x^{(j)}$ // Update the grid using the grid bases
  $j \leftarrow j + 1$
end while

2.2.4 Alternative interpretations of the proposed method

We have formulated the proposed method as an image registration problem, here we pose the method from other perspectives.

- **Change of frame of reference**

  The method is based on changing the frame of reference from a stationary observer to an arbitrarily moving observer, whose path can be characterized by a low-rank decomposition. The proposed method generates two sets of bases: firstly, a rank-k_r linear decomposition of the snapshots from the moving observer viewpoint: $\boldsymbol{UV}$, and secondly, a rank-r linear decomposition of the path of that observer: $\widetilde{\boldsymbol{X}} = \boldsymbol{X} + \boldsymbol{U}_x\boldsymbol{V}_x$. The observers moves such that the flow field from its viewpoint, i.e. an arbitrary Lagrangian-Eulerian (ALE) framework, is low rank. Note that for the moving observer, both sets of bases are time-invariant. While in the stationary framework, the observer identifies a single, but *time-varying* set of bases that optimally factorizes the solution. This interpretation is the foundation of the projection-based ROMs constructed in §3.2.1.

- **Manifold learning and approximation theory**

  The proposed approach identifies a nonlinear manifold characterized by a low-rank grid on which the Kolmogorov n-width of the solution is minimal.

- **Low-rank registration-based auto-encoder**

  The proposed approach can be viewed as an auto-encoder layer, where the parameterization of the activation functions is tied to a computational grid. In this interpretation, the latent variables – the code as defined in (2.6)– can be efficiently approximated in the linear POD subspace. Moreover, the activation function is a nonlinear function based on an interpolation scheme and parameter/time-varying grid. Moreover, the grid is identifiable on a rank-r space.

  The encoder function interpolates the data from the Eulerian grid onto the ALE grid using any of the arbitrary interpolation schemes, i.e. $\mathcal{G}^{-1}\left(\boldsymbol{w}[n]\right)$. The latent variables are therefore representation of the data on the time-varying grid. Subsequently, the latent representation is decoded to the Eulerian grid using an interpolation scheme, $\mathcal{G}^{-1}\left(\mathcal{G}\left(\boldsymbol{w}[n]\right)\right)$.

  Auto-encoders are seen as a dimensionality reduction if the dimension of the coded layer, k, is less than the dimension of the input layer, N, hence the name *bottleneck* [111]. However, there are use cases of *over-complete* auto-encoders, where $k > N$. The output of $\mathcal{G}\left(\boldsymbol{w}[n]\right)$, our proposed auto-encoder, is of the same dimension of the input vector, $k = N$. The

dimensionality reduction step of the approach is achieved using PCA/POD of the latent state. The major benefit of the proposed approach compared to the traditional nonlinear activation function is its physical interpretability. The traditional auto-encoders tend to detect edges at different positions and orientations of the input snapshot, without any recognizable structure. In the proposed approach, however, the latent parameters represent the variables on a low-rank time-varying frame, which in many cases, corresponds to a low-rank approximation of the characteristics lines of the hyperbolic PDEs (see 2.2.2).

In (2.15), both the high-dimensional snapshots and its low-dimensional representation on the learned manifold (latent variables) are stored in a matrix. In contrast to most machine vision tasks dealing with images, the PDEs are not necessarily solved on a uniform Cartesian grid. Many of the PDEs are discretized on unstructured computational grids, while many of the traditional machine learning tools are developed for uniform Cartesian grids representing an image. Often, the snapshot matrix of $\boldsymbol{M}$ is defined on a constant grid (Eulerian framework) and $\widetilde{\boldsymbol{M}}$, by construct, is associated to the snapshots of latent variables on a parameter/time-varying grid (ALE framework). The convolutional auto-encoders are successfully adopted to learn the low-dimensional features of fluids systems on such grids [79, 93], however, the extension of the method to unstructured grids requires exploring other less known approaches capable of handling irregular connectivities [119]. The proposed auto-encoder is, in principle, oblivious to the type of the computational grid and the associated connectivity.

Chapter 3

Reduced order models on the identified manifold

In §2, the first step of the model reduction procedure, i.e. identification of an efficient manifold, was discussed and developed. In this section, we construct the reduced order model on the identified subspace/manifold and evolve the model in time. We review the projection-based model order reduction with POD bases and a completely data-driven model order reduction architecture based on artificial neural networks (ANNs).

3.1 Preliminaries

3.1.1 Projection-based model order reduction

Consider a first order dynamical system in form of

$$\frac{d\boldsymbol{w}}{dt} = \boldsymbol{f}\left(\boldsymbol{w}\right), \tag{3.1}$$

where $t \in [0, T]$ denotes the range of time, and $\boldsymbol{f} : \mathbb{R}^N \to \mathbb{R}^N$ is a nonlinear function, and $\boldsymbol{w} = \boldsymbol{w}\left(t\right) \in \mathbb{R}^N$ is a vector of the state variables. The system is defined on the space $x \in \Omega$, Ω being the domain equipped with appropriate boundary conditions at $\partial\Omega$ and initial condition, $\boldsymbol{w}[0]$. The domain is discretized uniformly in space on a stationary Eulerian grid $\boldsymbol{x} = [x_1, \ldots, x_{N_x}]^T$ using standard techniques such as finite-volume or finite-elements.

Without any loss of generality, it is assumed throughout the remainder of this paper that (3.1) is discretized in time using an implicit linear multi-step scheme. Hence, if $t[0] = 0 < t[1] < \cdots < t[N_t] = T$ denotes a discretization of the time-interval $[0, T]$, where $t[n] = n\Delta t$, $n \in \{1, \cdots, N_t\}$, the

discrete counterpart of (3.1) at time-step n is

$$\boldsymbol{R}\left(\boldsymbol{w}[n]\right) = \sum_{j=0}^{s} \alpha_j \boldsymbol{w}[n-j] + \sum_{j=0}^{s} \beta_j \boldsymbol{f}(\boldsymbol{w}[n-j]) = 0 \tag{3.2}$$

where s is the order of accuracy of the chosen time-integrator and α_j and β_j are the two constants characterizing it.

In traditional projection-based MOR where the solution is approximated by a POD subspace as the global trial,

$$\boldsymbol{w}[n] \approx \widetilde{\boldsymbol{w}}[n] = \boldsymbol{w}[0] + \boldsymbol{U}\boldsymbol{w}_r[n], \tag{3.3}$$

where the columns of $\boldsymbol{U} \in \mathbb{R}^{N\times k}$ contain the bases for this subspace, and $\boldsymbol{w}_r[n] \in \mathbb{R}^k$ denotes the temporal coefficients of the bases at the corresponding time step. As extensively discussed in §2, the success of the ROM is based on the accuracy of the approximation (3.3), where $k \ll N$. Substituting (3.3) into (3.2) and projecting it onto the POD subspace, $\boldsymbol{U} \in \mathbb{R}^{N\times k}$, yields the projection-based reduced order model

$$\boldsymbol{U}^T \boldsymbol{R}\left(\boldsymbol{w}[0] + \boldsymbol{U}\boldsymbol{w}_r[n]\right) = \boldsymbol{0}. \tag{3.4}$$

3.1.2 Artificial neural network-based model order reduction

Any dynamical system of (3.1) or (3.4) can be approximated using a slew of neural network architectures, for a comparison see [80]. Amongst the most promising architectures, Recurrent Neural Networks (RNNs) are specifically designed to represent sequential data. RNNs can be trained either on the physical or latent space realized by auto-encoders as discussed in §2.1.3. In this thesis, we concentrate on RNNs on the identified manifolds to construct neural network-based ROMs, where the hidden state at n^{th} time-step, $\boldsymbol{w}_r[n] \in \mathbb{R}^k$, is calculated in

$$\boldsymbol{w}_r[n] = f_{\text{RNN}}\left(\boldsymbol{W}_{rr}\boldsymbol{w}_r[n-1] + \boldsymbol{U}_{ru}\boldsymbol{u}[n] + \boldsymbol{b}_r\right), \tag{3.5}$$

as a function of the previous time-step hidden state ($\boldsymbol{w}_r[n-1]$) and the current time-step input ($\boldsymbol{u}[n] \in \mathbb{R}^N$) and biases ($\boldsymbol{b}_r \in \mathbb{R}^k$), where $\boldsymbol{W}_{rr} \in \mathbb{R}^{k\times k}$, and $\boldsymbol{U}_{ru} \in \mathbb{R}^{k\times N}$. The hidden state is then

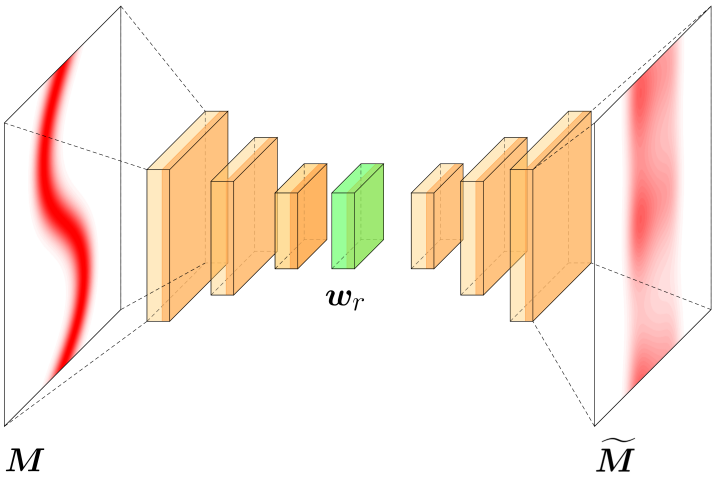


Figure 3.1: Traditional neural network architecture. The snapshots matrix, $\boldsymbol{M} \in \mathbb{R}^{N\times K}$, is coded to the latent layer, $\boldsymbol{w}_r \in \mathbb{R}^{k\times K}$, and then is decoded to the low-rank representation as $\widetilde{\boldsymbol{M}} \in \mathbb{R}^{N\times K}$. The RNN cells, f_{RNN}, are trained on the latent layer.

mapped to the n-dimensional space,

$$\boldsymbol{w}[n] = \sigma_{\text{dec}}\left(\boldsymbol{W}_{wr}\boldsymbol{w}_r[n] + \boldsymbol{b}_w\right), \tag{3.6}$$

where σ_{dec} is the activation function that maps the hidden state to the physical high-dimensional space, where $\boldsymbol{W}_{wr} \in \mathbb{R}^{N\times k}$ and $\boldsymbol{b}_w \in \mathbb{R}^N$. The process of training an RNN is to find the weight matrices and biases that minimizes a cost function, a distance between the solution of the HFM and the output of the RNN, $\forall n \in \{1, N_t\}$. Vanishing or exploding gradient leads poor training of an RNNs with significant long-term dependencies. Long short-term memory (LSTM) cells [120], and the gated recurrent unit (GRU) [121] can often address these issues. In the case where $k \ll N$, the hidden states evolve in time on a low-dimension manifold. LSTMs are shown to outperform other architectures [80], and are used to approximate reduced order models in the present thesis, i.e.

$$\boldsymbol{w}_r[n] = f_{\text{LSTM}}\left(\boldsymbol{w}_r[n-1]\right). \tag{3.7}$$

We employ the LSTM cells to approximate the PDEs on the learned manifold as a proven architecture [122]. The network is trained on a learned manifold using a dense neural network (§2.1.3) as illustrated in Fig. 3.1.

3.2 Proposed Approach: Construction of ROMs on the identified manifold

In this section, the ROMs are constructed on the time-varying grid, i.e. the solution of the proposed manifold learning problem introduced in §2.2, from an ALE point of view (see §2.2.4).

3.2.1 ALE projection-based ROM

In this section, the proposed nonlinear dimensionality reduction procedure is used to construct the projection-based reduced ROMs of the governing equations of interest. Without loss of generality, we utilize the Eulerian interpretation of the proposed method for this purpose (see §2.2.4). That is, the flow field is approximated using time-varying global bases, $\boldsymbol{w}[n] = \boldsymbol{U}[n]\boldsymbol{w}_r[n]$, where $\mathcal{G}^{-1}(\boldsymbol{U}) = \{\boldsymbol{U}[0], \boldsymbol{U}[1], \ldots, \boldsymbol{U}[N_t]\}$. Under this interpretation, construction of ROMs proceeds in a manner very similar to traditional projection-based MOR techniques discussed in §3.1.1.

In the proposed new approach, only a single modification is required; i.e. replacing the time-invariant basis $\boldsymbol{U}$ with the time-varying one $\boldsymbol{U}[n]$ as follows

$$\boldsymbol{U}^T[n]\boldsymbol{R}\left(\boldsymbol{w}[0] + \boldsymbol{U}[n]\boldsymbol{w}_r[n]\right) = \boldsymbol{0}. \tag{3.8}$$

The stability of the resulted time-varying reduced order model of (3.8) remains an open question. Although it can be seen as a caveat of the proposed approach, such lack of stability guarantee is common in the nonlinear or even linear time-invariant ROMs, since only a small category of projections and class of equations preserve energy, e.g. [123, 124]. To address the issue, *a posteriori* stabilization methods are often used [81, 125, 126]. The goal of this class of methods is to minimally modify/rotate/augment the reduced system to enforce stability constraints. In principle, a similar approach can be utilized to stabilize projection-based ALE ROMs. We develop a stabilization method applicable to time-varying ROMs in §4.

3.2.2 Low-rank registration-based auto-encoder neural network-based ROM

In this section, the proposed nonlinear dimensionality reduction is incorporated as an auto-encoder layer in a neural network architecture. Addition of this layer, by definition, lowers the dimensionality

of the latent variable snapshots, therefore, it is expected to lead to faster training and lower reconstruction error.

Consider a traditional neural network architecture to approximate a dynamical system onto a low-dimensional space, as depicted in Fig. 3.1. An auto-encoder layer, or layers of a deep auto-encoder, are trained to lower the dimensionality of the snapshots (see §2.1.3). The RNN/LSTM layers are then trained to evolve the dynamical system in time (see §3.1.2). For an unforced dynamical system a one-to-many network is trained. The initial condition of the dynamical system is the only input of the network and $\boldsymbol{u}[n] = \boldsymbol{0}, \forall n \in \{0, \cdots, N_t\}$ in (3.5) or (3.7). To summarize the training, similar to (2.7):

$$\underset{\boldsymbol{\theta}_{\text{enc}},\, \boldsymbol{\theta}_{\text{dec}}}{\text{minimize}} \quad \sum_{n=1}^{N_t} \mathcal{L}\left(\boldsymbol{w}[n], f_{\text{dec}}\left(f_{\text{enc}}\left(\boldsymbol{w}[n]; \boldsymbol{\theta}_{\text{enc}}\right); \boldsymbol{\theta}_{\text{dec}}\right)\right), \tag{3.9}$$

where $\boldsymbol{w}_r[n] = f_{\text{enc}}\left(\boldsymbol{w}[n]; \boldsymbol{\theta}_{\text{enc}}\right)$, and

$$\boldsymbol{w}_r[n] = f_{\text{RNN}}\left(\boldsymbol{w}_r[n-1]\right). \tag{3.10}$$

We propose the network architecture as depicted in Fig. 3.2, where the network is initialized with the low-rank registration-based auto-encoder (§2.2). The cost function is then re-written:

$$\underset{\boldsymbol{\theta}_{\text{enc}},\, \boldsymbol{\theta}_{\text{dec}}}{\text{minimize}} \quad \sum_{n=1}^{N_t} \mathcal{L}\left(\widetilde{\boldsymbol{w}}[n], f_{\text{dec}}\left(f_{\text{enc}}\left(\widetilde{\boldsymbol{w}}[n]; \boldsymbol{\theta}_{\text{enc}}\right); \boldsymbol{\theta}_{\text{dec}}\right)\right), \tag{3.11}$$

where $\widetilde{\boldsymbol{w}}[n]$ is the n^{th} column of $\widetilde{\boldsymbol{M}} = \mathcal{G}\left(\boldsymbol{M}\right)$, $\boldsymbol{w}_r[n] = f_{\text{enc}}\left(\widetilde{\boldsymbol{w}}[n]; \boldsymbol{\theta}_{\text{enc}}\right)$, and

$$\boldsymbol{w}_r[n] = f_{\text{RNN}}\left(\boldsymbol{w}_r[n-1]\right). \tag{3.12}$$

The additional low-rank registration-based auto-encoder layer, improves the accuracy and training costs of the neural network-based models by reducing the dimensionality of the latent variable. The proposed network with the addition of the low-rank registration-based auto-encoder layer is illustrated in Fig. 3.2.

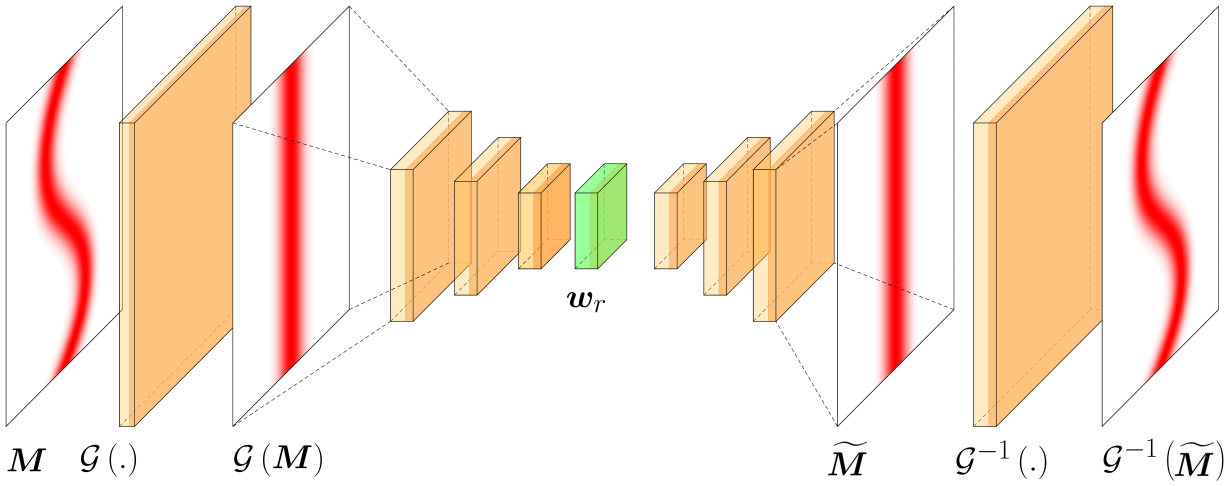


Figure 3.2: Neural network architecture equipped with the proposed low-rank registration-based auto-encoder layer. The proposed auto-encoder of $\mathcal{G}(.)$ maps the snapshots matrix of $\boldsymbol{M} \in \mathbb{R}^{N\times K}$ onto the learned manifold, i.e. $\widetilde{\boldsymbol{M}} = \mathcal{G}(\boldsymbol{M}) \in \mathbb{R}^{N\times K}$. The mapped snapshot of $\widetilde{\boldsymbol{M}}$ is coded to the latent layer, $\boldsymbol{w}_r \in \mathbb{R}^{k_r\times K}$, and then is decoded to the low-rank representation as $\widetilde{\boldsymbol{M}} \in \mathbb{R}^{N\times K}$, and finally is mapped to the physical space by $\mathcal{G}^{-1}\left(\widetilde{\boldsymbol{M}}\right) \in \mathbb{R}^{N\times K}$. The RNN cells, f_{RNN}, are trained on the latent layer.

3.2.3 Predictive ROMs

For a ROM to be truly useful, it must be capable of predicting, sufficiently accurately, new solutions outside of the training regime. For unsteady problems, the prediction regime usually refers to time intervals beyond the interval used to generate the snapshots.

It is well known that traditional ROMs perform relatively poorly in this predictive regime. The extension of the proposed ROMs, either projection-based or neural network-based, to enable predictive simulation beyond the final time step of the snapshots is quite straightforward. Considering that the learned manifold is described by a low-rank grid (§2.2.2), the temporal basis of the grid, $\boldsymbol{V}_x$, is extrapolated in time. In absence of any *a priori* knowledge of the system, a linear extrapolation is chosen.

Chapter 4

Stabilization of time-varying ROMs

In this section, we develop our new approach to stabilize LTV ROMs.[1] The goal is to design an input matrix and the associated control signal. This should not be confused with an optimal control problem, where the goal is to determine the control signal to minimize some performance criterion [128], e.g. flow control [129–131]. The proposed ROM stabilization method is motivated by pole placement or spectral assignment approaches, which belong to a class of methods called feedback stabilization methods [132]. The application of pole placement and eigenvalue reassignment is first introduced in context of stabilization of LTI ROMs [81].

In section §4.1, the fundamentals of the stability and stabilization of linear systems and ROMs are reviewed. The proposed methodology of designing a feedback controller appropriate for time-varying ROMs is motivated and formulated mathematically as a nonlinear constrained minimization problem in §4.2. The evaluation of the performance of the proposed method is postponed to §5.3.

4.1 Preliminaries

It is well known that projection-based ROMs do not, in general, inherit the stability properties of the corresponding HFMs. That is, even when the snapshots used to generate the bases arise from solutions of stable HFMs, there is no guarantee that the resulting ROMs will also be stable. Dimensionality reduction in these methods is based on discarding the less important modes of the solution space. The importance is often measured by energy content of the corresponding bases and does not take into account the underlying dynamics. This might lead to instability of the reduced order models, e.g. by truncating the dynamics corresponding to dissipation of the energy by high frequency, low-energy coherent structures in a turbulent flow. Stability of a ROM is a function of the

[1] Some of the material presented in this section and §5.3 also appears in [127] and is used following the Wiley's licensing and copyright guidelines.

number of bases retained [36, 125, 133], the particular dynamical system under consideration [133], and the method used to develop the ROM.

Balanced truncation of linear time-invariant systems preserve the stability of the resulted ROMs *a priori* [58, 134]. In this method, controllability and observability Gramians are computed, which is a very costly procedure especially for large-scale HFMs [123]. Balanced truncation requires solving matrix equations. Developing efficient approximate numerical algorithms for large matrix equations remains an active area of research [135–139]. To avoid the high cost of balanced truncation, POD based [41–47] and balanced POD (BPOD) [56, 57] reduction methods are often used. These approximate methods however, in general lack the stability guarantee of balanced truncation; although a variety of heuristics methods are available to improve the stability properties of the ROMs. To name a few, Amsallem and Farhat [140] show that the POD-based ROMs originating from the descriptor form of the linearized CFD equations tend to be more stable compared to the non-descriptor form in certain applications, or Rowley et al. [123] prove that an "energy-based" inner product preserves the stability of an equilibrium point in symmetric linear-time invariant (LTI) systems.

In the other perspective, stability is achieved *a posteriori* by modifying or controlling the identified ROM equations. Examples of such approaches include adding a pressure term to model the truncated scales [141] and including an additional constraint on the resolved kinetic energy [36]. In LTI ROMs, where the unstable modes are accessible, it is straightforward to reassign the eigenvalues of the reduced system matrix to the stable region and achieve stability with minimal effect on accuracy [81]. Similarly for the ROMs of compressible Navier-Stokes equations, comprising of time-invariant linear and quadratic nonlinear terms, the realized reduced subspace can be minimally rotated to achieve stability [142]. In an other attempt a margin of stability is achieved by minimally modifying the test subspace to satisfy Lyapunov equations for LTIs [125]. More recently, a hybrid approach is proposed that combines eigenvalues reassignment and modification of the test subspace, showing to be more robust in a test case of linearized Euler equations when the number of unstable eigenvalues is large [126].

Many of the reduction methods can be extended to the reduction of time-varying systems, for a review see [143] and the references therein. Balanced truncation methods, with *a priori* stability guarantee, are generalized to linear time-varying systems [144–147]. The cost of balanced truncation

for time-varying systems is even more prohibitive compared to LTIs, considering that observability and controllability Gramians as well as projection of the system matrices are computed at each time step [147]. To the best of our knowledge, there are no methods specifically developed to stabilize linear time-varying ROMs that scale to large system.

In this section, we propose an *a posteriori* method to stabilize linear time-varying ROMs. The approach is an extension on eigenvalue reassignment method of LTIs in [81] for more general cases of linear time-varying ROMs. Time-varying ROMs may originate from approximations of time-varying dynamical systems or arise as byproducts of time-varying bases, see §4.1.2. In our method developed in this thesis, we propose a feedback controller with minimal penalty on accuracy of the ROM. The function of the feedback controllers are not as straightforward as reassigning the unstable eigenvalues of one single system matrix to the stable half plane (inside unit circle for a discrete ROM) as it is in an LTI. More general criteria are taken into account as nonlinear constraints of an optimization problem to guarantee stability. The size of the optimization problem is reduced by a sporadic activation of the controller, and only controlling the most energy growing modes of the system. The method is non-intrusive, i.e. it does not require access to the HFM solver or system of equations, since it operates on reduced order system matrices directly. Therefore, the proposed method applies to reduced systems of any construct lacking *a priori* stability guarantee, e.g. POD, Krylov-based, balanced-POD, moment matching.

In §4.1.1, the stability of time-varying linear systems is reviewed. In §4.1.2, the standard projection-based model order reduction generating linear time-varying ROMs is reviewed. The projection-based ROM using POD bases is discussed, although the stabilization method proposed in this thesis is not restricted to this particular choice. The eigenvalue reassignment method of stabilizing linear time-invariant ROMs, as the foundation of the proposed approach, is summarized in §4.1.3.

4.1.1 Linear systems and stability

Consider a continuous time, linear time-invariant (CLTI) system

$$\dot{\boldsymbol{w}} = \boldsymbol{A}\boldsymbol{w} + \boldsymbol{B}\boldsymbol{u}, \tag{4.1}$$

where $\boldsymbol{A} \in \mathbb{R}^{N \times N}$ is the system (or state) matrix. The is system is exponentially stable if and only if all the eigenvalues of the system matrix have negative real parts, i.e. $\forall i,\ \mathrm{Re}\left\{\lambda_i\left(\boldsymbol{A}\right)\right\} \leq 0$ [148].

Either by discretizing the continuous system of (4.1) or by the modeling of a discrete phenomenon, a discrete time, linear time-invariant (DLTI) system is

$$\boldsymbol{w}[n+1] = \boldsymbol{A}\boldsymbol{w}[n], \tag{4.2}$$

where $\boldsymbol{A}$ is the system (or state) matrix. The is system is exponentially stable if and only if all the eigenvalues of the system matrix lie within the unit circle [148], i.e. $\forall i, |\lambda_i\left(\boldsymbol{A}\right)| \leq 1$.

Consider a time-discrete linear time-varying system, as the HFM,

$$\begin{aligned} \boldsymbol{w}[n+1] &= \boldsymbol{A}[n]\boldsymbol{w}[n] + \boldsymbol{B}[n]\boldsymbol{u}[n], \\ \boldsymbol{y}[n] &= \boldsymbol{C}[n]\boldsymbol{w}[n], \end{aligned} \tag{4.3}$$

where $\boldsymbol{A}[n] \in \mathbb{R}^{N \times N}$, $\boldsymbol{B}[n] \in \mathbb{R}^{N \times P}$, and $\boldsymbol{C}[n] \in \mathbb{R}^{Q \times N}$ are system (or state), input, and output matrices, $\boldsymbol{w}[n] \in \mathbb{R}^{N}$ is the state vector, $\boldsymbol{u}[n] \in \mathbb{R}^{P}$ is the control vector, and $\boldsymbol{y}[n] \in \mathbb{R}^{Q}$ is the output vector, at the n^{th} time step.

The system in the general form of (4.3), is either a representation of an inherently discrete dynamical system or discretization of (linear or linearized) system of PDEs in space and time.

The solution of (4.3) can be presented as summation of the zero-input, and the zero-state response of the system, i.e.

$$\boldsymbol{w}[n] = \boldsymbol{\phi}_{\boldsymbol{A}}[n, 0]\boldsymbol{w}[0] + \sum_{l=0}^{n-1} \boldsymbol{\phi}_{\boldsymbol{A}}[n, l]\boldsymbol{B}[l]\boldsymbol{u}[l], \tag{4.4}$$

where the state transition matrix ($\boldsymbol{\phi}_{\boldsymbol{A}}[n, n_0]$), by definition, relates the state of the autonomous system at time n to the state at an earlier time step n_0,

$$\boldsymbol{\phi}_{\boldsymbol{A}}[n, n_0] = \begin{cases} \boldsymbol{I} & n = n_0, \\ \boldsymbol{A}[n-1]\boldsymbol{A}[n-2]\cdots\boldsymbol{A}[n_0] & n > n_0. \end{cases} \tag{4.5}$$

Stability of linear time varying systems is a classic problem [149, 150] with ongoing interest [151–155]. However, it has only more recently gained attention in the model order reduction

community [145, 156–160]. We discuss some of the fundamentals of stability of such systems to the extents required to discuss the contributions of the present work.

For a TV system to be exponentially stable, in addition to satisfying the stability, the rate of change of the variation of $\boldsymbol{A}[n]$ also needs to be "slow enough" [149, 161]. Sufficient conditions for eigenvalues and upper bounds on rate of change of the system can be found in Ilchmann et al. [162]. However, for general DLTVs there is no obvious relationship between stability and the eigenvalue locations of the system matrices [148]. Only in the case of periodic systems, using Floquet-Lyapunov theory or its discrete-time counterpart [163], the stability can be studied by an equivalent LTI system [164]. In general, there are three families of criteria to assess the stability of LTVs: state transition matrix based, Lyapunov based, and methods based on poles and zeros [165].

The following theorem provides upper and lower bounds for the state parameter growth or decay of a DLTV [148]. Rugh [148] proves the following upper and lower bounds on rate of growth/decay of the states of the system. For the (zero-input) discrete-time LTV denote the largest and smallest point-wise eigenvalues of $\boldsymbol{A}^T[n]\boldsymbol{A}[n]$ by $\lambda_{max}[n]$ and $\lambda_{min}[n]$. Then for any $\boldsymbol{w}[0]$ and n the solution of (4.3) satisfies

$$\prod_{j=0}^{n-1}\sqrt{\lambda_{min}[j]} \leq \frac{\|\boldsymbol{w}[n]\|}{\|\boldsymbol{w}[0]\|} \leq \prod_{j=0}^{n-1}\sqrt{\lambda_{max}[j]},\ n \geq 0. \tag{4.6}$$

Note that the singular values of $\boldsymbol{A}[n]$ are the square roots of the eigenvalues of $\boldsymbol{A}^T[n]\boldsymbol{A}[n]$ [166]. The computation of singular-values of $\boldsymbol{A}[n]$ is numerically more stable than computation of eigenvalues of $\boldsymbol{A}^T[n]\boldsymbol{A}[n]$ [167], therefore we replace the square root of the eigenvalues in (4.6) with the corresponding singular-values.

In the case of discrete time, time-periodic systems, with period of p, i.e. $\boldsymbol{A}[n+p]{=}\boldsymbol{A}[n]$. The LTP is stable if and only if modulus of all the eigenvalues of $\boldsymbol{\Pi} = \boldsymbol{A}[p]\cdots\boldsymbol{A}[1]$, lie within the unit circle.

4.1.2 Time-varying ROMs

In the traditional projection-based model order reduction approach, the solution is approximated on a global trial subspace

$$\boldsymbol{w}[n] \approx \boldsymbol{U}\boldsymbol{w}_r[n], \tag{4.7}$$

where the columns of $\boldsymbol{U} \in \mathbb{R}^{N \times k}$ contain the bases for this subspace, and $\boldsymbol{w}_r[n] \in \mathbb{R}^k$ denotes the temporal coefficients of the bases at the corresponding time step, n. In the case that the state parameter can be represented using a small number of bases, $k \ll N$, the dimension of the model is significantly reduced.

In general, $\boldsymbol{U}$ can be computed using a number of different approaches. In spectral methods, analytical functions such as trigonometric functions or Chebyshev polynomials are often used [168]. In the context of data-driven ROMs, the bases are usually extracted from snapshots of the solution, $\boldsymbol{M}$, capturing some major characteristics of the solution, e.g. POD captures the energy content [47], DMD approximates the best linear operator fitting consecutive-time of data vectors [169] and optimally time-dependent modes capture directions of transient instabilities [170]. POD bases, as a common choice of reduced bases with a broad range of applications, is briefly discussed in §2.1.2. It is emphasized that the methods developed here are not restricted to the choice of the bases and can be applied to any linear/linearized reduced order system of equations.

The separation of the spatial and temporal bases, as in (4.7), is a common construct of the bases in projection-based model order reduction. However, this assumption can be relaxed introducing a time-varying subspace,

$$\boldsymbol{w}[n] \approx \boldsymbol{U}[n]\boldsymbol{w}_r[n], \tag{4.8}$$

as in recent literature in regard to parametric time-varying systems [171], a time-varying shift of the reduced systems [26], or reduced description of transient chaotic systems by definition of optimally time-dependent modes [78].

In the present thesis, we are especially interested in the time-varying ROMs as the result of time-varying bases since the projection-based ROMs on the time-varying grid introduced in §3.2.1 results in time-varying bases, i.e.

$$\boldsymbol{U}[n] = \mathcal{G}^{-1}\left(\boldsymbol{U}\right), \tag{4.9}$$

where $\mathcal{G}^{-1}\left(.\right)$ maps the bases from the time-varying grid at n^{th} time step to a constant grid. Time-dependent bases also naturally arise by a change of reference framework as in [26, 60, 64].

Substituting (4.8) in the HFM of (4.3), and projecting the equation onto a test subspace, $\boldsymbol{U}[n]$,

leads to a time-varying reduced order system of equation

$$\begin{aligned} \boldsymbol{w}_r[n+1] &= \boldsymbol{A}_r[n]\boldsymbol{w}_r[n] + \boldsymbol{B}_r[n]\boldsymbol{u}[n], \\ \boldsymbol{y}_r[n] &= \boldsymbol{C}_r[n]\boldsymbol{w}_r[n], \end{aligned} \tag{4.10}$$

where

$$\begin{aligned} \boldsymbol{A}_r[n] &= \left(\boldsymbol{U}^T[n+1]\boldsymbol{U}[n+1]\right)^{-1}\boldsymbol{U}^T[n+1]\boldsymbol{A}[n]\boldsymbol{U}[n] \in \mathbb{R}^{k\times k}, \\ \boldsymbol{B}_r[n] &= \left(\boldsymbol{U}^T[n+1]\boldsymbol{U}[n+1]\right)^{-1}\boldsymbol{U}^T[n+1]\boldsymbol{B}[n] \in \mathbb{R}^{k\times P}, \\ \boldsymbol{C}_r[n] &= \boldsymbol{C}[n]\boldsymbol{U}[n] \in \mathbb{R}^{Q\times P}. \end{aligned} \tag{4.11}$$

In the case of Galerkin projection, $\boldsymbol{U}[n] = \boldsymbol{U}[n]$.

4.1.3 LTI ROM stabilization via eigenvalue reassignment

To help motivate the approach for time-varying system, we begin by first considering the time-invariant case. Consider a linear time-invariant reduced order system of the form

$$\begin{aligned} \boldsymbol{w}_r[n+1] &= \boldsymbol{A}_r\boldsymbol{w}_r[n] + \boldsymbol{B}_r\boldsymbol{u}[n], \\ \boldsymbol{y}_r[n] &= \boldsymbol{C}_r\boldsymbol{w}_r[n], \end{aligned} \tag{4.12}$$

where $\boldsymbol{w}_r[n] \in \mathbb{R}^k$ is the state vector, $\boldsymbol{u}[n] \in \mathbb{R}^P$ is the control vector. $\boldsymbol{A}_r \in \mathbb{R}^{k\times k}$, $\boldsymbol{B}_r \in \mathbb{R}^{k\times P}$, and $\boldsymbol{C}_r \in \mathbb{R}^{Q\times k}$ are constant matrices of system (or state), input, and output. In the pole placement approach, the system is augmented by a controller and an input, $\boldsymbol{B}_c\boldsymbol{u}_c[n]$, i.e.

$$\boldsymbol{w}_r[n+1] = \boldsymbol{A}_r\boldsymbol{w}_r[n] + \boldsymbol{B}_r\boldsymbol{u}[n] + \boldsymbol{B}_c\boldsymbol{u}_c[n]. \tag{4.13}$$

Assuming a full-state feedback of the form $\boldsymbol{u}_c[n] = -\boldsymbol{K}\boldsymbol{w}_r[n]$, where $\boldsymbol{K} \in \mathbb{R}^{P\times k}$ is the feedback or gain matrix,

$$\boldsymbol{w}_r[n+1] = \left(\boldsymbol{A}_r - \boldsymbol{B}_c\boldsymbol{K}\right)\boldsymbol{w}_r[n] + \boldsymbol{B}_r\boldsymbol{u}[n]. \tag{4.14}$$

The closed-loop controlled system of form (4.14) is in the form of (4.12), where the stabilized/controlled system matrix, $\hat{\boldsymbol{A}}_r$, is defined as

$$\hat{\boldsymbol{A}}_r := \boldsymbol{A}_r - \boldsymbol{B}_c\boldsymbol{K}, \tag{4.15}$$

and $\boldsymbol{v}[n]$ is now the updated input, and we refer to the output of the stabilized ROM as $\hat{\boldsymbol{y}}_r[n]$.

The pole placement control problem determines $\boldsymbol{K}$ to achieve the desired poles/eigenvalues of the closed loop system [132]. The locations of the poles and the quality of choice of the feedback are measured by the "tracking performance". In other words, a tracking problem is a stabilization problem with the goal of minimizing the deviation of the current states from the desired trajectory [172]. In the ROM framework the tracking performance is defined using the ROM error, i.e. the goal of the controller is to minimize the distance between the stable HFM and the stabilized ROM;

$$\text{minimize} \quad \sum_{n=0}^{N_t} \|\boldsymbol{y}[n] - \hat{\boldsymbol{y}}_r[n]\|_2 \,, \tag{4.16}$$

where $\|.\|_2$ denotes the $\mathcal{L}_2$ -norm of a vector.

In the traditional viewpoint of controller design and its applications to physical systems, there are limitations in the choice of the controller in addition to the feasible input rate or magnitude and actuators' saturation. Considering our goal of stabilization of a reduced order model, the design of $\boldsymbol{B}_c$ and $\boldsymbol{K}$ is in the algebraic level. Control and feedback matrices can be arbitrarily chosen, given the pair $(\boldsymbol{A}_r, \boldsymbol{B}_c)$ is stabilizable (or controllable in a more strict sense).

The optimization problem of (4.16) minimizes the ROM output error via transformation of the reduced system matrices. The eigenvalues of LTI systems dictates the decay/growth rate of the response, therefore by placing the eigenvalues of $\hat{\boldsymbol{A}}_r$ in the unit circle, or in the left half plane in the continuous cases, the system can be stabilized. The algorithms developed by Kalashnikova et al. [81] minimize the ROM error by placing the unstable eigenvalues in the stable region in continuous systems. Reformulation of the algorithm to a discrete system is trivial, i.e.

$$\begin{aligned} \underset{\lambda_i}{\text{minimize}} \quad & \sum_{n=0}^{N_t} \|\boldsymbol{y}[n] - \hat{\boldsymbol{y}}_r[n]\|_2 \,, \\ \text{subject to} \quad & 1.\ \ \lambda_i \in \mathbb{C}, \\ & 2.\ \ |\lambda_i| \leq 1, \ \text{for } i \in \{1, \cdots, L \leq k\} \,, \end{aligned} \tag{4.17}$$

where λ_i are the unstable complex conjugate pairs and real eigenvalues of the state matrix of reduced and controlled system, and L is the number of the unstable eigenvalues of the original ROM of size k.

4.2 Proposed approach: LTV ROM stabilization via feedback controllers

To extend the pole-placement method summarized in the previous section to LTV systems, the control input matrices are assumed to be time-dependent [132], i.e.

$$\boldsymbol{w}_r[n+1] = \boldsymbol{A}_r[n]\boldsymbol{w}_r[n] + \boldsymbol{B}_r[n]\boldsymbol{u}[n] + \boldsymbol{B}_c[n]\boldsymbol{u}_c[n], \tag{4.18}$$

and subsequently the feedback matrices are chosen to be time-varying,

$$\boldsymbol{w}_r[n+1] = \left(\boldsymbol{A}_r[n] - \boldsymbol{B}_c[n]\boldsymbol{K}[n]\right)\boldsymbol{w}_r[n] + \boldsymbol{B}_r[n]\boldsymbol{u}[n]. \tag{4.19}$$

The extension of the approach to the time-varying ROMs is challenging due to two main factors. Firstly, in the LTI case, reassigning the eigenvalues of the system matrix to the stable region leads to a stable ROM; However, the location of the point-wise eigenvalues in the LTV ROMs does not provide sufficient condition for stability [173]. Secondly, in LTI case, the rate of growth/decay of the state parameter is directly bounded by magnitude of the eigenvalues; However, in LTV ROMs, the stability depends on the point-wise eigenvalues of ${\boldsymbol{A}_r}^T[n]\boldsymbol{A}_r[n]$, i.e. to the largest and smallest eigenvalues of point-wise gram matrices.

In the following, we discuss the choice of input and gain matrix design as well as the condition for the stability to equip the trajectory error minimization of (4.16). Consider

$$\begin{aligned} \boldsymbol{B}_c[n] &= \boldsymbol{U}[n], \\ \boldsymbol{K}[n] &= \boldsymbol{\Sigma}_c[n]\boldsymbol{V}^T[n], \end{aligned} \tag{4.20}$$

where $\boldsymbol{U}[n]$ and $\boldsymbol{V}[n]$ are left and right singular vectors of $\boldsymbol{A}_r[n]$ and $\boldsymbol{\Sigma}_c[n]$ is a real, positive, diagonal matrix, i.e. $\boldsymbol{\Sigma}_c[n] = \mathrm{diag}\left(\sigma_1[n], \sigma_2[n], \cdots, \sigma_k[n]\right)$, where $\sigma_1[n] \geq \sigma_2[n] \geq \cdots \geq \sigma_k[n]$. Substituting (4.20) in (4.19)

$$\boldsymbol{w}_r[n+1] = \hat{\boldsymbol{A}}_r[n]\boldsymbol{w}_r[n] + \boldsymbol{B}_r[n]\boldsymbol{u}[n], \tag{4.21}$$

where the controlled/stabilized reduced system matrix at the n^{th} time-step, $\hat{\boldsymbol{A}}_r[n]$, is defined as

$$\hat{\boldsymbol{A}}_r[n] = \boldsymbol{U}[n]\boldsymbol{\Sigma}^{\boldsymbol{\Delta}}[n]\boldsymbol{V}^T[n] := \boldsymbol{U}[n]\left(\boldsymbol{\Sigma}[n] - \boldsymbol{\Sigma}_c[n]\right)\boldsymbol{V}^T[n]. \tag{4.22}$$

Therefore, the choice of the input and gain matrices as in (4.20) can be interpreted as calibration of the singular values of point-wise system matrices and provides direct control on the upper and lower bounds of the growth/decay of the state parameter.

The final minimization problem has the following form:

$$\begin{aligned} \underset{\sigma_i[n]}{\text{minimize}} \quad & \sum_{n=0}^{N_t} \left\|\boldsymbol{y}[n] - \hat{\boldsymbol{y}}_r[n]\right\|_2, \\ \text{subject to} \quad & 1.\ \sigma_i^{\Delta}[n] \in \mathbb{R}^+, \text{ for } n \in \{0, \cdots, N_t\}, \\ & 2.\ \sup_{n \geq n_0} \left\|\boldsymbol{\phi}_{\hat{\boldsymbol{A}}_r}[n, n_0]\right\|_2 \leq c[n_0], \end{aligned} \tag{4.23}$$

where $\sigma_i^{\Delta}[n]$ are the singular values of $\boldsymbol{\Sigma}^{\boldsymbol{\Delta}}[n]$.

For a discrete-time linear time-periodic system (DLTP), with period of p, i.e.

$$\boldsymbol{A}_r[n] = \boldsymbol{A}_r[n+p], \tag{4.24}$$

the stability is dictated by eigenvalues of the monodromy matrix, i.e. $\boldsymbol{\Pi}_r = \boldsymbol{\phi}_{\boldsymbol{A}_r}[p, n_0]$ [164]. Therefore, in a linear time-periodic system, the optimization problem of (4.23) reads

$$\begin{aligned} \underset{\sigma_i[n]}{\text{minimize}} \quad & \sum_{n=0}^{N_t} \left\|\boldsymbol{y}[n] - \hat{\boldsymbol{y}}_r[n]\right\|_2, \\ \text{subject to} \quad & 1.\ \sigma_i^{\Delta}[n] \in \mathbb{R}^+, \text{ for } n \in \{0, \cdots, p\}, \\ & 2.\ |\lambda_i| \leq 1, \text{ for } i \in \{1, \cdots, k\}, \end{aligned} \tag{4.25}$$

where $\sigma_i^{\Delta}[n]$ are the singular values of $\boldsymbol{\Sigma}^{\boldsymbol{\Delta}}[n]$ and λ_i are the eigenvalues of the monodromy matrix of reduced and controlled system, $\hat{\boldsymbol{\Pi}}_r[n]$. In the special case of discrete-time linear time-invariant

systems, the time-varying controller of (4.20) reduces to

$$\begin{aligned} \boldsymbol{B}_c &= \boldsymbol{U}, \\ \boldsymbol{K} &= \boldsymbol{\Sigma}_c \boldsymbol{V}^T, \end{aligned} \tag{4.26}$$

and the optimization problem of (4.23) reads

$$\begin{aligned} \underset{\sigma_i}{\text{minimize}} \quad & \sum_{n=0}^{N_t} \|\boldsymbol{y}[n] - \hat{\boldsymbol{y}}_r[n]\|_2 \,, \\ \text{subject to} \quad & 1.\ \sigma_i^{\Delta} \in \mathbb{R}^+, \\ & 2.\ |\lambda_i| \leq 1, \ \text{for } i \in \{1, \cdots, k\} \,, \end{aligned} \tag{4.27}$$

where σ_i^{Δ} are the singular values of $\boldsymbol{\Sigma}^{\boldsymbol{\Delta}}$ and λ_i are the eigenvalues of the state matrix of reduced and controlled system, $\hat{\boldsymbol{A}}_r$.

4.2.1 Solution of constrained optimization problem

As proposed so far, the size of the optimization problems is kN_t, leading to a costly optimization problem for large systems, especially in numerical simulation of the discretized PDEs where a fine temporal discretization is required for the stability and the accuracy of HFMs. To address this problem, two components are introduced to this algorithm. Firstly, since the designed controller is not necessarily required to be applied at all point-wise matrices, a "sporadic feedback control" is incorporated. In this approach, the controller is sporadically and sparsely applied in time, i.e. control one out of every Δn point-wise matrices. Secondly, the controller is applied only to the first few most energy growing modes of the system. In the case of time-varying systems, since the rate of the growth is bounded by the product of the largest singular values, it is sufficient to control only the first few largest singular values of the system matrices. Consider the l largest singular values of the point-wise system matrix at the n^{th} time-step, i.e. $\sigma_1[n] \geq \sigma_2[n] \geq \cdots \geq \sigma_l[n]$, then $\boldsymbol{\Sigma}_c[n]$ of (4.20) is redefined as

$$\boldsymbol{\Sigma}_c[n] = \left[\begin{array}{c|c} \boldsymbol{\Sigma_l}[n] & \boldsymbol{0} \\ \hline \boldsymbol{0} & \boldsymbol{0} \end{array}\right] \in \mathbb{R}^{k \times k}, \tag{4.28}$$

where $\boldsymbol{\Sigma_l}[n] \in \mathbb{R}^{l \times l}$ is a diagonal matrix of the first l leading singular values.

Although this problem is non-convex, it is noteworthy that the proposed optimization problem is guaranteed to be feasible for a small enough choice of Δn and a large enough choice of l. By selecting $l = k$ and $\Delta n = 1$, all the singular values of all the point-wise system matrices are included in the optimization, and by setting $\mathbf{\Sigma}_c[n] = \mathbf{\Sigma}[n],\ \forall n \in \{0, 1, 2, \cdots, N_t\}$ the system is dead-beat controlled, i.e. the state of the system is brought to zero.

It is well known that optimizing over singular values is challenging due to the high likelihood of local minima. Our numerical experiments confirm that this is also an issue for the stabilization approach proposed here. To address this issue, the optimization problem is solved using the multi-start global optimization algorithm; implementation details are summarized in the results section of this thesis. It is emphasized that – although a global optimization is required in our approach – the size of the optimization depends on the size of the ROM, and not the original HFM. Thus, in multi-query applications - such as design, optimization, or real-time embedded applications - the relatively high costs of the off-line stabilization stage can can be justified by the savings during the on-line stage. Finally, although a simple multi-start global search is proposed here, some further computational speedups can be expected through the use of a more sophisticated global optimization algorithm.

The proposed algorithm is summarized in Alg. 4. An upper bound on the cost of each iteration of the proposed optimization problem is $\mathcal{O}\left(k^3 M^3\right)$ and thus it is bounded cubically with the dimension k of the LTV ROM and the number of time varying system matrices, M. For a general time-varying system $M = N_t$, and for time-periodic and time-invariant systems, $M = p$ and $M = 1$, respectively. Additionally, by introducing the sporadic control (Δn) of the l leading singular values of the reduced point-wise system matrices ($\mathbf{\Sigma_l}[n]$), the cost of the stabilization method is reduced to $\mathcal{O}\left(l^3 {N_t}^3/\Delta n^3\right)$. The cost of all other steps in Alg. 4 are independent of the dimension of the HFM. The cost of the proposed algorithm is therefore significantly lower than competing approaches. For example, the balanced truncation [147] approach scales as $\mathcal{O}\left(N^3 M\right)$, where N is the size of the HFM, making it prohibitive for realistic large-scale problems.

Finally, the choice of l or Δn is a compromise between ROM performance and available off-line computational resources.

Algorithm 4 Stabilization of linear time-varying/time-periodic/time-invariant ROMs

Input: $M = \begin{cases} N_t, & \text{for a linear time-varying ROM with } N_t \text{ time steps,} \\ p, & \text{for a linear time-periodic ROM with period } p, \\ 1, & \text{for a linear time-invariant ROM,} \end{cases}$

The reduced point-wise matrices (i.e. $\boldsymbol{A}_r[n]$, $\boldsymbol{B}_r[n]$, $\boldsymbol{C}_r[n]$ $\forall n \in \{0, \cdots, M\}$),

The number of time steps (N_t),

Output of the high-fidelity model (i.e. $\boldsymbol{y}[n]$ $\forall n \in \{0, 1, 2, \cdots, N_t\}$)

The rank of the stabilizer/controller matrix (l),

The temporal sparsity in the activation of the controller matrix (Δn),

The number of the starting points in the search loop (j_{max})

Output: The stabilized reduced system matrices, $\hat{\boldsymbol{A}}_r[n]$ $\forall n \in \{0, \cdots, M\}$

The stabilizer/controller matrices of singular values, $\boldsymbol{\Sigma}_c[n]$ $\forall n \in \{0, \Delta n, 2\Delta n, \cdots\}$

Compute SVD of $\boldsymbol{A}_r[n]$, i.e. $\boldsymbol{U}[n]\boldsymbol{\Sigma}[n]\boldsymbol{V}^T[n]$ $\forall n \in \{0, \cdots, M\}$

Initialize the controlled/stabilized reduced matrices, $\hat{\boldsymbol{A}}_r[n]$, with $\boldsymbol{A}_r[n]$ $\forall n \in \{0, \cdots, M\}$

$j \leftarrow 0$

while $j \leq j_{max}$ **do** // Multi-start search loop

  Initialize $\boldsymbol{\Sigma}_c[n]$ $\forall n \in \{0, \Delta n, 2\Delta n, \cdots\}$ as defined in (4.28)

  Solve the constrained optimization problem of (4.23), (4.25), or (4.27) // Local optimization algorithm

  Construct the controlled reduced matrices as in (4.22), $\hat{\boldsymbol{A}}_r[n]$ $\forall n \in \{0, \cdots, M\}$

  Integrate numerically the stabilized ROM, i.e. $\hat{\boldsymbol{y}}_r[n]$ $\forall n \in \{0, 1, 2, \cdots, N_t\}$

  Calculate the ROM error, $\sum_{n=0}^{N_t} \|\boldsymbol{y}[n] - \hat{\boldsymbol{y}}_r[n]\|_2$

  Keep $\boldsymbol{\Sigma}_c[n]$ $\forall n \in \{0, \Delta n, 2\Delta n, \cdots\}$ and $\hat{\boldsymbol{A}}_r[n]$ $\forall n \in \{0, \cdots, M\}$ of the ROM with the lowest error

  $j \leftarrow j + 1$

end while

4.2.2 Stabilization in predictive regimes

Predictive regimes in context of ROM stabilization is for the ROM to be capable of predicting the system response, given system inputs and state parameters outside of those used during the off-line training stage. The introduced stabilization approach generates ROMs that are, in principle, capable of performing all such predictive simulations; and a large number of such simulations are illustrated in the results section of this thesis. Here, we briefly discuss some of the stability/accuracy guarantees, and lack thereof, associated with our method.

In the case of novel inputs and initial conditions, predictive ROM performance can be expected to be relatively high due to the fact that the proposed stabilization approach generates ROMs that are guaranteed stable for arbitrary input signals and initial conditions; see section §4.1.1. In the case of novel system state parameters, ROM stability cannot be guaranteed. Thus, predictive simulations featuring new system parameters can be expected to be more challenging. However, it is emphasized that most ROM methodologies in the literature also tend to lack out-of-the-box stability/accuracy guarantees. Rigorous robustness to parameter variations is an active area of research and it is realistic to expect many of the methods developed in this area to be applicable to our proposed approach. Such extensions, however, are beyond the scope of the present thesis.

Chapter 5

Numerical experiments

In this section, the methods developed in this thesis are evaluated using several numerical experiments.

In §5.1, several parameter-dependent and time-dependent problems are designed to feature traveling waves in different directions, shock formations and moving fronts, and nonlinear PDEs. The optimal manifolds and the corresponding time-varying spatial grids are identified using the low-rank registration-based auto-encoder developed in §2. The reconstruction mean squared error, as a measure of reducibility of the mapped snapshots on the identified manifold and the traditional POD subspace, are compared. In §5.2, the projection-based and neural network-based ROMs of the time-dependent problems of §5.1 are constructed using the methods discussed in §3. The ROMs on the POD subspace and the identified manifolds are compared and the enhancement in predictive capabilities of the ROMs on the identified manifolds are demonstrated. In §5.3, different projection-based ROMs, of time-invariant, time-periodic and time-varying systems are stabilized using the stabilization method proposed in §4. The robustness of the method with respect to parameter and input states is evaluated using an uncertainty quantification analysis.

5.1 Identification of the low-rank registration-based manifold

We provide several experiments to demonstrate the capabilities of the proposed low-rank registration-based auto-encoder. In this section, we solve for the low-rank grid and test the reducibility/compression of the snapshots. We attempt to demonstrate both compression and interpretability of our results in these experiments. These problems were selected to illustrate the unique capabilities of the proposed new approach: i.e. the compression efficiency, the ability to handle multiple waves traveling in multiple different directions and nonlinear PDEs. In all these experiments, we resort to

readily available optimization packages capable of solving optimization with nonlinear constraints; such as `minimize` in `scipy` for our Python implementation, or interior-point method in `fmincon` in our Matlab implementation. The gradients in the update rule of the optimization are calculated using finite-differences.

The mean squared error, ε, is defined as the measure of accuracy of the reconstructions, i.e.

$$\varepsilon = \frac{1}{N \times K} \sum_{i=1}^{N} \sum_{j=1}^{K} \left(w_{i,j} - \widetilde{w}_{i,j} \right)^2 , \tag{5.1}$$

where $w_{i,j}$ and $\widetilde{w}_{i,j}$ are the entries on the i^{th} row and j^{th} column of $\boldsymbol{M} \in \mathbb{R}^{N \times K}$ and $\widetilde{\boldsymbol{M}} \in \mathbb{R}^{N \times K}$, respectively. The relation between the mean squared error and the Frobenius norm error is straightforward.

5.1.1 Manifold learning in rotated character "A"

Consider a computer vision task of learning the nonlinear transformation, rotation, given a data-set comprised of a rotated character "A". The image of character "A" is stored in a 50×50 matrix and is rotated a total of 90 degrees with 3 degrees increments resulting in a snapshot matrix of dimension 2500×31. A representative sample of the snapshots is shown in Fig. 5.1a, and a single POD mode reconstruction is illustrated in Fig. 5.1b. In this problem, $\boldsymbol{U}_x$ is down-sampled to size of 7, i.e. the total of 49 control points. Moreover, $v_{\min} = 0$, $\boldsymbol{\Gamma}_1 = 100\boldsymbol{D}_{xx}$ and $\boldsymbol{\Gamma}_2 = (100/\pi)\, \boldsymbol{D}_{\theta\theta}$, where $\boldsymbol{D}_{xx}$ and $\boldsymbol{D}_{\theta\theta}$ are the second derivative matrices in the spatial and parameter space, respectively. The boundary point constraints are removed for this particular problem. The optimization problem of (2.15) approximates the rigid body rotation. In Fig. 5.1d and Fig. 5.1f, the snapshots are approximated using a single basis ($k_r = 1$) on the learned manifold of $r = 1$ and $r = 2$, respectively. The rank-1 grid, Fig. 5.1c, introduces a slight scaling and shear in the reconstruction, however, the character and its rotation is clearly realized (Fig. 5.1d). By increasing the rank of the grid to $r = 2$, Fig. 5.1e, the scaling artifact is removed from the reconstruction of the snapshots (Fig. 5.1f). The reconstruction delivered on the learned grid, using the proposed approach, is remarkably more accurate compared to the traditional POD approach.

(a) The snapshots.

(b) The rank-1 reconstruction of the snapshots ($k = 1$) on the POD subspace.

(c) The parameter-varying rank-1 grid corresponding to the identified manifold.

(d) The rank-1 reconstruction of the snapshots ($k_r = 1$) on the rank-1 grid ($r = 1$).

(e) The parameter-varying rank-2 grid corresponding to the identified manifold.

(f) The rank-1 reconstruction of the snapshots ($k_r = 1$) on the rank-2 grid ($r = 2$).

Figure 5.1: 90 degrees rotation of character "A".

5.1.2 Quasi-1D Euler flow in parameter-varying nozzles

Consider the non-linear, quasi-one-dimensional Euler equations modeling fluid flows in a variable-area stream tube,

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(\rho u) = -\frac{1}{A}\frac{dA}{dx}\rho u, \tag{5.2a}$$

$$\frac{\partial \rho u}{\partial t} + \frac{\partial}{\partial x}(\rho u^2 + p) = -\frac{1}{A}\frac{dA}{dx}\rho u^2, \tag{5.2b}$$

$$\frac{\partial \rho E}{\partial t} + \frac{\partial}{\partial x}([\rho E + p]u) = -\frac{1}{A}\frac{dA}{dx}(\rho E + p)u, \tag{5.2c}$$

with $A(x)$ in a finite domain $x \in [0, 10]$, where ρ is the fluid density, u is the fluid velocity, p is the thermodynamic pressure, and

$$\rho E = \rho e + \frac{1}{2}\rho u^2, \tag{5.3}$$

is the total energy density. The pressure is related to ρE by the equation of state

$$p = (\gamma - 1)\left(\rho E - \frac{1}{2}\rho u^2\right), \tag{5.4}$$

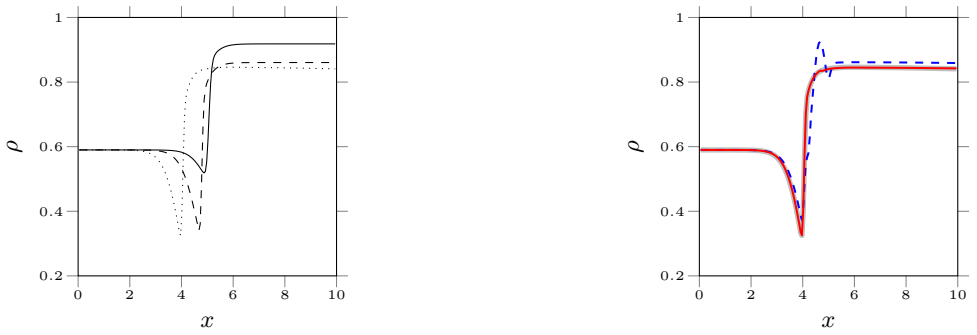
for a perfect gas with specific heat ratio of $\gamma = 1.4$. The $x = 0$ boundary models a reservoir with specified total stagnation pressure $p_t = 101325$ Pa, and stagnation temperature $T_t = 300$ K, while the right boundary at $x = 10$ enforces a specific static back pressure, $p_b = 73145$ Pa. The variable-area stream tube is defined as follows

$$A(x) = 1.398 + \mu \tanh(1.8(x - 5)), \tag{5.5}$$

where $\mu \in [0.1, 127]$.

Equation (5.2) is discretized using central finite differences and stabilized using a first-order artificial viscosity scheme. $N = 200$ grid points are used to discretized the domain $0 \leq x \leq 10$. The solution is marched to steady state using the implicit Euler time integration scheme.

Solution snapshots are computed using 10 instances of the parameter $\mu_i = 0.1 + 0.13(i - 1)$, for $i = \{1, \ldots, 10\}$. The manifold is learned by solving the low-rank optimization problem of Alg. 3. As in Fig. 5.2, the solution compressed with $k_r = 2$ on the learned manifold of a rank-2 grid is



(a) Snapshots (μ_1, μ_5, μ_{10}) **(b)** Reconstruction of μ_{10} snapshot

Figure 5.2: The density along a converging/diverging nozzle; (5.2a): some different cases (μ_1, μ_5, μ_{10}) comprising the snapshot matrix; (5.2b): comparison of snapshots (thick grey) and their reduced order representation on the POD subspace (dashed blue) and the learned manifold of $r = 2$ (solid red) for μ_{10} case.

indistinguishable from the high fidelity snapshots, while the $k = 2$ approximation on the linear POD subspace contains large amplitude oscillations in the vicinity of the shock.

5.1.3 Manifold learning in second-order wave equation

Consider the second-order wave equation in a single dimension,

$$\frac{\partial^2 w(x,t)}{\partial t^2} - \frac{\partial^2 w(x,t)}{\partial x^2} = 0, \tag{5.6}$$

in the domain $(x,t) \in [0,1] \times [0,1]$, with initial conditions $w_0 = e^{-((x-0.5)/0.05)^2}$, and $\partial w/\partial t(x,0) = \partial^2 w/\partial t^2(x,0) = 0$ and Dirichlet boundary conditions $w(0,t) = w(1,t) = 0$. The second-order order equation is chosen specifically to demonstrate the proposed method's ability to efficiently approximate multiple waves traveling in different directions; a phenomena which often presents a challenge to competing methodologies. Equation (5.6) is discretized using an implicit second-order discretization in time and second-order central discretization in space leading to a "4-level scheme", with second-order, central finite difference discretization of the second derivative on equidistant grid points of $N_x = 500$ and $\Delta t = 5 \times 10^{-4}$. A rank-2 time-varying grid ($r = 2$) is learned via (2.15) having set $k_r = 4$, and $\boldsymbol{\Gamma}_1 = 1.3 \times 10^{-3} \boldsymbol{D}_{xx}$ and $\boldsymbol{\Gamma}_2 = 1.3 \times 10^{-3} \boldsymbol{D}_{tt}$, where $\boldsymbol{D}_{xx}$ and $\boldsymbol{D}_{tt}$ are the second derivative matrices in space and time, respectively. The rank-2 time-varying grid ($r = 2$) representing the learned manifold is depicted in Fig. 5.3. For illustration purposes, only every 20^{th} grid points are plotted.

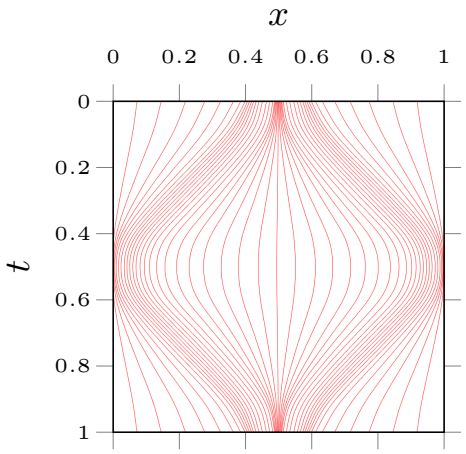
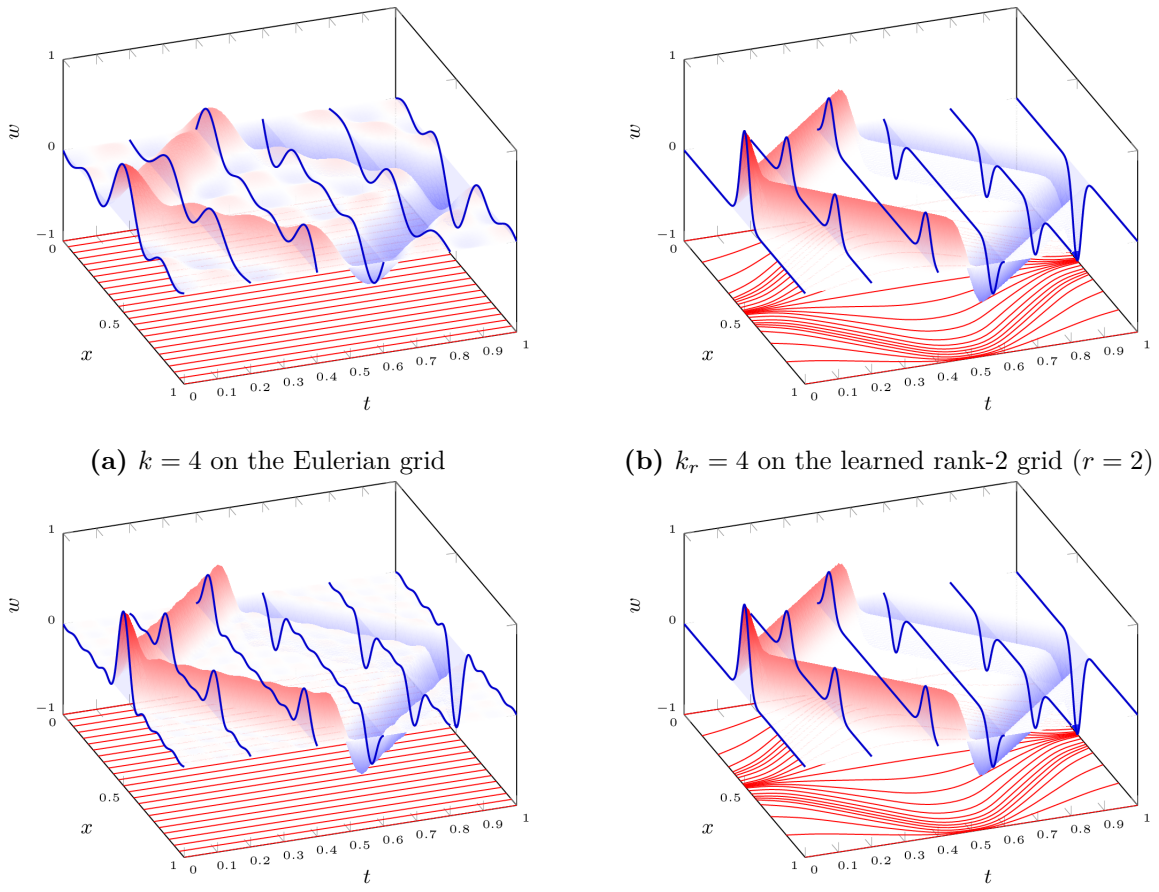


Figure 5.3: The time-varying rank-2 grid corresponding to the identified manifold of wave equation.

The rank-4 and rank-8 low-dimensional representation of snapshots are compared on the constant grid and the time-varying grid in Fig. 5.4. On the learned manifold, the reconstruction is closer to the snapshots and free of non-physical oscillations, the artifacts of discarding the low-energy/high-

frequency POD bases on the Eulerian grid. The error decreases by increasing the size of the reconstruction on the constant and time-varying grids and are compared in Fig. 5.5. In this particular problem, the reconstruction error at the training point ($k_r = 4$) is approximately 2 orders of magnitude lower than the reconstruction error on the POD subspace. By increasing the rank of the reconstructions, the error decreases. However, at $k = k_r \approx 22$, the reconstruction error on both manifolds are equal. This is mainly due to the error introduced by using the interpolation operator, $\mathcal{G}(.)$. Specifically, the aliasing caused by representing the bases with high frequency on a locally coarse grid, and therefore the introduced error only dominates at a relatively high-rank reconstruction.



(a) $k = 4$ on the Eulerian grid

(b) $k_r = 4$ on the learned rank-2 grid ($r = 2$)

(c) $k = 8$ on the Eulerian grid

(d) $k_r = 8$ on the learned rank-2 grid ($r = 2$)

Figure 5.4: The wave snapshots of the one-dimensional wave, on top of the computational grid representative of the corresponding manifolds.

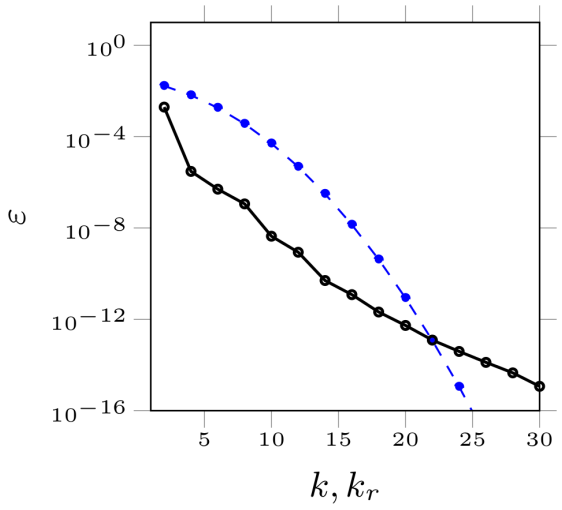


Figure 5.5: Reconstruction error in second-order wave problem. The rank-k reconstruction on POD subspace (dashed blue line), and rank-k_r reconstruction on the identified manifold of $r = 2$ (solid black line).

5.1.4 Manifold learning in one-dimensional Riemann problem

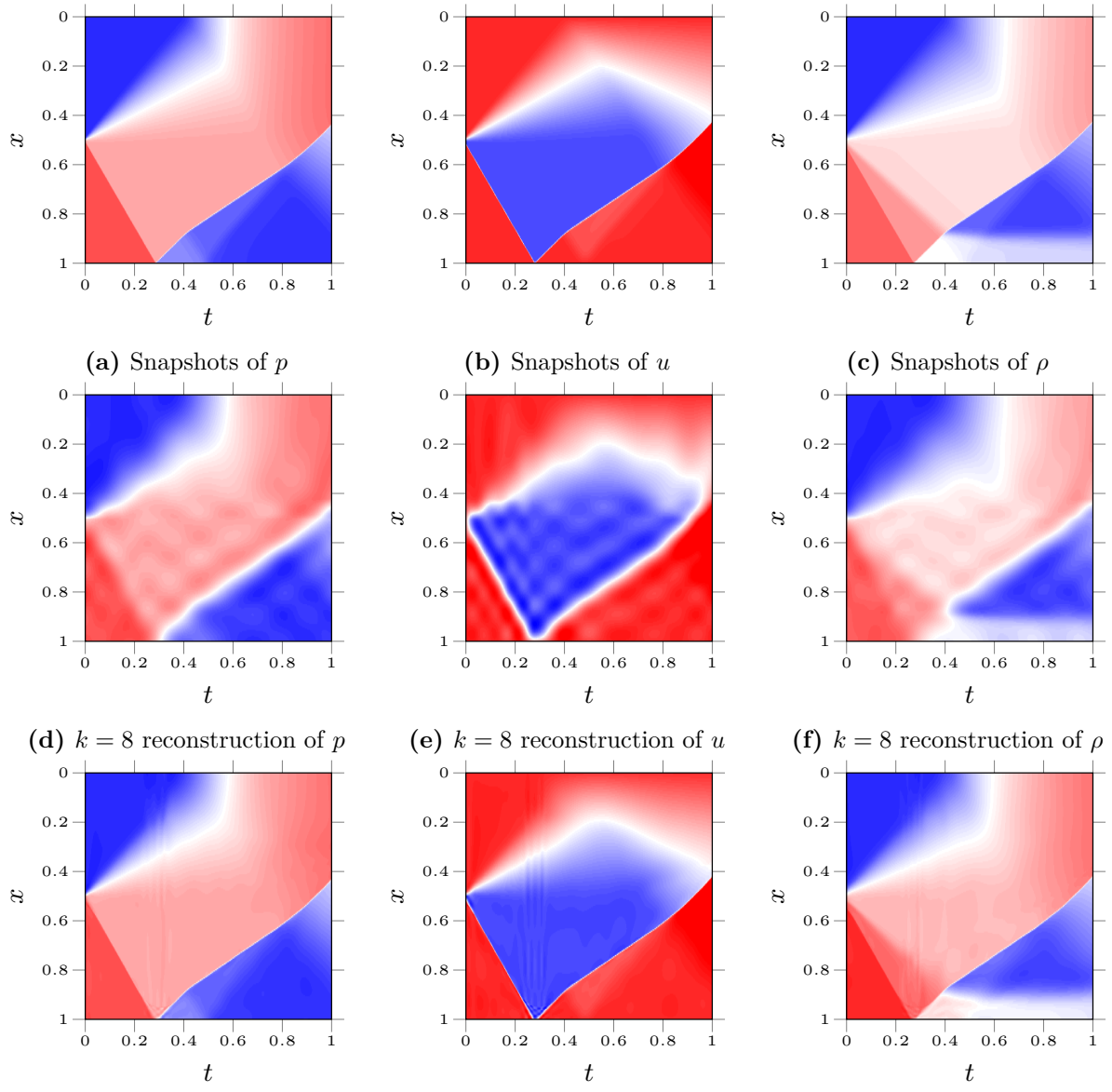
Consider the inviscid Euler equations of fluid flows for one-dimensional Riemann problem,

$$\frac{\partial}{\partial t}\boldsymbol{q} + \frac{\partial}{\partial x}\boldsymbol{f}_x = 0, \tag{5.7}$$

where $\boldsymbol{q} = [\rho, \rho u, \rho e]^T$, $\boldsymbol{f}_x = [\rho u, \rho u^2 + p, \rho u H]^T$, $H = e + p/\rho$, and $p = \rho(\gamma - 1)(e - 0.5u^2)$ for air as a perfect gas with specific heat ratio of $\gamma = 1.4$.

We consider the initial conditions corresponding to the Sod's shock tube problem [174]: $[\rho, p, u]_L = [1, 1, 0]$ for $x = [0, 0.5]$ and $[\rho, p, u]_R = [0.125, 0.1, 0]$ for $x = [0.5, 1]$. The inviscid Euler equation of (5.7) is solved using a first-order Roe scheme for $t \in [0, 1.0]$. The optimization problem of (2.15) are solved setting $r = 2$, $k_r = 4$, $\mathfrak{v}_{\min} = \Delta x_{\min} = 10^{-4}$, $\boldsymbol{\Gamma}_1 = 0.015\boldsymbol{D}_{xx}$ and $\boldsymbol{\Gamma}_2 = 0.015\boldsymbol{D}_{tt}$, where $\boldsymbol{D}_{xx}$ and $\boldsymbol{D}_{tt}$ are the second derivative matrices in space and time, respectively.

The solution of the HFM and rank-8 reconstructions of the primitive variables on the POD subspace and the identified manifold are compared in Fig. 5.6 and Fig. 5.7. The non-physical oscillations of the low-rank reconstructions are significantly diminished on the identified manifold while the shocks fronts are preserved. The error decreases in both cases by increasing the number of bases (Fig. 5.8). Both the error and the rate of decay of the reconstruction error on the learned manifold of time-varying grids are improved compared to the POD subspace on the Eulerian grid; Therefore, for a large range of k and k_r, the proposed method outperforms the traditional method.

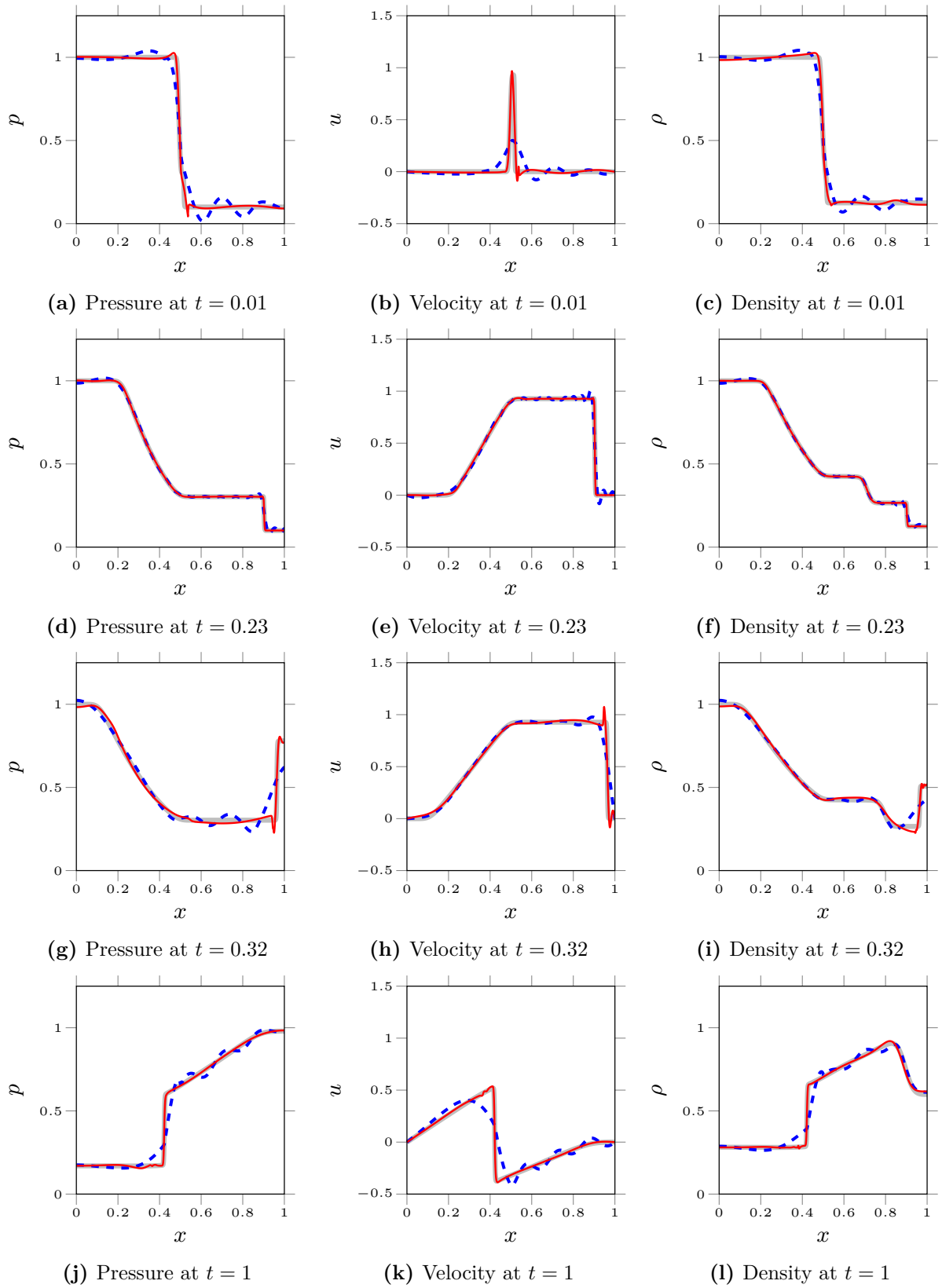


(a) Snapshots of p **(b)** Snapshots of u **(c)** Snapshots of ρ

(d) $k = 8$ reconstruction of p **(e)** $k = 8$ reconstruction of u **(f)** $k = 8$ reconstruction of ρ

(g) $k_r = 8$ reconstruction of p **(h)** $k_r = 8$ reconstruction of u **(i)** $k_r = 8$ reconstruction of ρ

Figure 5.6: One-dimensional Riemann problem (Sod's shock tube). (5.6a) to (5.6c): Snapshots of the primitive variables, (5.6d) to (5.6f): rank-8 POD reconstruction of primitive variables, (5.6g) to (5.6i): rank-8 reconstruction of primitive variables on the learned manifold, with rank-2 grid ($r = 2$).



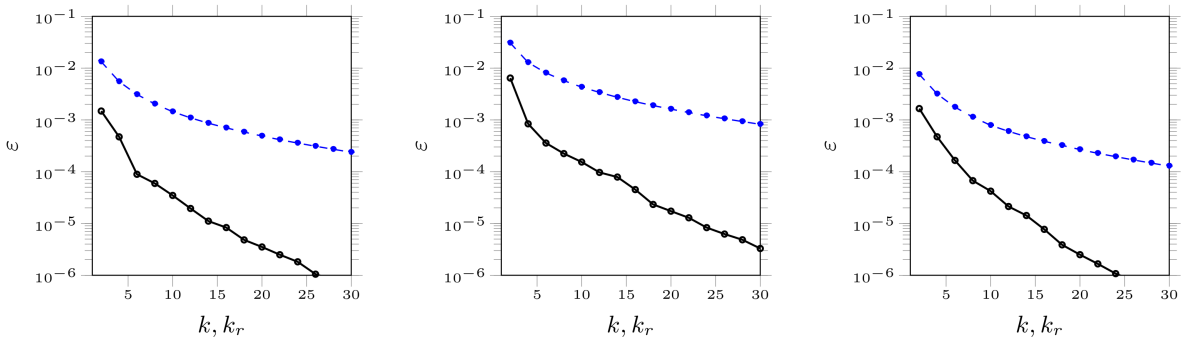
(a) Pressure at $t = 0.01$ **(b)** Velocity at $t = 0.01$ **(c)** Density at $t = 0.01$

(d) Pressure at $t = 0.23$ **(e)** Velocity at $t = 0.23$ **(f)** Density at $t = 0.23$

(g) Pressure at $t = 0.32$ **(h)** Velocity at $t = 0.32$ **(i)** Density at $t = 0.32$

(j) Pressure at $t = 1$ **(k)** Velocity at $t = 1$ **(l)** Density at $t = 1$

Figure 5.7: The primitive variables of the one-dimensional Riemann problem (Sod's shock tube) at $t \in \{0.01, 0.23, 0.32, 1\}$. Snapshots (thick gray line), the rank-8 POD reconstruction (dashed blue line), and the rank-8 reconstruction on low-rank registration-based manifold (solid red line).



(a) Pressure **(b)** Velocity **(c)** Density

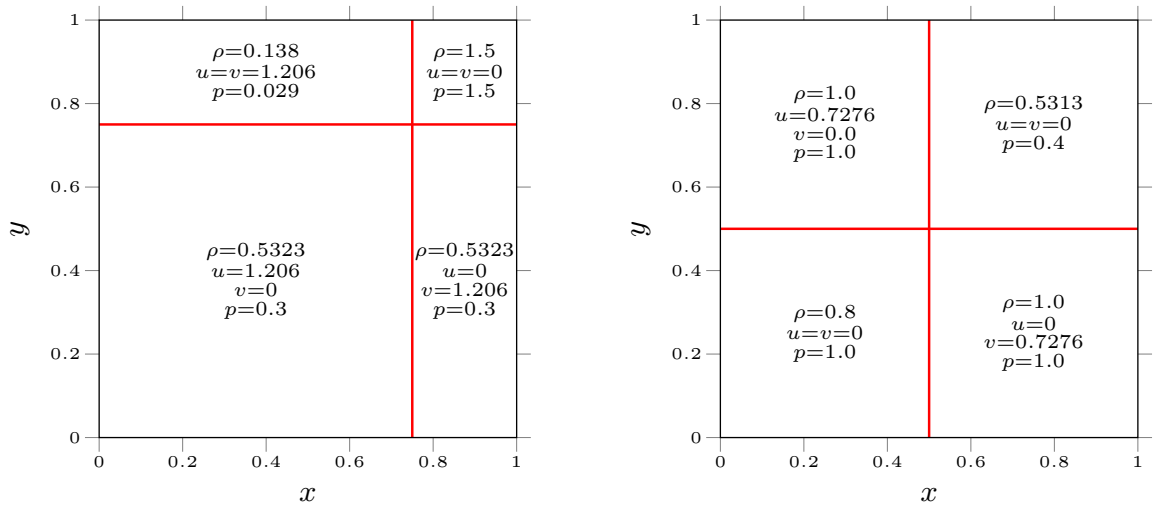
Figure 5.8: Reconstruction error of the one-dimensional Riemann problem (Sod's shock tube). The rank-k reconstruction on POD subspace (dashed blue line), and rank-k_r reconstruction on the identified manifold of $r = 2$ (solid black line).

5.1.5 Manifold learning in two-dimensional Riemann problem

Consider the inviscid Euler equations in two dimensions,

$$\frac{\partial}{\partial t}\boldsymbol{q} + \frac{\partial}{\partial x}\boldsymbol{f}_x + \frac{\partial}{\partial y}\boldsymbol{f}_y = 0, \tag{5.8}$$

where $\boldsymbol{q} = [\rho, \rho u, \rho v, \rho e]^T$, $\boldsymbol{f}_x = [\rho u, \rho u^2 + p, \rho uv, \rho uH]^T$, $\boldsymbol{f}_y = [\rho v, \rho uv + p, \rho v^2 + p, \rho vH]^T$, and $H = e + p/\rho$, $p = \rho(\gamma - 1)\left(e - 0.5\left(u^2 + v^2\right)\right)$ in the domain $(x, y, t) \in [0, 1] \times [0, 1] \times [0, t_{\max}]$, with initial conditions as illustrated in Fig. 5.9. Configurations are named corresponding to the paper in which they were originally proposed [2].



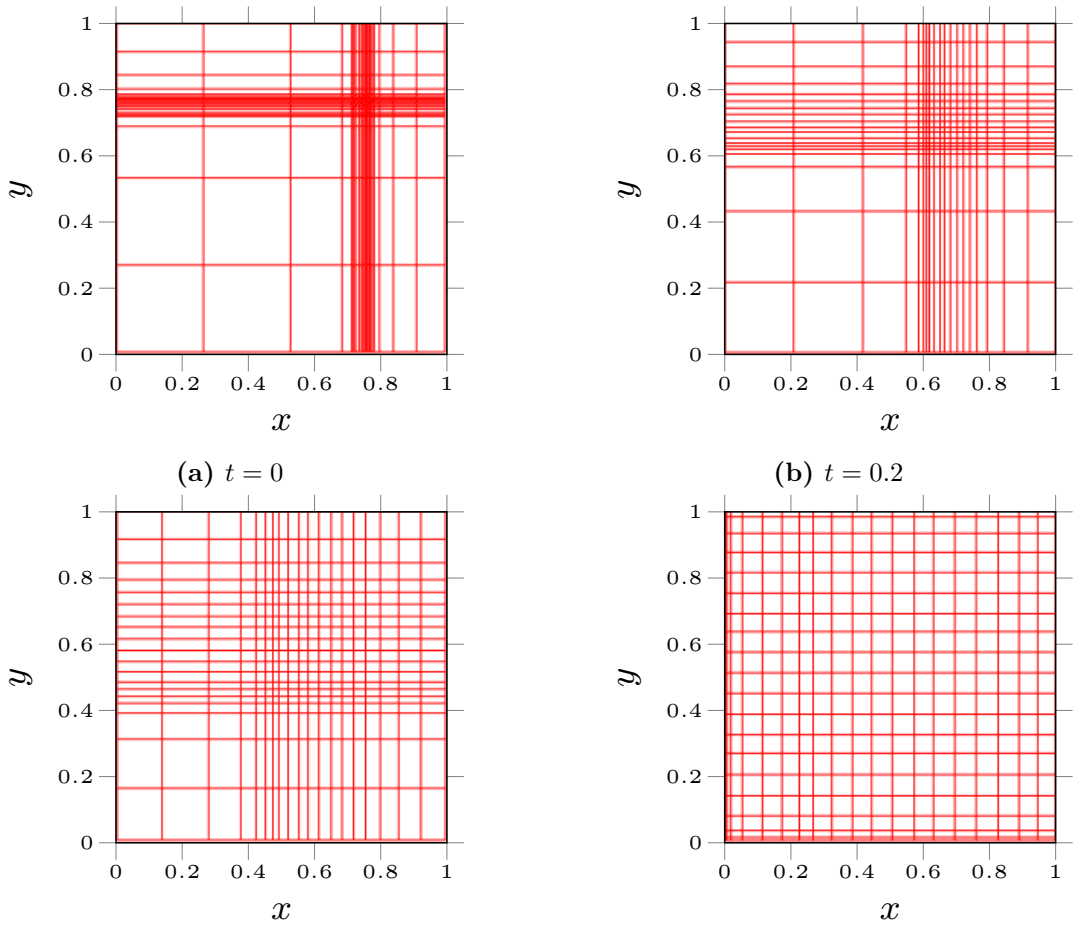
(a) Configuration 3 **(b)** Configuration 12

Figure 5.9: Initial conditions for two-dimensional Riemann problems, taken from Lax and Liu [2].

The snapshots of primitive variables are generated using a high-order artificial viscosity scheme coupled with a 4^{th}-order Runge-Kutta time discretization with $\Delta t = 5 \times 10^{-4}$ on a 150×150 grid. A rank-2 time-varying grid ($r = 2$) is learned via (2.15) setting $k_r = 4$, and $\boldsymbol{\Gamma}_1 = 0.05\boldsymbol{D}_{xx} = 0.05\boldsymbol{D}_{yy}$, where $\boldsymbol{D}_{xx}$ and $\boldsymbol{D}_{yy}$ are the second derivative matrices in x and y directions and $\boldsymbol{\Gamma}_2 = 0.05\boldsymbol{D}_{tt}$, where $\boldsymbol{D}_{tt}$ is the second derivative matrix in time. Also, $v_{\min} = \Delta x_{\min}\Delta y_{\min}$, where $\Delta x_{\min} = \Delta y_{\min} = 6.7 \times 10^{-4}$. The rank-2 time-varying grids ($r = 2$) representing the learned manifold for order reduction of configuration 3 and 12 are depicted in Fig. 5.10 and Fig. 5.14. In Fig. 5.11 and Fig. 5.15, the rank-8 low-dimensional representations of density contours are compared on the constant grid ($k = 8$) and the learned manifold ($k_r = 8$). On the learned manifold, traveling shocks

are conserved and free of non-physical oscillatory solutions, resulting in a significant error reduction (Fig. 5.13c and Fig. 5.17c). The density on the diagonal of the domain are plotted in Fig. 5.12 and Fig. 5.16.

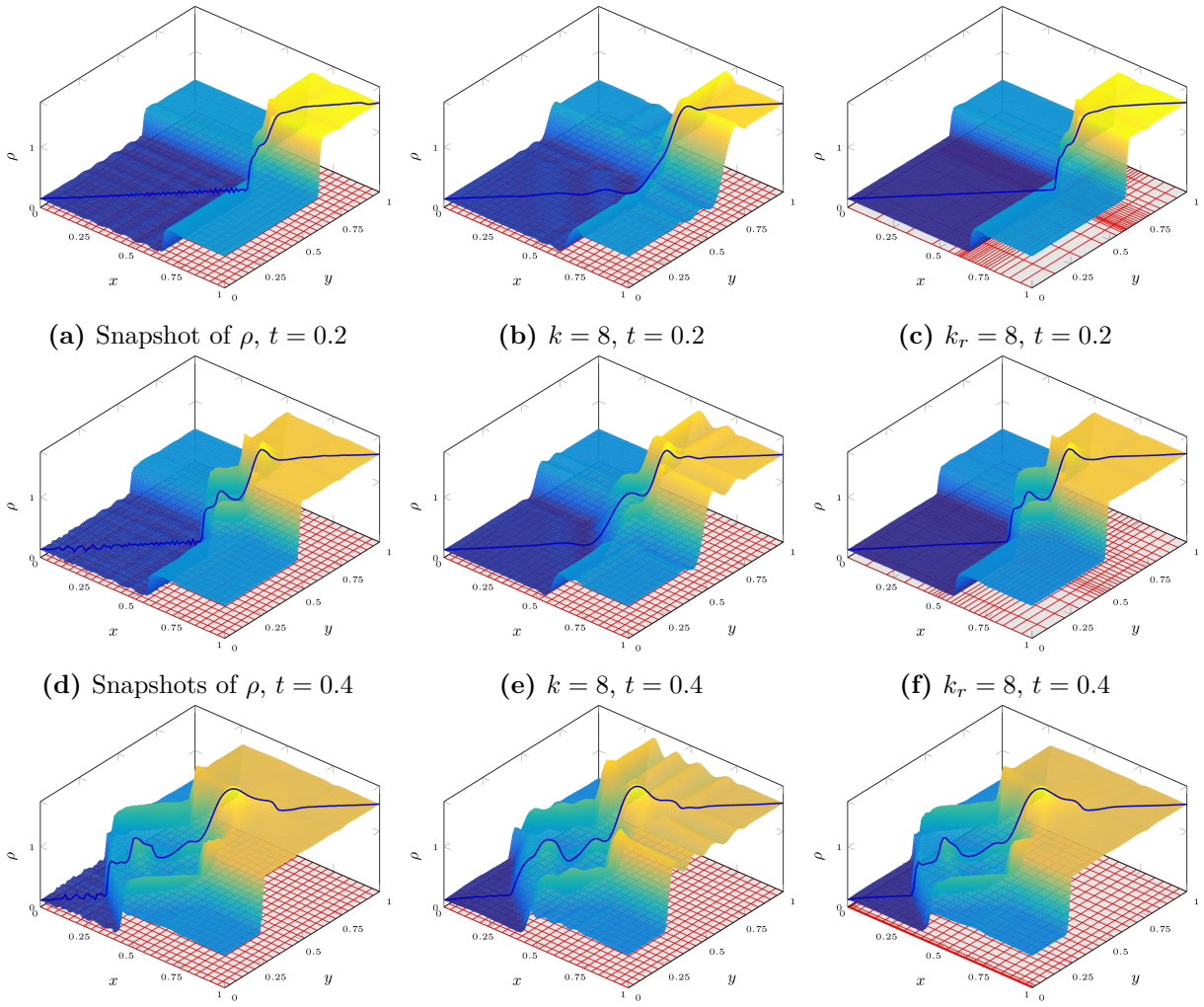
The numerical scheme used to solve the high fidelity models in the present section is not total variation diminishing (TVD), therefore it leads to spurious oscillations as the shock travels. These are different from the oscillations caused by low-rank approximation of a traveling shock discussed in §1.1. It is especially clear in configuration 3, as an example, in the region between the shock and $x = 0$ in Fig. 5.12. The oscillations are filtered in construction of low-rank representations, since these high-frequency bases are of low-energy content and are dismissed at the truncation step. This can also be seen in Fig. 5.12. Implementing flux-splitting weighted essentially non-oscillatory (WENO) schemes can resolve this issue at the high fidelity level, for a survey see [175]. However, the evaluation of the flux limiters on the reduced order level requires special considerations that is out of the scope of the present thesis.



(a) $t = 0$ **(b)** $t = 0.2$

(c) $t = 0.4$ **(d)** $t = 0.8$

Figure 5.10: The time-varying rank-2 grid corresponding to the identified manifold of the configuration 3 of two-dimensional Riemann problem.

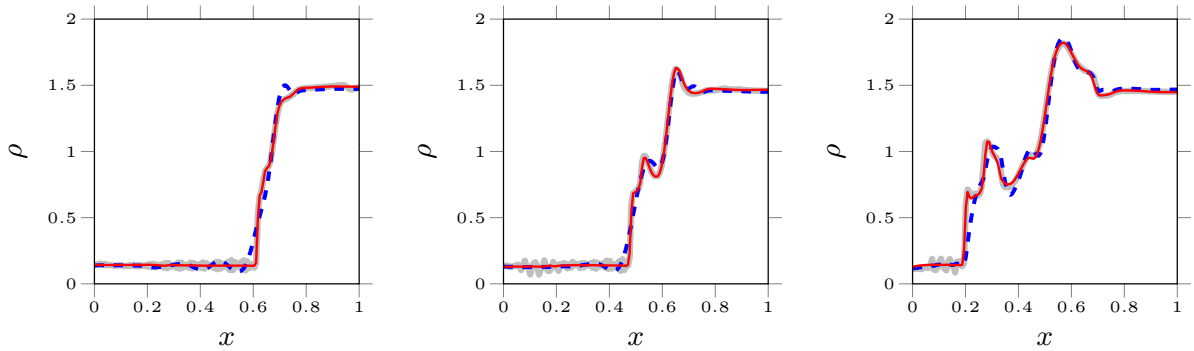


(a) Snapshot of ρ, $t = 0.2$ **(b)** $k = 8$, $t = 0.2$ **(c)** $k_r = 8$, $t = 0.2$

(d) Snapshots of ρ, $t = 0.4$ **(e)** $k = 8$, $t = 0.4$ **(f)** $k_r = 8$, $t = 0.4$

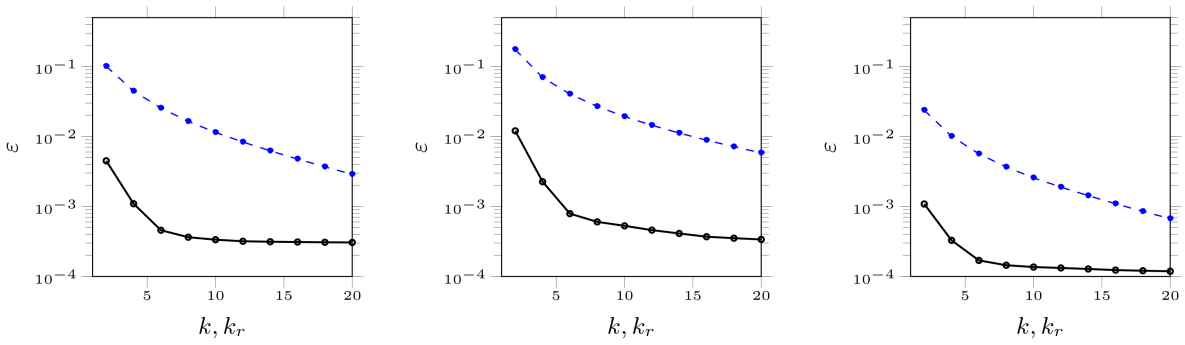
(g) Snapshots of ρ, $t = 0.8$ **(h)** $k = 8$, $t = 0.8$ **(i)** $k_r = 8$, $t = 0.8$

Figure 5.11: The density snapshots of the two-dimensional Riemann problem at $t \in \{0.2, 0.4, 0.8\}$ of the simulations with configuration 3 initial conditions and their rank-8 reconstruction, on top of the computational grid representative of the corresponding manifolds.



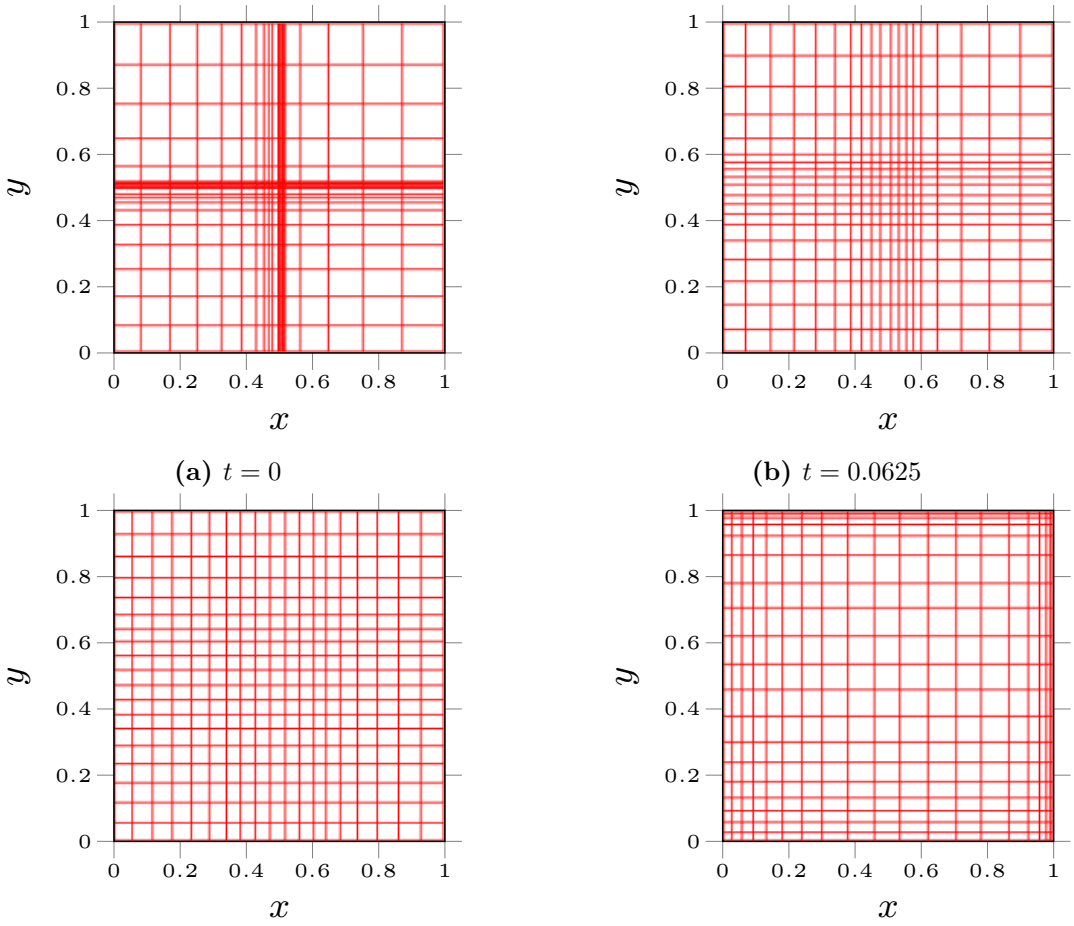
(a) Density at $t = 0.20$ **(b)** Density at $t = 0.40$ **(c)** Density at $t = 0.80$

Figure 5.12: Configuration 3 of two-dimensional Riemann problem, Snapshots (thick gray line), POD reconstruction with $k = 8$ (dashed blue line), Reconstruction on time-varying grid with $k_r = 8$ (solid red line).



(a) Pressure **(b)** Velocity **(c)** Density

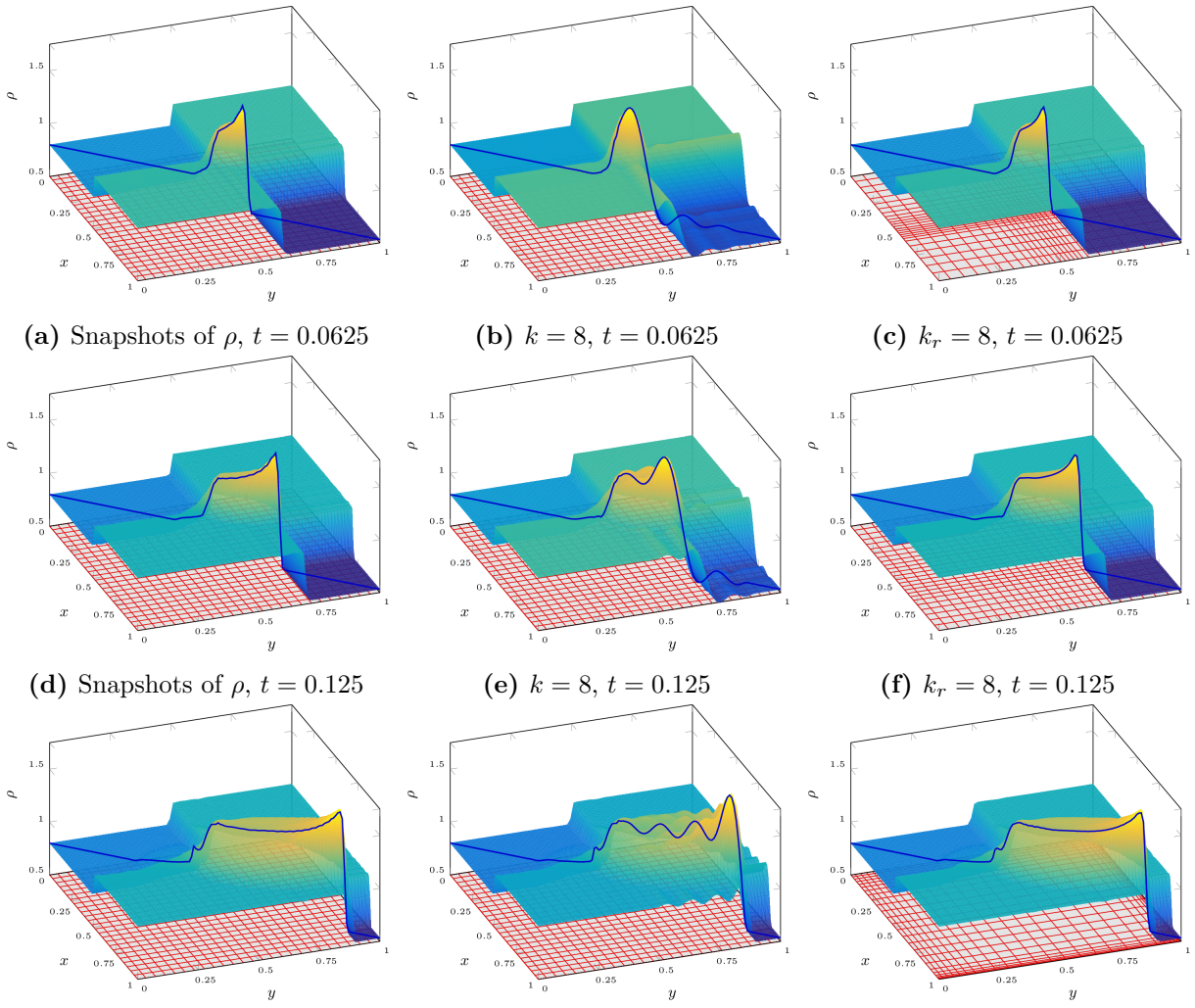
Figure 5.13: Reconstruction error in configuration 3 of the two-dimensional Riemann problem. The rank-k reconstruction on POD subspace (dashed blue line), and rank-k_r reconstruction on the identified manifold of $r = 2$ (solid black line).



(a) $t = 0$ **(b)** $t = 0.0625$

(c) $t = 0.125$ **(d)** $t = 0.25$

Figure 5.14: The time-varying rank-2 grid corresponding to the identified manifold of the configuration 12 of two-dimensional Riemann problem.

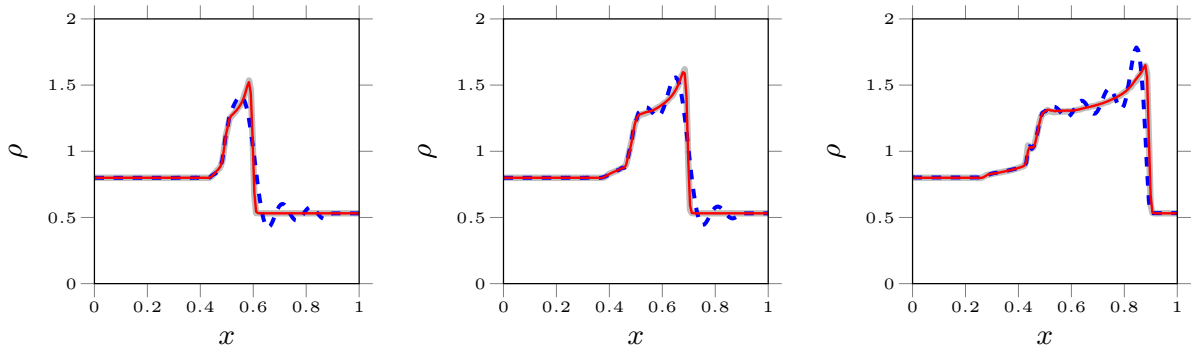


(a) Snapshots of ρ, $t = 0.0625$ **(b)** $k = 8$, $t = 0.0625$ **(c)** $k_r = 8$, $t = 0.0625$

(d) Snapshots of ρ, $t = 0.125$ **(e)** $k = 8$, $t = 0.125$ **(f)** $k_r = 8$, $t = 0.125$

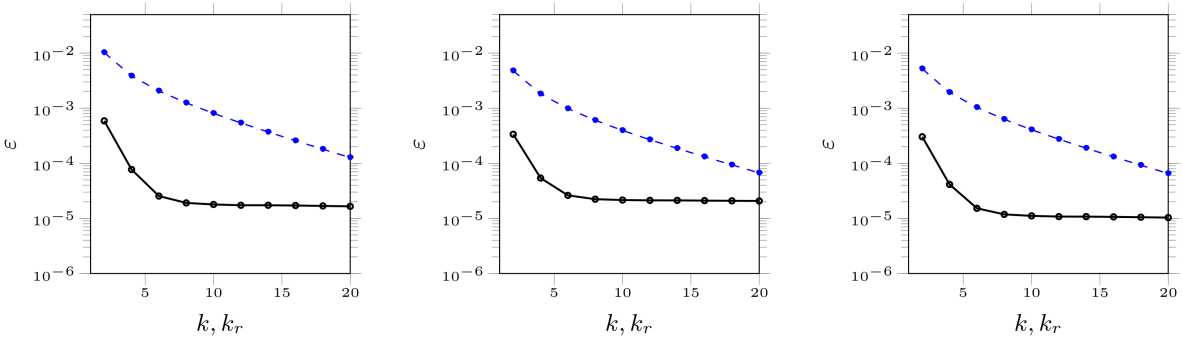
(g) Snapshots of ρ, $t = 0.25$ **(h)** $k = 8$, $t = 0.25$ **(i)** $k_r = 8$, $t = 0.25$

Figure 5.15: The snapshots of density of the two-dimensional Riemann problem at $t \in \{0.0625, 0.125, 0.25\}$ of the simulations with configuration 12 initial conditions and their rank-8 reconstruction, on top of the computational grid representative of the corresponding manifolds.



(a) Density at $t = 0.0625$ **(b)** Density at $t = 0.125$ **(c)** Density at $t = 0.25$

Figure 5.16: Configuration 12 of two-dimensional Riemann problem, Snapshots (thick gray line), POD reconstruction with $k = 8$ (dashed blue line), Reconstruction on time-varying grid with $k_r = 8$ (solid red line).



(a) Pressure **(b)** Velocity **(c)** Density

Figure 5.17: Reconstruction error in configuration 12 of the two-dimensional Riemann problem. The rank-k reconstruction on POD subspace (dashed blue line), and rank-k_r reconstruction on the identified manifold of $r = 2$ (solid black line).

5.2 ROMs on the low-rank registration-based manifold

In this section, we construct ROMs on the identified optimal manifolds. The performance of both traditional projection-based ROMs and neural network-based ROMs, discussed in (§3), are demonstrated for time dependent examples first introduced in §5.1.

5.2.1 Second-order wave equation

In this section, we construct the projection-based ROMs on the constant Eulerian grid (§3.1.1) and the time-varying grids (§3.2.1) of the wave equation problem discussed in §5.1.3. The time-varying grid represents the manifold on which the Kolmogorov n-width are optimally reduced as identified in §5.1.4. The ROMs' error are compared in Fig. 5.18. As expected, the increase in reducibility of the snapshots on the learned manifold directly translates to more efficient ROMs.

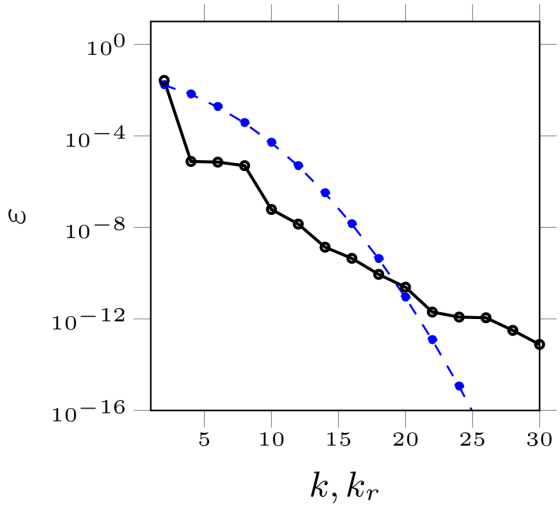
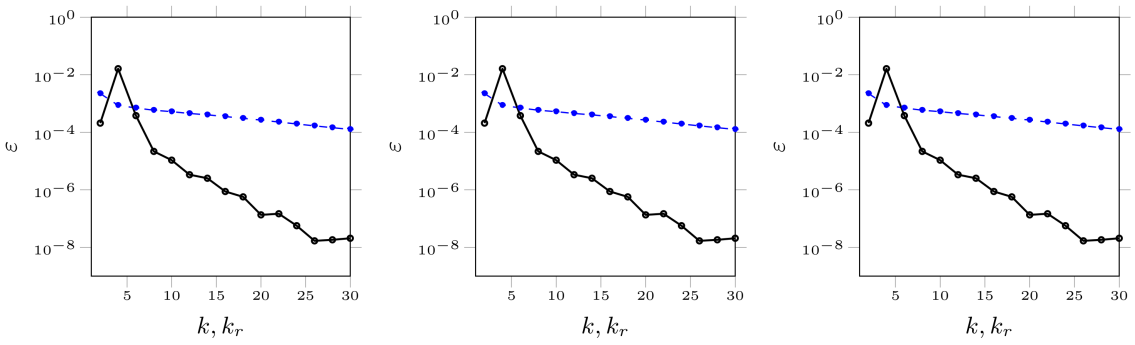


Figure 5.18: ROM error in second-order wave equation. The rank-k ROM on POD subspace (dashed blue line), and rank-k_r ROM on the identified manifold of $r = 2$ (solid black line).

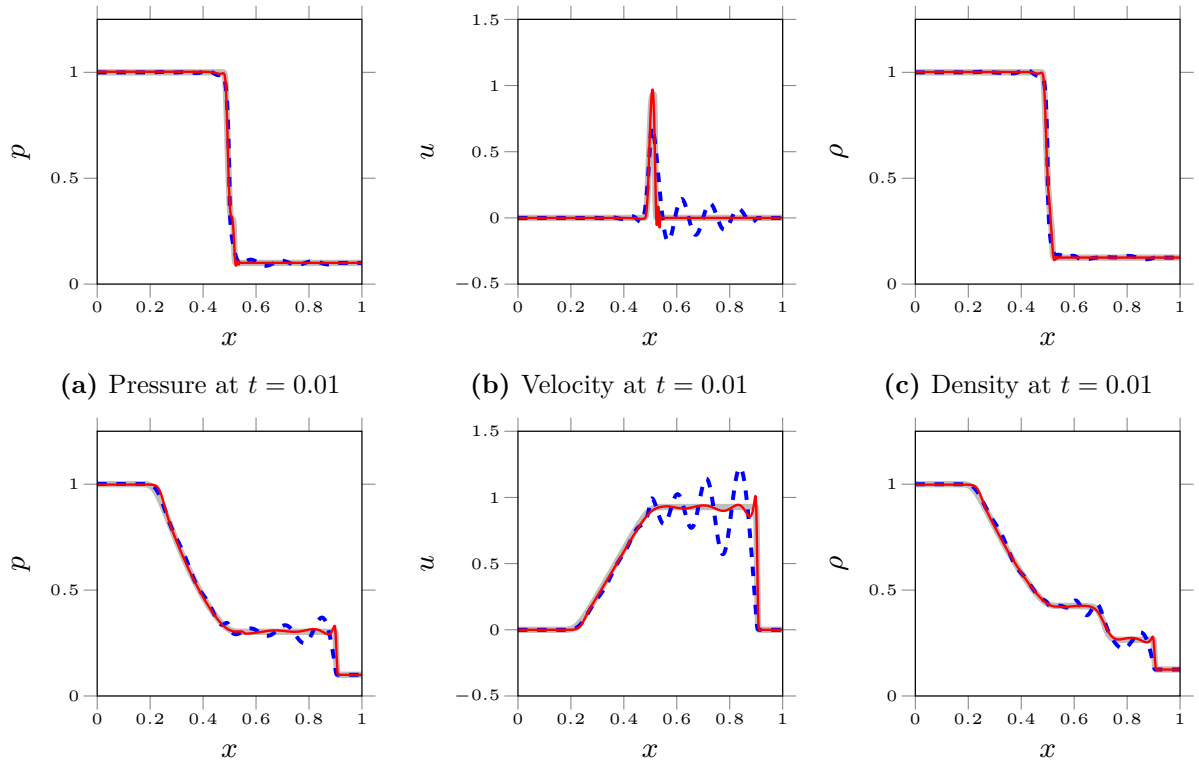
5.2.2 One-dimensional Riemann problem

In this section, we construct projection-based ROMs on the constant Eulerian grid (§3.1.1) and the time-varying grids (§3.2.1) of the one-dimensional Riemann problem discussed in §5.1.4. The time-varying grids represent the manifold on which the Kolmogorov n-width are optimally reduced as realized in §5.1.4 for $t \in [0, 0.23]$. The error of these ROMs are compared in Fig. 5.19. Overall, the ROMs on the optimal manifold capture the evolution of the moving discontinuity with substantially fewer oscillations that results in a lower ROM error (Fig. 5.20 and Fig. 5.21). The non-monotonic decrease of the ROM error is due to the absence of the stability guarantee of the ROMs (see §4).



(a) Pressure **(b)** Velocity **(c)** Density

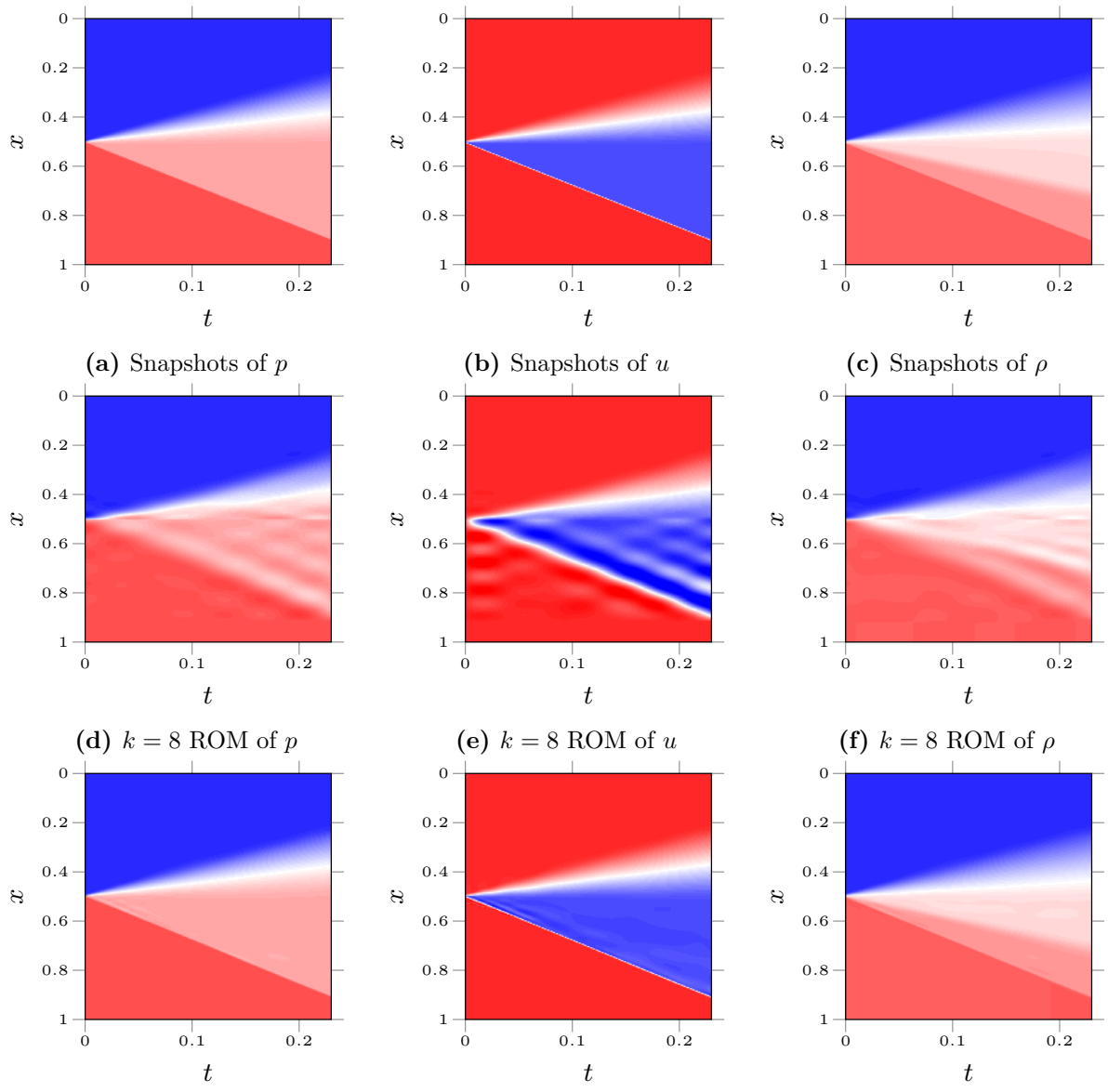
Figure 5.19: ROM error of the one-dimensional Riemann problem. The rank-k ROM on POD subspace (dashed blue line), and rank-k_r ROM on the identified manifold of $r = 2$ (solid black line).



(a) Pressure at $t = 0.01$ **(b)** Velocity at $t = 0.01$ **(c)** Density at $t = 0.01$

(d) Pressure at $t = 0.23$ **(e)** Velocity at $t = 0.23$ **(f)** Density at $t = 0.23$

Figure 5.20: The primitive variables of the one-dimensional Riemann problem (Sod's shock tube) at $t \in \{0.01, 0.23\}$. Snapshots (thick gray line), POD ROM with $k = 8$ (solid red line). ROM with $k_r = 8$ (solid red line).



(a) Snapshots of p **(b)** Snapshots of u **(c)** Snapshots of ρ

(d) $k = 8$ ROM of p **(e)** $k = 8$ ROM of u **(f)** $k = 8$ ROM of ρ

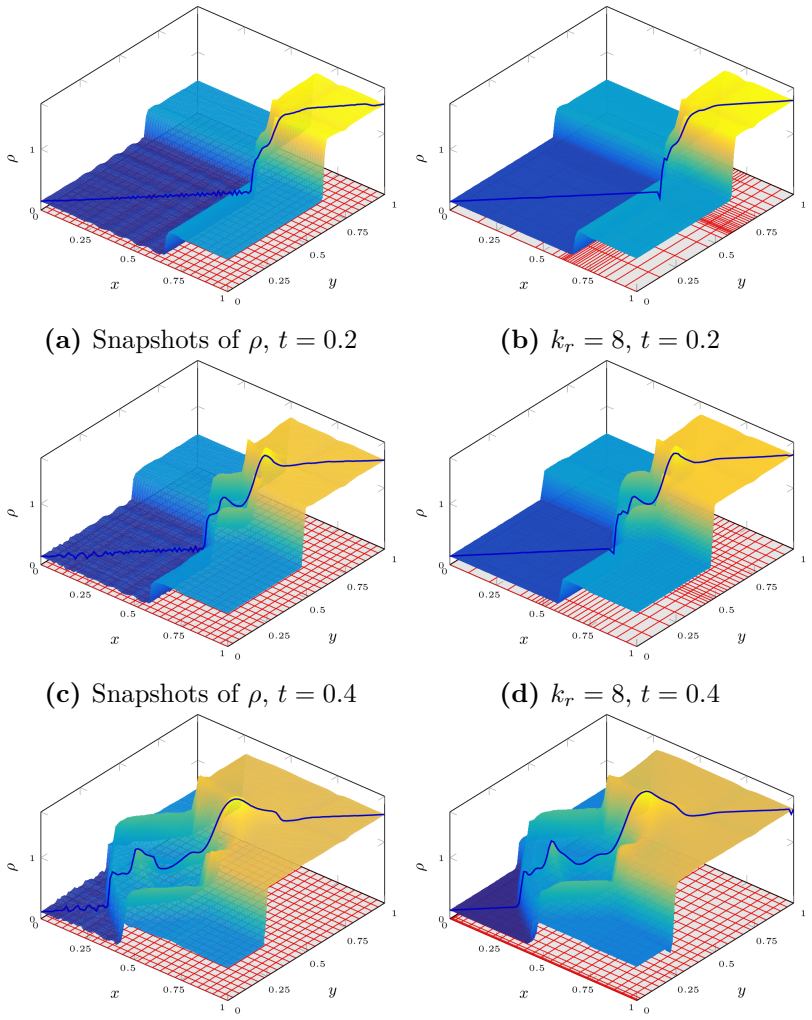
(g) $k_r = 8$ ROM of p **(h)** $k_r = 8$ ROM of u **(i)** $k_r = 8$ ROM of ρ

Figure 5.21: The primitive variables snapshots on the domain of the one-dimensional Riemann problem (Sod's shock tube). (5.21a) to (5.21c): Snapshots of the primitive variables, (5.21d) to (5.21f): POD ROM with $k = 8$, (5.21g) to (5.21i): ROM with $k_r = 8$.

5.2.3 Two-dimensional Riemann problem

In this section, we construct projection-based ROMs on the constant Eulerian grid (§3.1.1) and the time-varying grids (§3.2.1) of the two-dimensional Riemann problem discussed in §5.1.5. The optimal manifolds are realized in §5.1.5.

The solution of the ROMs are compared to the high fidelity models for configuration 3 and 12 in Fig. 5.22 to Fig. 5.25. The Eulerian ROMs for configuration 3 are unstable and become unbounded within the range of the simulation, while some of the ROMs on the time-varying grid remain a close approximation of the snapshots. Overall, the ROMs on the optimal manifold capture the evolution of the moving discontinuity with substantially fewer oscillations that results in a lower ROM error (Fig. 5.26 and Fig. 5.27). The missing points in the figures are due to the unstable and unbounded ROMs. Attributing the instability of Galerkin ROMs in convection-dominated flows with large Kolmogorov n-width has recently gained more attentions [176].

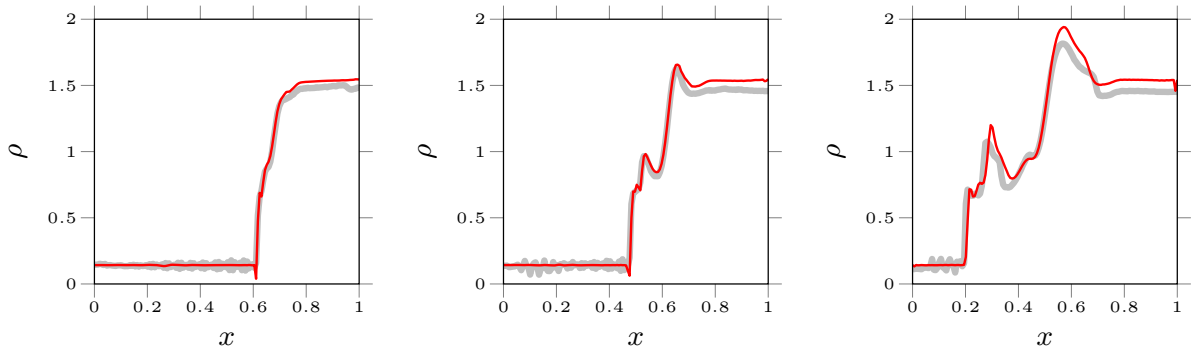


(a) Snapshots of ρ, $t = 0.2$ **(b)** $k_r = 8$, $t = 0.2$

(c) Snapshots of ρ, $t = 0.4$ **(d)** $k_r = 8$, $t = 0.4$

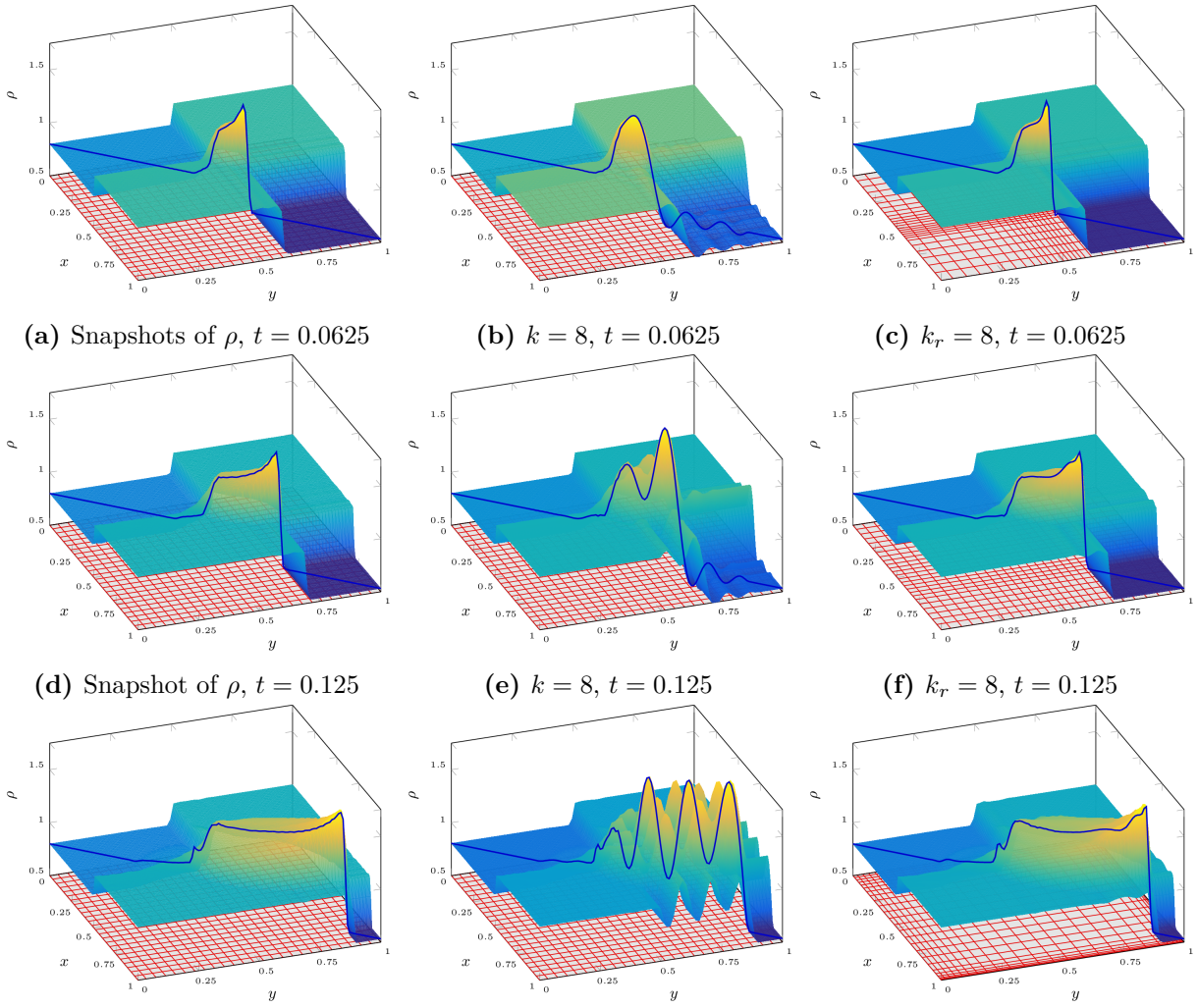
(e) Snapshots of ρ, $t = 0.8$ **(f)** $k_r = 8$, $t = 0.8$

Figure 5.22: The density snapshots of the two-dimensional Riemann problem at $t \in \{0.2, 0.4, 0.8\}$ of configuration 3 and its rank-8 ROMs on the time-varying grid. POD ROMs are unstable and unbounded.



(a) Density at $t = 0.20$ **(b)** Density at $t = 0.40$ **(c)** Density at $t = 0.80$

Figure 5.23: Density on diagonal of the domain in configuration 3 of the two-dimensional Riemann problem at $t \in \{0.2, 0.4, 0.8\}$. HFM (thick gray line), POD ROMs are unstable and unbounded. ROM with $k_r = 8$ (solid red line).

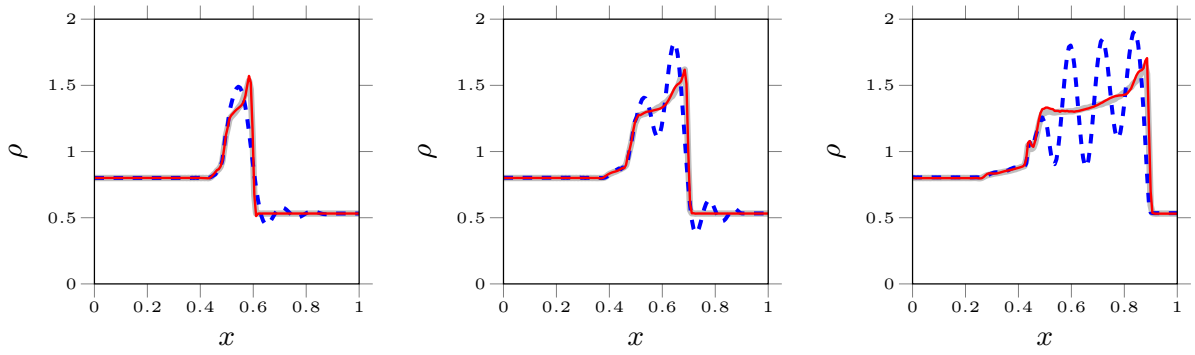


(a) Snapshots of ρ, $t = 0.0625$ **(b)** $k = 8$, $t = 0.0625$ **(c)** $k_r = 8$, $t = 0.0625$

(d) Snapshot of ρ, $t = 0.125$ **(e)** $k = 8$, $t = 0.125$ **(f)** $k_r = 8$, $t = 0.125$

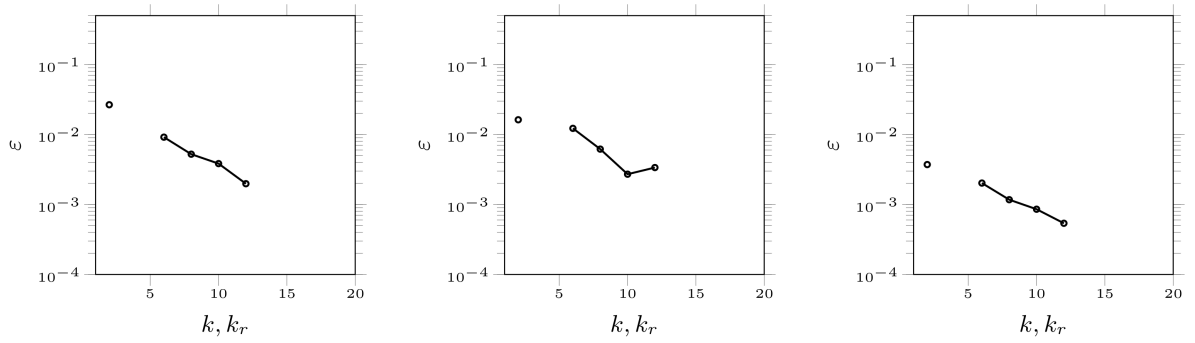
(g) Snapshot of ρ, $t = 0.25$ **(h)** $k = 8$, $t = 0.25$ **(i)** $k_r = 8$, $t = 0.25$

Figure 5.24: Density snapshots on the domain in configuration 12 of the two-dimensional Riemann problem at $t \in \{0.0625, 0.125, 0.25\}$. Snapshots and rank-8 ROMs, on top of the computational grid representative of the corresponding manifolds.



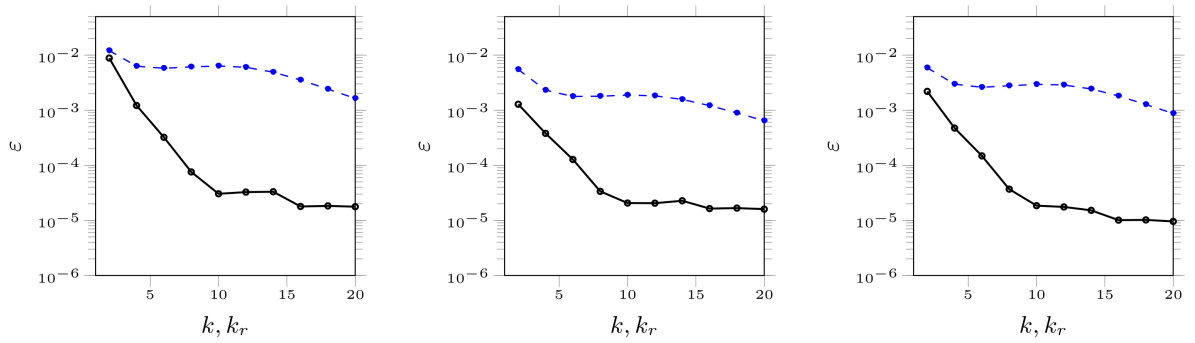
(a) Density at $t = 0.0625$ **(b)** Density at $t = 0.125$ **(c)** Density at $t = 0.25$

Figure 5.25: Density on diagonal of the domain in configuration 12 of the two-dimensional Riemann problem at $t \in \{0.0625, 0.125, 0.25\}$. Snapshots (thick gray line), POD ROM with $k = 8$ (dashed blue line), ROM with $k_r = 8$ (solid red line).



(a) Pressure **(b)** Velocity **(c)** Density

Figure 5.26: ROM error in configuration 3 of the two-dimensional Riemann problem. The rank-k ROM on POD subspace (dashed blue line), and rank-k_r ROM on the identified manifold of $r = 2$ (solid black line).

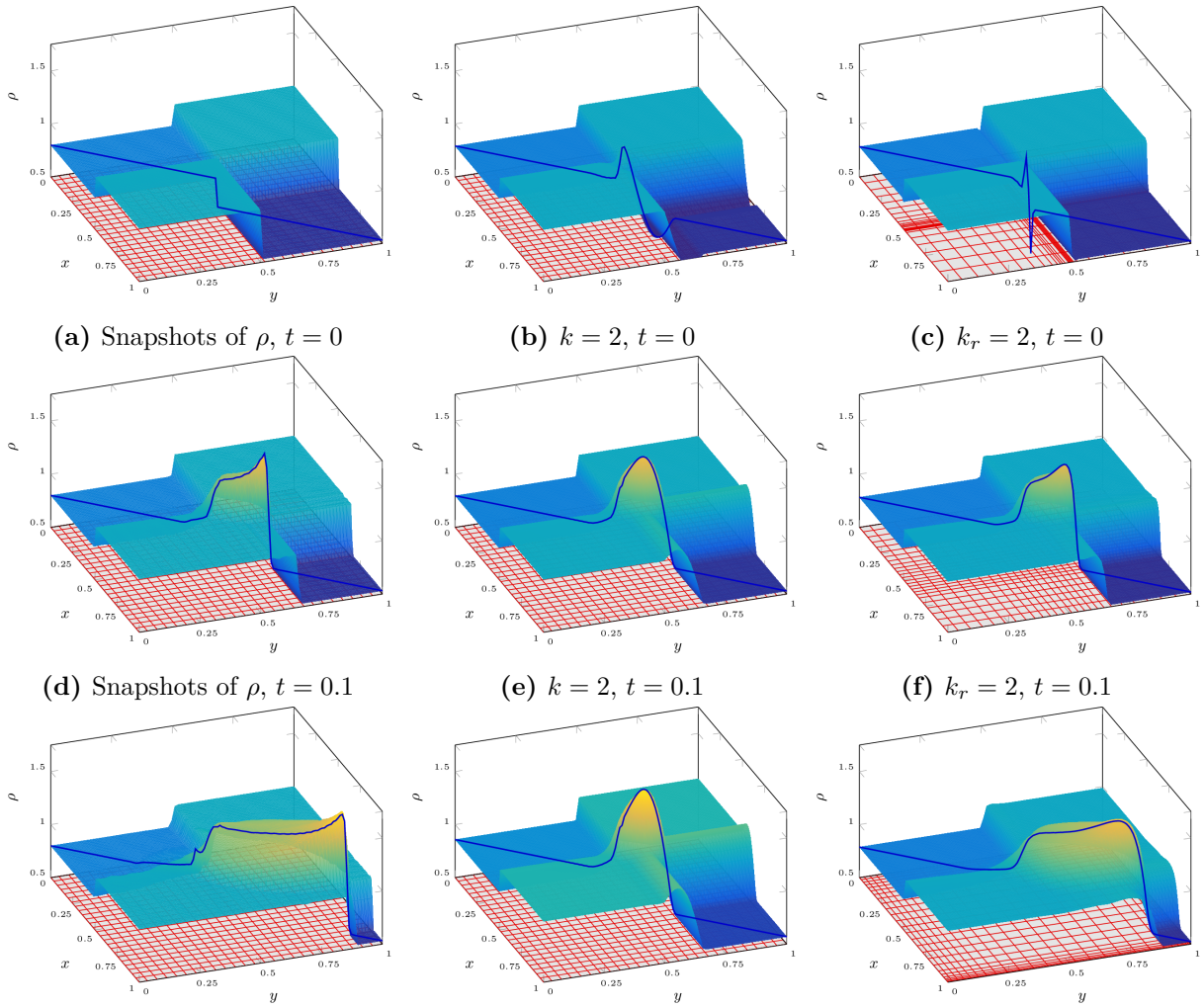


(a) Pressure **(b)** Velocity **(c)** Density

Figure 5.27: ROM error in configuration 12 of the two-dimensional Riemann problem. The rank-k ROM on POD subspace (dashed blue line), and rank-k_r ROM on the identified manifold of $r = 2$ (solid black line).

5.2.4 Prediction of two-dimensional Riemann using projection-based ROMs

In this section, the predictive capabilities of the projection-based ROMs on the learned manifold are compared to traditional ROMs on the POD subspace. Consider configuration 12 of the two-dimensional Riemann problem as described in §5.1.5. The high fidelity model is solved for $t = [0, 0.1]$ and the snapshot matrix is constructed. In the training stage, the time-varying grid is identified to minimize the error of the low-rank reconstruction given $k_r = 4$, and $r = 2$. The POD bases of the primitive variables are then constructed on the Eulerian grid and on the learned manifold. To reconstruct/predict beyond the training stage, the bases are extrapolated linearly in time beyond the training stage. Specifically, in the case of POD reconstruction where $\widetilde{\boldsymbol{M}} = \boldsymbol{U}\boldsymbol{V}$, the temporal bases ($\boldsymbol{V}$) are extrapolated linearly beyond the training range, and in the case of reconstruction on the learned manifold both of the snapshots and grid temporal bases ($\boldsymbol{V}$ and $\boldsymbol{V}_x$) are extrapolated beyond the training range, where $\widetilde{\boldsymbol{M}} = \mathcal{G}^{-1}(\boldsymbol{U}\boldsymbol{V})$ and the low-rank time varying grid is $\boldsymbol{U}_x\boldsymbol{V}_x$. In both cases the ROMs are evolved in time. In the case of the ROM on the identified manifold, the grid beyond the training range is similarly extrapolated in time, by extending $\boldsymbol{V}_x$. The rank-2 solution of ROMs on the domain and ROM and low-rank reconstruction on the diagonal of the domain are compared in Fig. 5.28 to Fig. 5.30. Both the POD ROM on the Eulerian grid and the ROM on the time-varying grid capture the formation of the shock and are close to the high fidelity model within the training range. However, beyond the training range, the shock is predicted to remain stationary on the Eulerian grid, while the shock captured by the ROM on the time-varying grid closely follows the solution of the high fidelity model. The ROM error the primitive variables on the diagonal of the domain are compared in Fig. 5.31, showing an order of magnitude decrease in the error on the identified manifold.

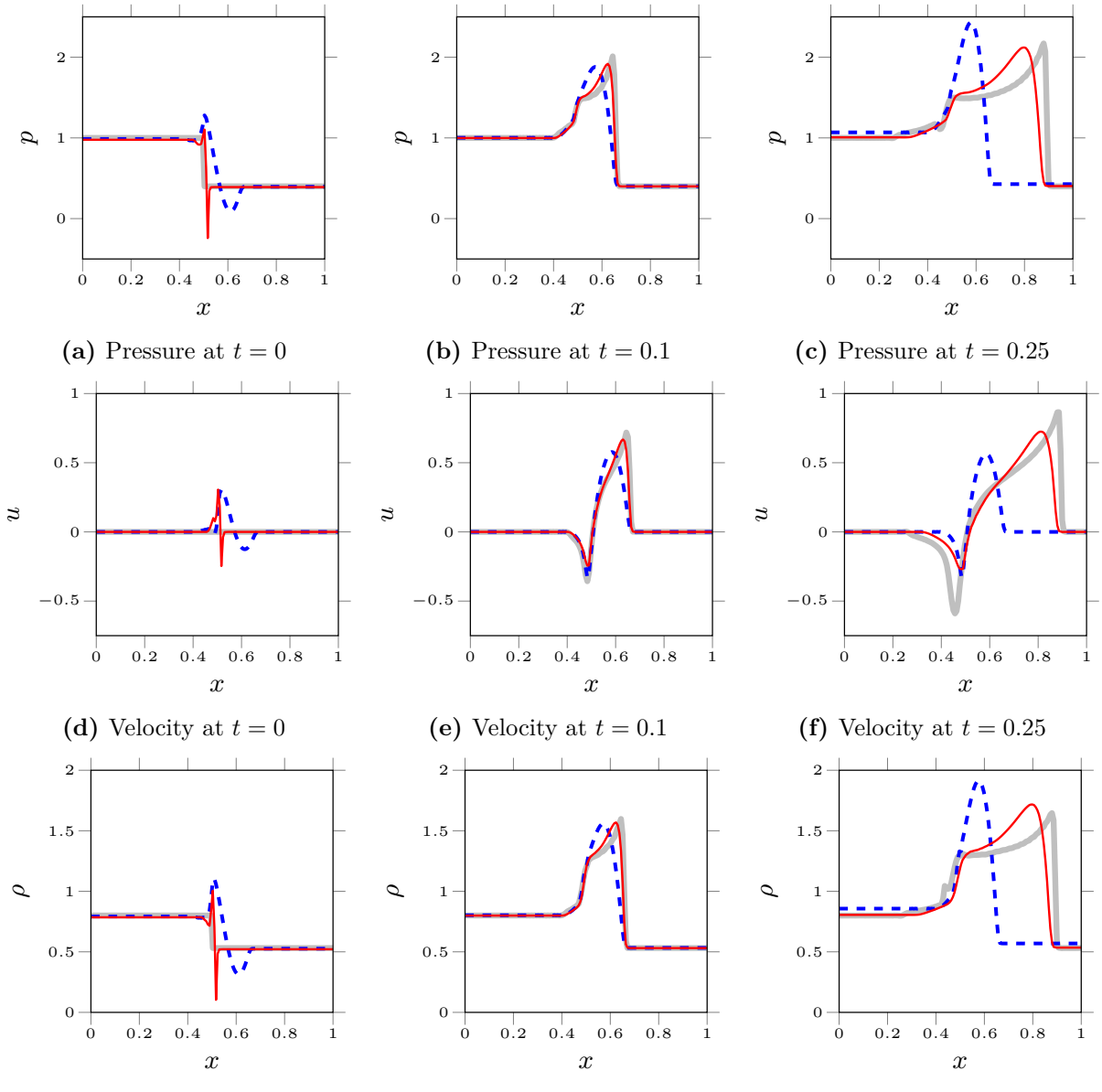


(a) Snapshots of ρ, $t = 0$ **(b)** $k = 2$, $t = 0$ **(c)** $k_r = 2$, $t = 0$

(d) Snapshots of ρ, $t = 0.1$ **(e)** $k = 2$, $t = 0.1$ **(f)** $k_r = 2$, $t = 0.1$

(g) Snapshots of ρ, $t = 0.25$ **(h)** $k = 2$, $t = 0.25$ **(i)** $k_r = 2$, $t = 0.25$

Figure 5.28: The density snapshots of the two-dimensional Riemann problem at $t \in \{0, 0.1, 0.25\}$ of configuration 12 and their rank-2 ROMs, on top of the computational grid representative of the corresponding manifolds. The bases are trained for $t = [0, 0.1]$.

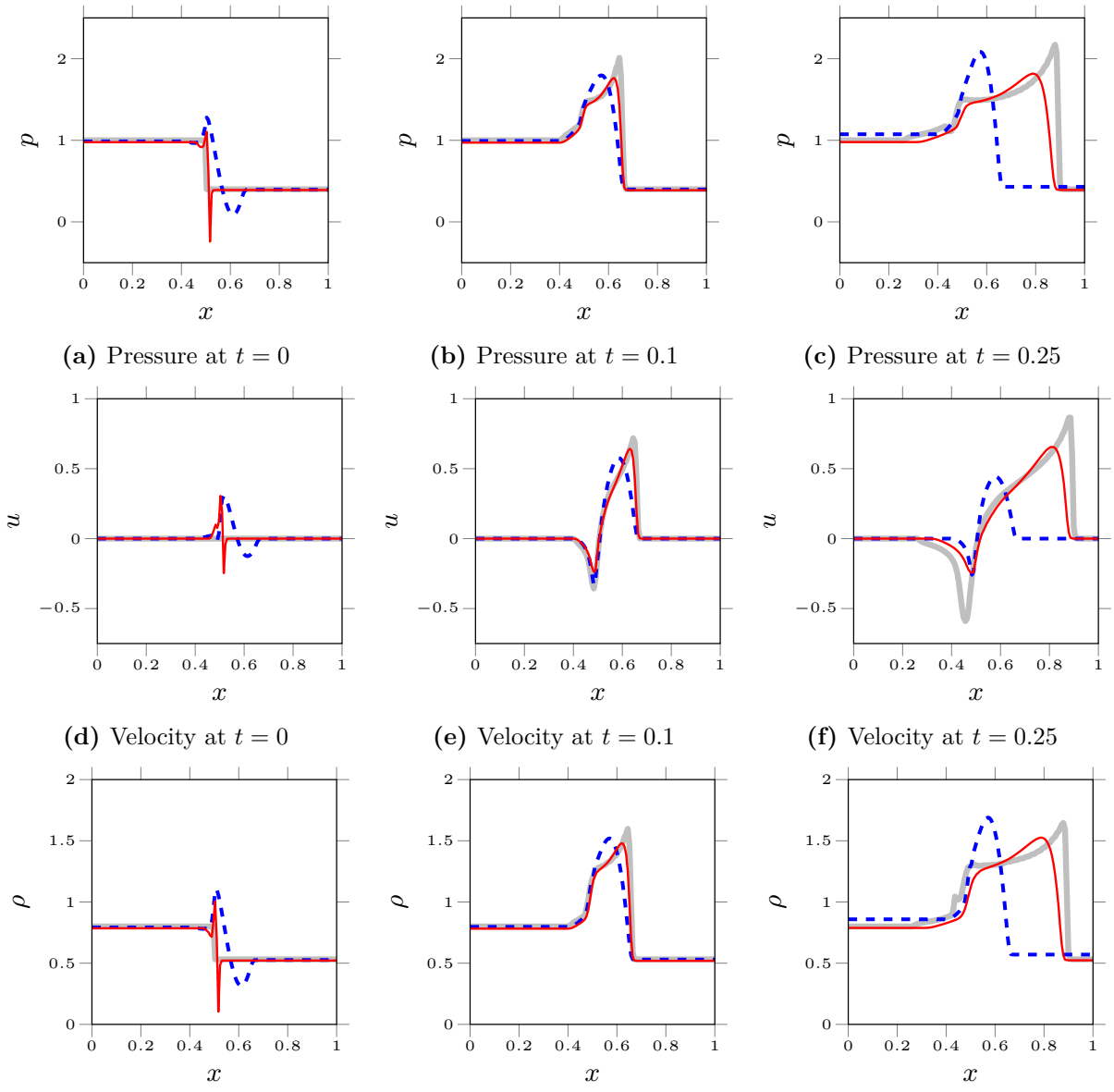


(a) Pressure at $t = 0$ **(b)** Pressure at $t = 0.1$ **(c)** Pressure at $t = 0.25$

(d) Velocity at $t = 0$ **(e)** Velocity at $t = 0.1$ **(f)** Velocity at $t = 0.25$

(g) Density at $t = 0$ **(h)** Density at $t = 0.1$ **(i)** Density at $t = 0.25$

Figure 5.29: Configuration 12 of two-dimensional Riemann problem, HFM (thick gray line), POD reconstruction with $k = 2$ (dashed blue line), reconstruction on time-varying grid with $k_r = 2$ (solid red line).

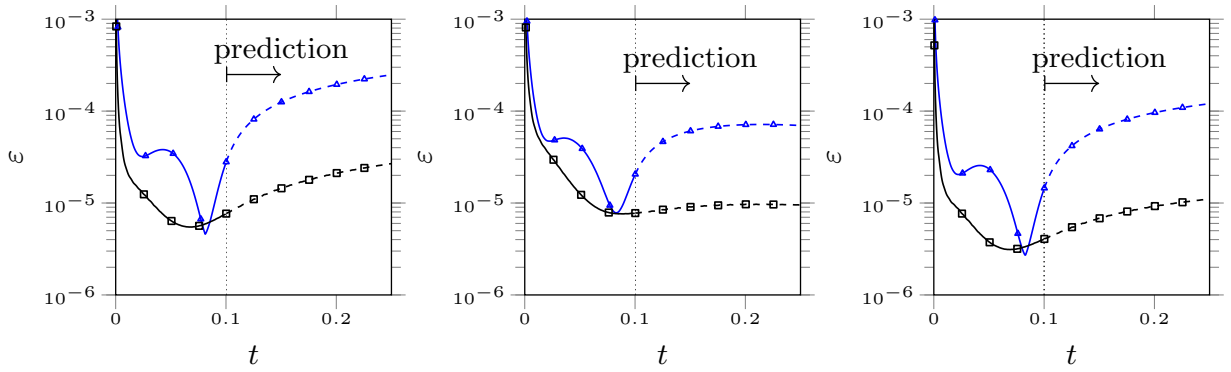


(a) Pressure at $t = 0$ **(b)** Pressure at $t = 0.1$ **(c)** Pressure at $t = 0.25$

(d) Velocity at $t = 0$ **(e)** Velocity at $t = 0.1$ **(f)** Velocity at $t = 0.25$

(g) Density at $t = 0$ **(h)** Density at $t = 0.1$ **(i)** Density at $t = 0.25$

Figure 5.30: Configuration 12 of two-dimensional Riemann problem, HFM (thick gray line), POD ROM with $k = 2$ (dashed blue line), ROM with $k_r = 2$ (solid red line).



(a) Pressure **(b)** Velocity **(c)** Density

Figure 5.31: Transient ROM error in configuration 12 of the two-dimensional Riemann problem. The error is measured on the diagonal of the domain (see Fig. 5.28). POD ROM (black lines with square markers), rank-2 ROMs on the identified manifold (blue lines triangle markers). The bases are trained on $t = [0, 0.1]$ (solid lines) and the prediction is extended to $t = 0.25$ (dashed lines).

5.2.5 Low-rank auto-encoder in an LSTM architecture

In this section, the proposed method is implemented as an auto-encoder layer wrapped around a traditional machine learning architecture to increase reducibility and subsequently to improve the predictive capabilities of an RNN approximating the governing PDEs.

In the traditional architecture, the densely connected auto-encoder and the LSTM are trained simultaneously (Fig. 3.1). In the proposed architecture (Fig. 3.2), the low-rank registration-based manifold identified by $\mathcal{G}(.)$, is trained and in the next step the densely connected auto-encoder and the LSTM cells are trained separately, i.e. the LSTM is trained to approximate $\mathcal{G}(\boldsymbol{M})$, where $\boldsymbol{M} \in \mathbb{R}^{N_x \times N_t}$ is the snapshots matrix on the constant Eulerian grid and $\mathcal{G}(.)$ interpolates the given snapshots to the time-varying grid. To identify the latent space, two consecutive neural network layers map the input of size $\mathbb{R}^{N_x}$ to $\mathbb{R}^{N_{\text{enc}}}$ and finally to $\mathbb{R}^{k}$ or $\mathbb{R}^{k_r}$, where the LSTM cells are trained to approximate the evolution of the latent variables in the traditional and the proposed architectures in $\mathbb{R}^{k}$ or $\mathbb{R}^{k_r}$, respectively. The neural network architectures are deployed in Keras [177]. The parameters and hyper-parameters are summarized in Table 5.1 and for any of the undeclared parameters, the default values as in [177] are used.[1] The predictive capabilities of the LSTM trained on the proposed low-rank registration-based auto-encoder is demonstrated in §5.2.6.

Table 5.1: Summary of the hyper-parameters and the neural network architectures.

	N_x	$N_{\mathbf{enc}}$	$k, k_r{}^*$	$k, k_r{}^+$	$\sigma_{\mathbf{enc}}$	$\sigma_{\mathbf{LSTM}}$	r	γ_x	γ_t
Viscous Burgers' equation	250	20	4	$[5, 20]$	tanh	tanh	1	1	1
Second-order wave equation	500	20	2	$[5, 20]$	tanh	tanh	2	10^{-3}	10^{-3}

* At the training stage of the low-rank registration-based auto-encoder,
$^+$ At the training and evolution of neural network-based LSTM

Viscous Burgers' equation

Consider the scalar, one-dimensional viscous Burgers' equation,

$$\frac{\partial w(x,t)}{\partial t} + w\frac{\partial w(x,t)}{\partial x} = \nu \frac{\partial^2 w(x,t)}{\partial x^2}, \tag{5.9}$$

[1] The codes are made available at `github.com/rmojgani/PhysicsAwareAE`.

in the domain $(x,t) \in [0,2.5] \times [0,t_{\max}]$, equipped with initial conditions $w(x,0) = w_0(x)$, and Dirichlet boundary conditions at $x = 0$ and $x = 2.5$, where $\nu = 10^{-3}$, $w(x,0) = 0.8 + 0.5\ e^{-(x-0.5)^2/0.1^2}$, $w(0,t) = w(2.5,t) = 0$, for $t = [0,t_{\max}]$. An implicit second order time discretization is used with $\Delta t = 8 \times 10^{-3}$ and space is uniformly discretized where $\Delta x = 10^{-2}$. In the proposed architecture, the rank-1 time-varying grid ($r = 1$), representing the low-rank registration-based auto-encoder, is trained as in (2.15) with $k_r = 4$. In this problem, $v_{\min} = \Delta x_{\min} = 10^{-3}$, $\boldsymbol{\Gamma}_1 = \boldsymbol{D}_{xx}$ and $\boldsymbol{\Gamma}_2 = \boldsymbol{D}_{tt}$, and $\boldsymbol{D}_{xx}$ and $\boldsymbol{D}_{tt}$ are second derivative matrices in space and time. The grid bases are down-sampled to 15 and 5 control points in space and time, respectively.

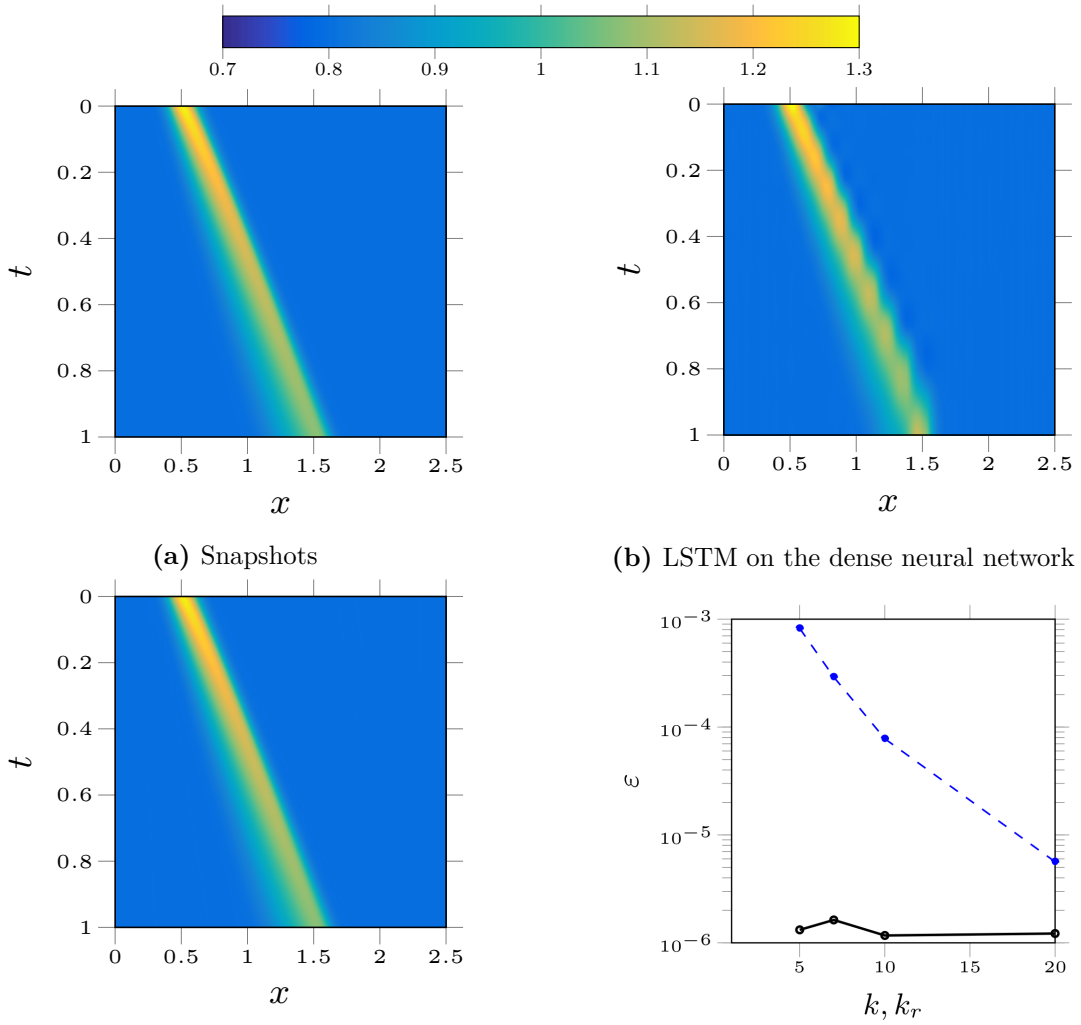
The snapshots and output of the LSTM network on the NN and low-rank registration-based auto-encoder are compared in Fig. 5.32 for a case of $k = 10$. The error for a range of sizes of LSTM, k and k_r, are plotted in Fig. 5.32d. In this case, the error has reached it's plateau at $k_r \leq 4$, showing the low-dimensionality of the snapshots on the trained auto-encoder.

Second-order wave equation

Consider the one-dimensional second-order wave equation,

$$\frac{\partial^2 w(x,t)}{\partial t^2} - \frac{\partial^2 w(x,t)}{\partial x^2} = 0, \tag{5.10}$$

in the domain $(x,t) \in [0,1] \times [0,1]$, equipped with initial conditions $w(x,0) = w_0(x)$, and Dirichlet boundary conditions at x_a, and x_b, where $w(x,0) = e^{-(x-0.5)^2/0.1^2}$, $w(0,t) = w(1,t) = 0$, for $t = [0,1]$. An implicit second-order time-discretization is used with $\Delta t = 2.5 \times 10^{-3}$ and space is uniformly discretized where $\Delta x = 10^{-2}$. The grid manifold is identified with the following parameters: the time-varying grid is of rank-2 ($r = 2$), the reconstruction on the learned manifold is of rank-2 ($k_r = 2$), $v_{\min} = \Delta x_{\min} = 10^{-3}$, $\gamma_x = \gamma_t = 10$, and the size of the grid bases, in both space and time, are down-sampled to 15 control points. The solution of wave equation and output of the reduced LSTMs are plotted in Fig. 5.33, showing the increase in the performance of the low-rank registration-based auto-encoder compared to the traditional architecture.



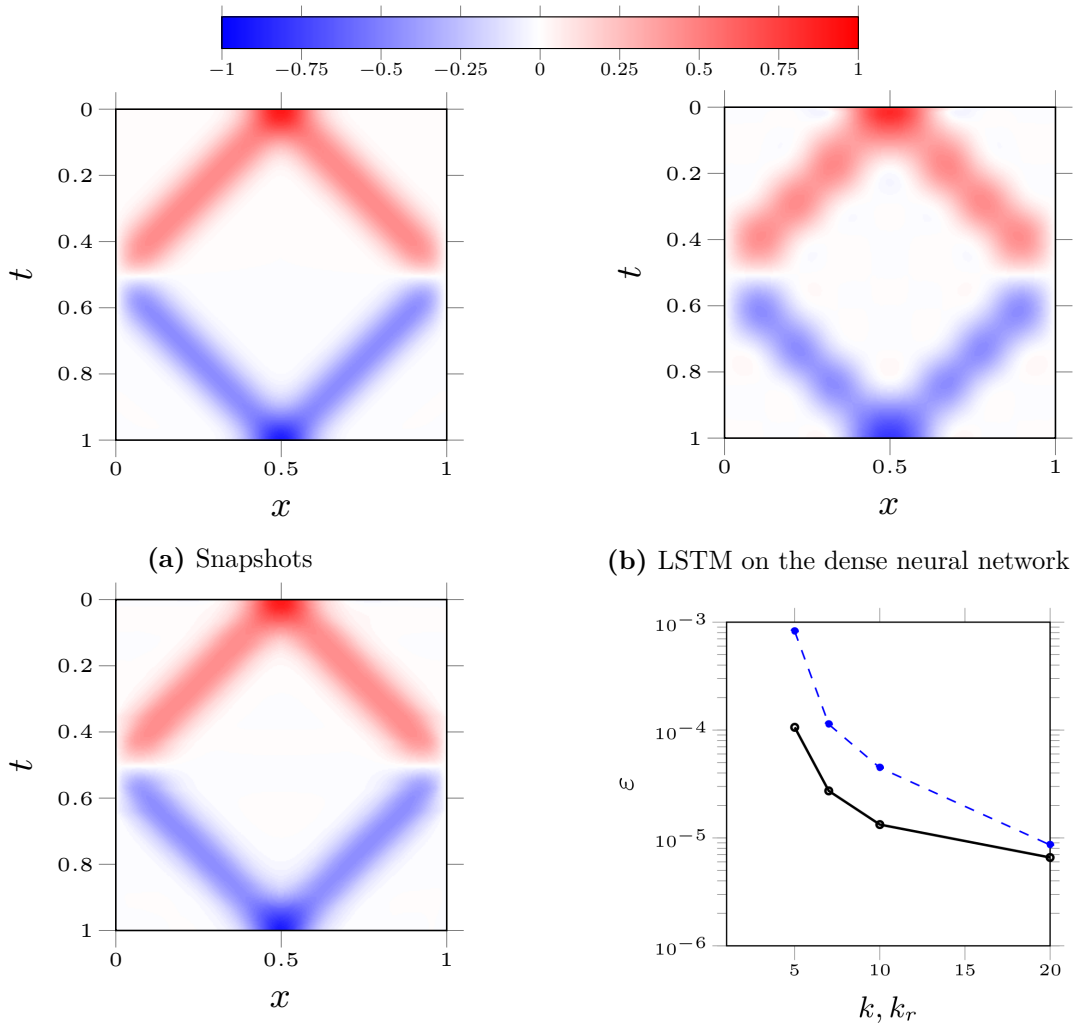
(a) Snapshots

(b) LSTM on the dense neural network

(c) LSTM on the low-rank registration-based auto-encoder

(d) The error of LSTM

Figure 5.32: LSTM with $k = 10$ approximating the solution of the viscous Burgers' equation for $t = [0, 1]$. In (d), dashed blue line corresponds to the LSTM on dense neural network and solid black line corresponds to LSTM on low-rank registration-based auto-encoder.



(a) Snapshots

(b) LSTM on the dense neural network

(c) LSTM on the low-rank registration-based auto-encoder

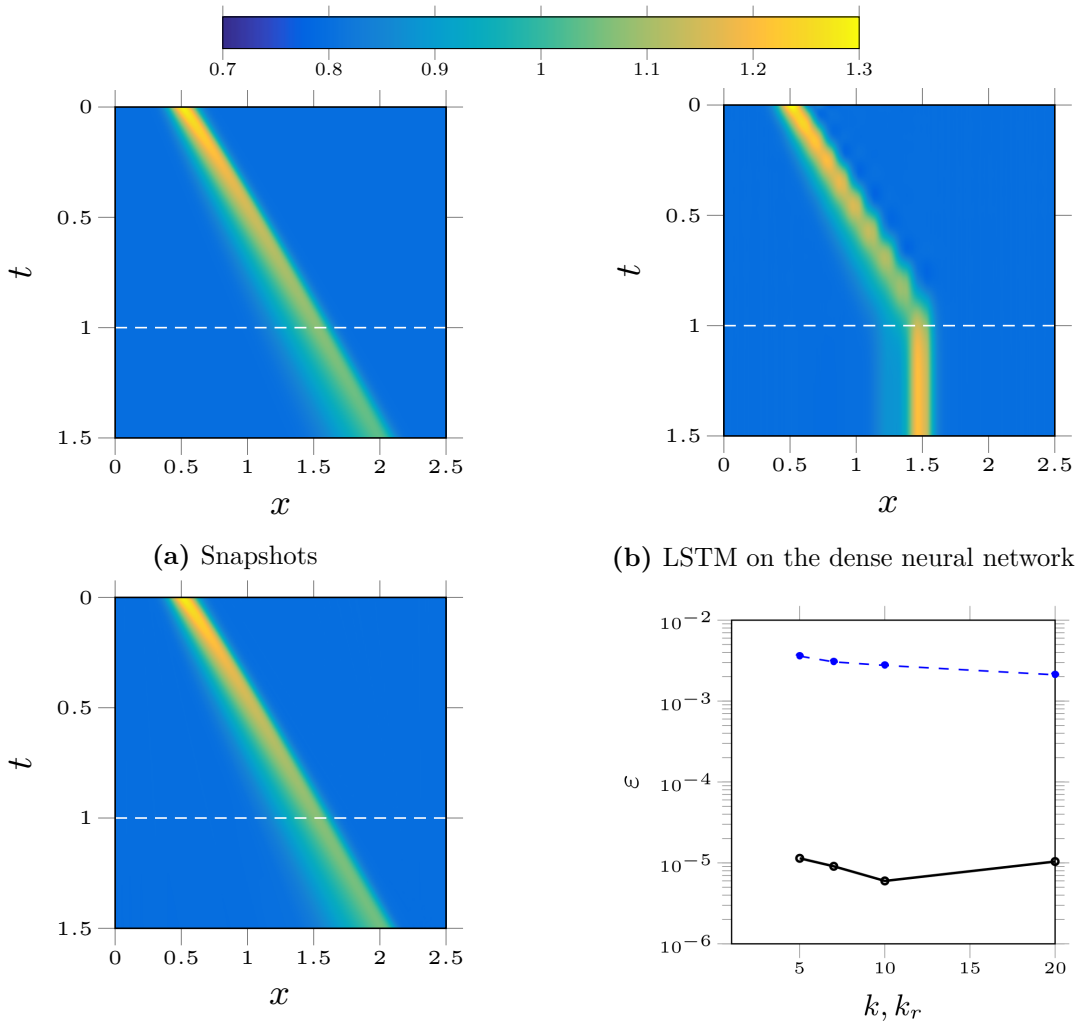
(d) The error of LSTM

Figure 5.33: LSTM with $k = 5$ approximating the solution of the second-order wave equation for $t = [0, 1]$. In (d), dashed blue line corresponds to the LSTM on dense neural network and solid black line corresponds to LSTM on low-rank registration-based auto-encoder.

5.2.6 Prediction of one-dimensional problems using LSTM ROMs

Consider the neural network-based ROMs trained on the solution of the Burgers' equation in $t = [0, 1]$. In this section, we evaluate the trained models in the prediction range of $t = [1, 1.5]$. To extend the identified manifold of the proposed low-rank registration-based auto-encoder, $\boldsymbol{V}_x$ is extrapolated linearly in time (similar to §5.2.4).

In Fig. 5.34, the models are extended beyond the training range, i.e. $t = [1, 1.5]$, and the LSTM solutions, Fig. 5.34b and Fig. 5.34c, are compared to the Burgers' solution in Fig. 5.34a. The error of LSTMs-based ROMs trained on the densely connected neural network layer and low-rank registration-based auto-encoder layer for different sizes of LSTM, k and k_r, are compared in Fig. 5.34d. As expected, the neural network auto-encoder cannot predict the convection underlying the physics of the problem outside the training range (Fig. 5.34b); however, by levering the convection identified in the low-rank registration-based auto-encoder, the LSTM trained on the low-dimensional manifold realized by the proposed approach leads to a solution much closer to the solution of the Burgers' equation. In the predictive regime, even increasing the dimension of the latent variable (k) does not decrease the error.



(a) Snapshots

(b) LSTM on the dense neural network

(c) LSTM on the low-rank registration-based auto-encoder

(d) The error of LSTM

Figure 5.34: LSTM with $k = 10$ approximating and predicting the solution of the Burgers' equation for $t = [0, 1.5]$. The LSTM models are trained for $t = [0, 1]$.

5.3 Stabilization of time-varying ROMs

In this section, we demonstrate the performance of the proposed stabilization method on several time-invariant, time-periodic and general time-varying linear systems: a model of the international space station (ISS) perturbed by an impulse (LTI), a synthetic time-varying system based on the ISS system matrices, a second order wave equation on a time-varying grid, and a mechanical vibrations system. Robustness of the proposed algorithms with respect to the inputs and system parameter is also presented for the mechanical vibrations system. Finally, the computational costs of the experiments and the complexity of the method are discussed.

The HFMs are solved numerically using standard finite element and time discretization approaches. The spatial bases are constructed via POD of the collected snapshots, unless otherwise stated. The bases are time-independent in §5.3.1 and §5.3.2 and they are time-varying in §5.3.3 through §5.3.5.

In all the cases the, reduced systems are generated off-line and the stabilization algorithm is utilized to derive the "stabilized ROMs" via the proposed approach. The transient error ε, is defined as the distance between the HFM and the ROM solution between the initial time and the corresponding time step n_t, i.e.

$$\varepsilon = \frac{\sum_{n=0}^{n_t} \|\boldsymbol{y}[n] - \hat{\boldsymbol{y}}_r[n]\|_2}{\sum_{n=0}^{n_t} \|\boldsymbol{y}[n]\|_2}. \tag{5.11}$$

The constrained nonlinear optimization problems are solved using in Matlab using the interior-point algorithm implementation of the Matlab optimization toolbox [178]. The gradients are approximated using first-order finite-differences.

5.3.1 Time-invariant international space station

For the first example, we consider the dynamical system associated with the structural vibrations of the Russian service module (component 1R) of the international space station (ISS) [179, 180]. The discretized system corresponds to a time-invariant dynamical system with a sparse system matrix and the associated input and output matrices. Following [181], the HFM is solved using a backward Euler time integration (first order implicit) scheme. The initial condition is assumed to be the response of the system to an impulse of $\boldsymbol{u}(t) = \left(1 \times 10^4\right) \delta_{t=0}$ from an stationary condition,

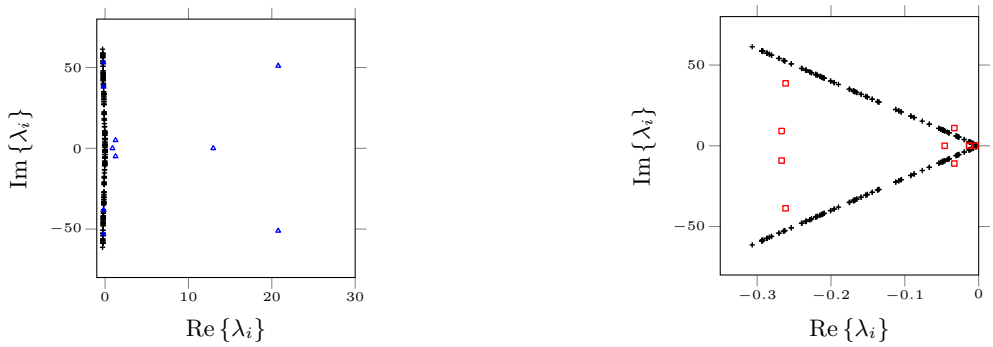


Figure 5.35: Eigenvalues of the the HFM of the international space station (black plus markers), and the corresponding the POD Galerkin ROM (blue triangle markers) and the stabilized ROM (red square markers). For all ROMs $k = 10$.

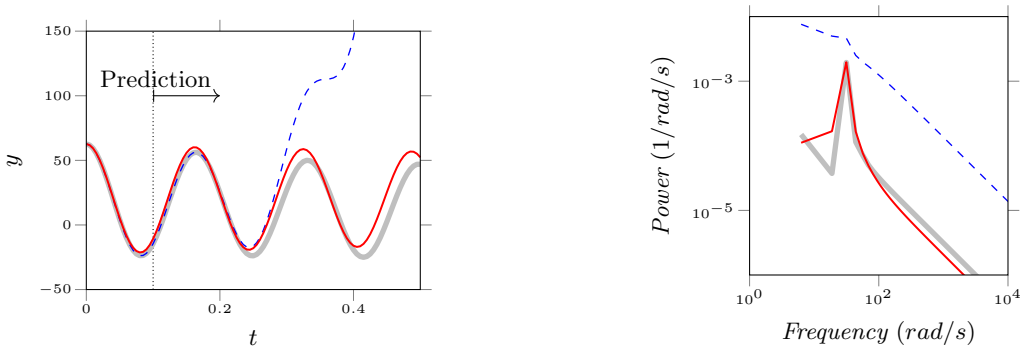
i.e. $\boldsymbol{w}[0] = \boldsymbol{0}$. In the off-line stage, the snapshot of $\boldsymbol{M} \in \mathbb{R}^{270\times 2000}$ is collected where $t = 0.1$ and $\Delta t = 5 \times 10^{-5}$. Although the solution of the Galerkin ROM with $k = 10$ appears bounded within the training stage, i.e. $t \in [0, 0.1]$, the system actually contains 6 unstable eigen-modes (two complex conjugate pairs and two real eigenvalues as in Fig. 5.35a). The instability of the system becomes apparent in the predictive regimes, where the solution of the ROM clearly diverges from the HFM as shown in Fig. 5.36a.

The ROM of size $k = 10$ is stabilized using the proposed algorithm summarized in Alg. 4. The six largest singular values of the ROM's system matrix is calibrated to stabilize the system matrix ($l = 6$), leading to $\hat{\boldsymbol{A}}_r$ and the corresponding eigenvalues in Fig. 5.35b. It is noteworthy that the stable complex conjugate pairs of eigenvalues are resolved without any assumptions on types of the eigenvalues, in contrast to the eigenvalue reassignment methods in [81].

Figure 5.36a shows the solution of the stabilized ROM when $l = 6$, i.e. controlling the first 6 energy-growing modes. The Power Spectral Density (PSD) of the HFM solution, the original ROM and the stabilized ROM are also compared in Fig. 5.36b, showing the convergence of the stabilized ROM to the HFM in the frequency domain, as well as the time domain.

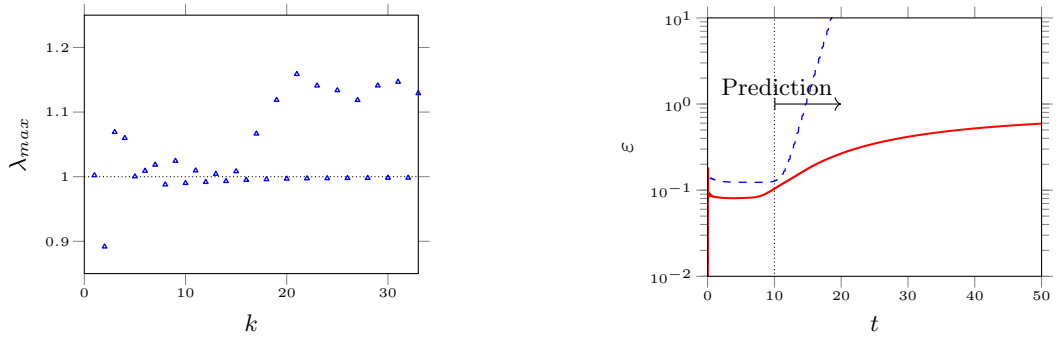
5.3.2 Time-varying international space station

To obtain a linear time-periodic model, the constant system matrix of module 1R of ISS (§5.3.1) are modified to be time-periodic as in [147]. The time-varying system matrix is defined as $\boldsymbol{A}[n] = \alpha[n-1]\Delta t\boldsymbol{A}$, where $t[n] = (n-1)\Delta t$ and $\alpha[t] = \sin(\omega_0 t)$. In this example $\omega_0 = 8\pi$ and the initial



(a) (b)

Figure 5.36: International space station excited by an impulse. (a) Output of the system in time; (b) Spectral density of system output. The high-fidelity model (thick gray line), the original Galerkin ROM (dashed blue line) and the stabilized ROMs (solid red line). For all ROMs $k = 10$.



(a) (b)

Figure 5.37: Time-varying international space station. (a) The maximum modulus of eigenvalue the monodromy matrices of different sizes of ROM; (b) The transient error of the Galerkin ROM (dashed blue line) and the stabilized ROMs (solid red line).

condition is $\boldsymbol{w}[0] = \mathbf{1} \in \mathbb{R}^{270\times 1}$, and the input matrix is assumed to be zero $\boldsymbol{B} = \mathbf{0} \in \mathbb{R}^{270\times 270}$. Consider the output matrix of $\boldsymbol{C} = [\boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{c}_3] \in \mathbf{R}^{3\times 270}$ as presented in §5.3.1, we define the output matrix in the current section as $\boldsymbol{c}_1 + \boldsymbol{c}_2 \cos(\omega_0 t) + \boldsymbol{c}_3 \cos(\omega_0 t) \in \mathbf{R}^{1\times 270}$.

The HFM is solved using a backward Euler time integration (first order implicit) scheme, where $\Delta t = 5 \times 10^{-3}$. The snapshot, $\boldsymbol{M} \in \mathbb{R}^{270\times 1000}$, is collected for 20 time periods ($t_{max} = 5$). The LTP ROM is then constructed using the Galerkin projection. The maximum modulus of the eigenvalues of the monodromy matrix of the reduced system for different ROM sizes are plotted in Fig. 5.37a, showing many of the ROMs are unstable. We choose the ROM of size $k = 21$ with $\lambda_{max} = 1.1588$ to demonstrate our stabilization method. Every one out of ten system matrices ($\Delta n = 10$) with $l = 8$ are calibrated in this example. The maximum modulus of the eigenvalue of the monodromy matrix of the stabilized reduced order matrix is within the unit circle, $\lambda_{max} = 0.9983$.

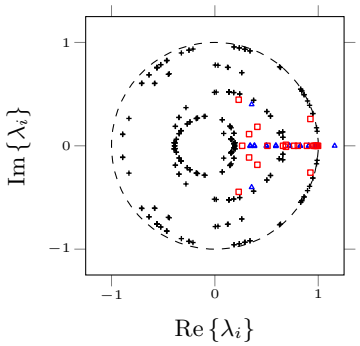


Figure 5.38: Eigenvalue of the monodromy matrices of time-periodic ISS model, HFM (black plus markers), the Galerkin ROM (blue triangle markers) and the stabilized ROM (red square markers).

5.3.3 Wave equation on a time-varying grid

In this section, the proposed stabilization method is applied to a second-order wave equation in a single dimension on a time-varying grid. Consider

$$\frac{\partial^2 w(x,t)}{\partial t^2} = \frac{\partial^2 w(x,t)}{\partial x^2}, \tag{5.12}$$

in the domain $(x,t) \in [0,1] \times [0,1]$, where w is the state parameter. We set initial conditions, $w_0 = e^{-((x-0.5)/0.05)^2}$, and $\partial w/\partial t(x,0) = 0$ and Dirichlet boundary conditions $w(0,t) = w(1,t) = 0$.

Equation (5.12) is discretized using an implicit second-order discretization in time and second-order central discretization in space leading to a "4-level scheme", i.e.

$$\left(2\boldsymbol{I} - \Delta t^2 \boldsymbol{D}_{xx}\right) \boldsymbol{w}[n+1] = 5\boldsymbol{w}[n] - 4\boldsymbol{w}[n-1] + \boldsymbol{w}[n-2], \tag{5.13}$$

where $\boldsymbol{D}_{xx}$ discrete approximation of the second derivative on equidistant grid points.

We construct the reduced order model of equation (5.13) on a time-varying grid, i.e.

$$\begin{aligned} &\boldsymbol{U}^T[n+1]\left(2\boldsymbol{I} - \Delta t^2 \boldsymbol{D}_{xx}\right)\boldsymbol{U}[n+1]\boldsymbol{w}_r[n+1] = \\ &5\boldsymbol{U}^T[n+1]\boldsymbol{U}[n]\boldsymbol{w}_r[n] - 4\boldsymbol{U}^T[n+1]\boldsymbol{U}[n-1]\boldsymbol{w}_r[n-1] \\ &+ \boldsymbol{U}^T[n+1]\boldsymbol{U}[n-2]\boldsymbol{w}_r[n-2], \end{aligned} \tag{5.14}$$

where $\boldsymbol{w}[n] \approx \boldsymbol{U}[n]\boldsymbol{w}_r[n]$ is approximation of the state parameter on the uniform grid.

It is noteworthy that the reduced bases on the time-varying grid as presented in this thesis is

different from what often appears in literature [37, 182]. The motivation of this choice is out of the scope of this thesis and is discussed in [39].

Letting

$$\boldsymbol{z}[n] = \begin{bmatrix} \boldsymbol{w}_r[n] \\ \boldsymbol{w}_r[n-1] \\ \boldsymbol{w}_r[n-2] \end{bmatrix} \in \mathbb{R}^{3k}, \tag{5.15}$$

the first order equivalent of system of (5.14) reads

$$\boldsymbol{z}[n+1] = \boldsymbol{A}[n]\boldsymbol{z}[n], \tag{5.16}$$

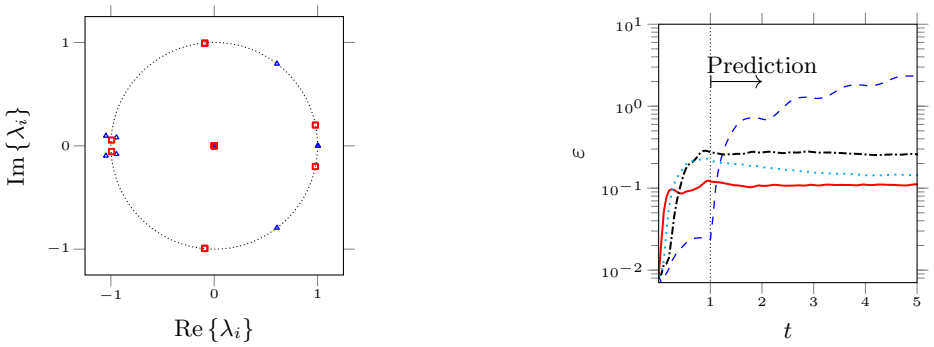
where

$$\boldsymbol{A}[n] = \begin{bmatrix} 5\boldsymbol{F}[n]\boldsymbol{U}^T[n+1]\boldsymbol{U}[n] & -4\boldsymbol{F}[n]\boldsymbol{U}^T[n+1]\boldsymbol{U}[n-1] & \boldsymbol{F}[n]\boldsymbol{U}^T[n+1]\boldsymbol{U}[n-2] \\ \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \end{bmatrix}, \tag{5.17}$$

and $\boldsymbol{F}^{-1}[n] := \boldsymbol{U}^T[n+1]\left(2\boldsymbol{I} - \Delta t^2 \boldsymbol{D}_{xx}\right)\boldsymbol{U}[n+1]$. We use Galerkin projection in each time step, i.e. $\boldsymbol{U}[n] = \boldsymbol{U}[n]$.

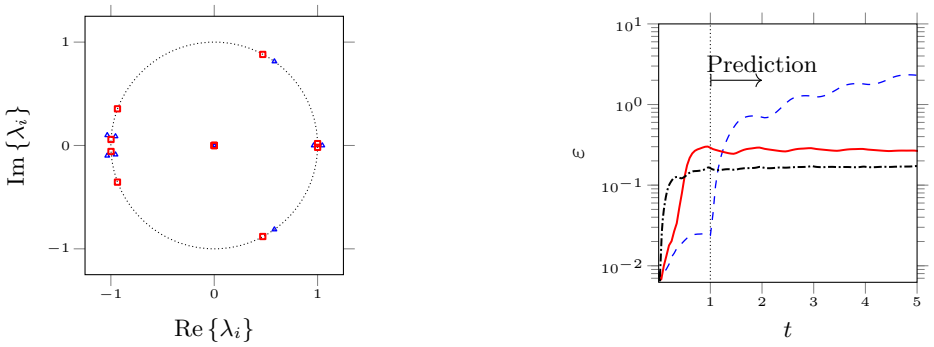
The HFM of (5.12) is solved on two uniform grids of different sizes and the results are compared. In the coarse problem $N_x = 5 \times 10^2$ and $N_t = 2 \times 10^3$ and in the fine problem $N_x = 5 \times 10^3$ and $N_t = 2 \times 10^4$. The $k = 4$ reduced order system of the wave equation is formed as in (5.16), a POD Galerkin method resulting in $\boldsymbol{A}[n] \in \mathbb{R}^{12\times 12}$. Some of the eigenvalues of the monodromy matrix lie outside the unit circle (Fig. 5.39a and Fig. 5.40a), which lead to the exponential growth of the state parameter and therefore the corresponding error (Fig. 5.39b and Fig. 5.40b).

The system is then stabilized using the proposed method of Alg. 4 with $l = 4$ and different choices of Δn (choosing the first 4 leading singular values of the selected matrices). The stabilization method reassigns the eigenvalues of the monodromy matrix to their optimal and stable locations (Fig. 5.39a and Fig. 5.40a), where the ROM error is minimized using (4.16). The stabilized reduced system is constructed in one period using selected point-wise system matrices. The transient error of the original ROM and the stabilized over a longer time interval are compared in Fig. 5.39b and



(a) **(b)**

Figure 5.39: Second-order wave equation on time-varying grid. (a) Eigenvalue of the monodromy matrix of the ROM on the time-varying grid (blue triangle markers) and the corresponding stabilized ROM with $\Delta n = 25$ (red square markers); (b) Transient error of the POD Galerkin ROM (dashed blue line) and the stabilized ROM (solid red line for $\Delta n = 25$, dotted cyan line for $\Delta n = 100$ and dashed-dotted black lines for $\Delta n = 400$).



(a) **(b)**

Figure 5.40: Second-order wave equation on time-varying grid. (a) Eigenvalue of the monodromy matrix of the ROM on the time-varying grid (blue triangle markers) and the corresponding stabilized ROM with $\Delta n = 1000$. (b) Transient error of the POD Galerkin ROM (dashed blue line) and the stabilized ROM (solid red line for $\Delta n = 1000$ and dashed-dotted black line for $\Delta n = 400$).

Fig. 5.40b. Note that enforcing the stability guarantee is at the cost of higher error in the first time-period of the solution. However, this compromise leads to lower error in the predictive regime.

5.3.4 Time-varying mechanical vibrations

In this section, the proposed stabilization method is applied to a variation of an academic linear structural dynamics system, a system of mass-spring-damper as introduced in [1]. This model is often used in stability studies of reduced order models as a benchmark problem, e.g. [125, 183]. The system is comprised of 4 masses connected to each-other with total of 6 springs and 4 dampers (Fig. 5.41). The original problem is a parameter-varying LTI system, where $k_5 = 1 + 2\alpha$, $d_5 = \alpha$

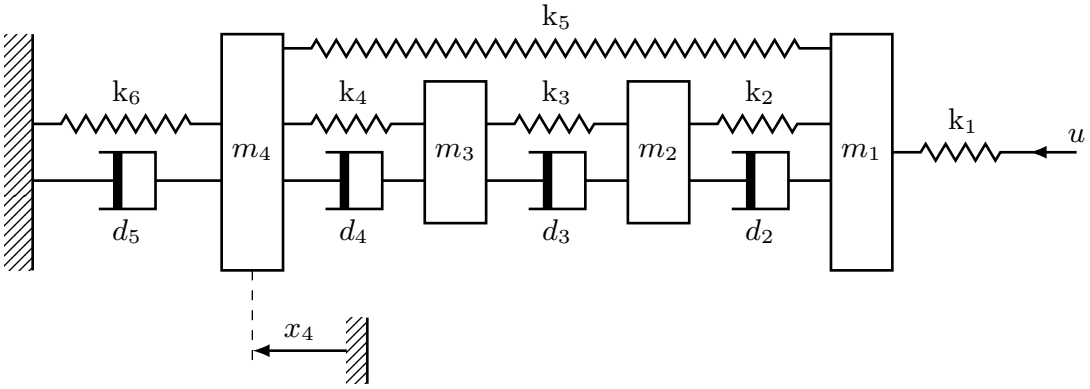


Figure 5.41: The schematic of the spring-mass-damper system.

Table 5.2: Masses, damper and spring coefficients [1].

mass	(kg)	**Damper**	$(N.s/m)$	**Spring**	(N/m)
m_1	1	d_2	0.1	k_1	27
m_2	5	d_3	0.4	k_2	9
m_3	25	d_4	1.6	k_3	3
m_4	125	d_5	α	k_4	1
				k_5	$1+2\alpha$
				k_6	$2+2\alpha$

and $k_6 = 2 + 2\alpha$, with all the values reported in Table 5.2.

The equations of motion are in a linear semi-discrete system of the form $\boldsymbol{M}\ddot{\boldsymbol{x}}(t) + \boldsymbol{E}(t)\dot{\boldsymbol{x}}(t) + \boldsymbol{K}(t)\boldsymbol{x}(t) = \boldsymbol{B}\boldsymbol{u}(t)$, where $\ddot{\boldsymbol{x}}(t) = \partial^2\boldsymbol{x}/\partial t^2$, $\dot{\boldsymbol{x}}(t) = \partial\boldsymbol{x}/\partial t$ and $\boldsymbol{x}(t) \in \mathbb{R}^4$ is the position vector. The mass matrix is denoted by $\boldsymbol{M}$. The damping and stiffness matrices, denoted respectively by $\boldsymbol{E}(t)$ and $\boldsymbol{K}(t)$, are functions of $\alpha\,(t) \in [0, 2]$ as in Fig. 5.42a. The input is a unit step acting on k_1 and the output is the displacement of the 4^{th} mass, m_4.

The first-order state-space representation of the system is discretized using a backward Euler time integration (first order implicit) scheme with $\Delta t = 0.1$, leading to $\boldsymbol{A}_r[n] \in \mathbb{R}^{8\times8}$ and $N_t = 5000$, where $t \in [0, 500]$. The Krylov subspace method of model order reduction is used in reduction of the original LTI variation of the problem in [1, 125]. In this example, the reduced system matrices of $k = 4$ are constructed at each time step using the two-sided Lanczos moment matching algorithm [184]. More compact representation of the projection can be built considering that the projection matrices span over the union of all the projection sub-spaces [143], however, it's application is beyond the scope of this thesis.

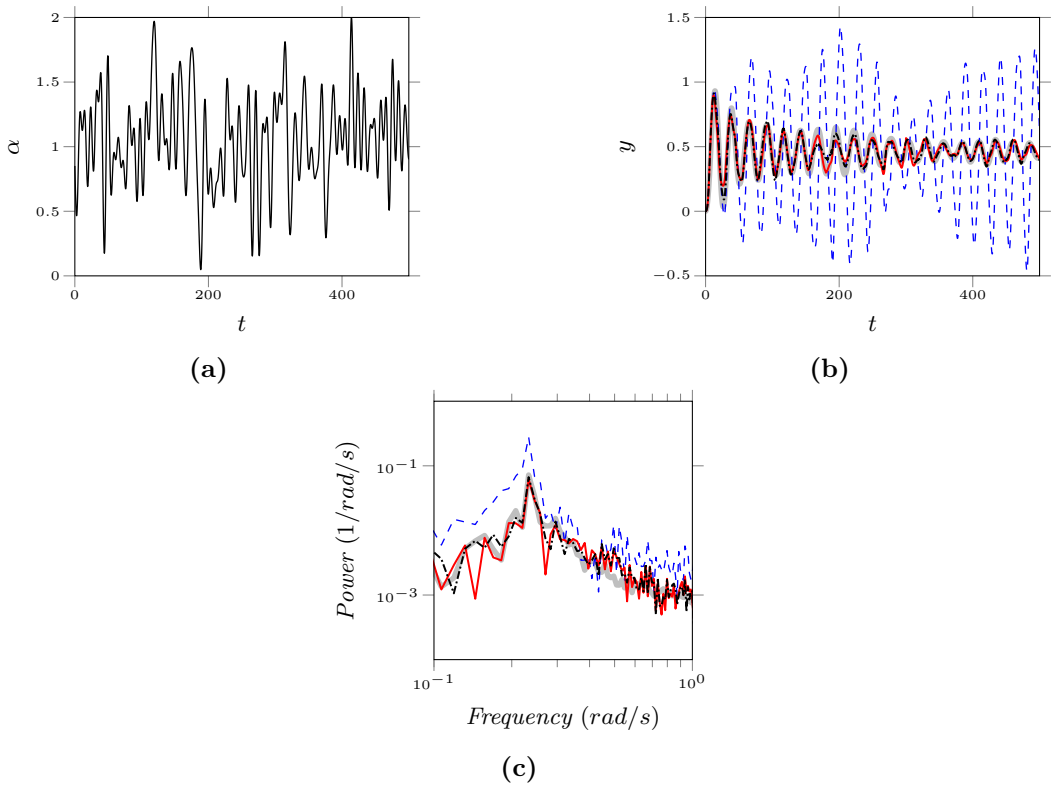


Figure 5.42: The time-varying mechanical vibrations system. (a) Time-varying system coefficient, $\alpha[t]$; (b) Output of the system in time; (c) Power spectral density (PSD) of system output. The high-fidelity model (thick gray line), the original unstable Krylov ROM (dashed blue line) and the stabilized ROMs (solid red lines for $\Delta n = 100$ and dashed-dotted black lines for $\Delta n = 10$). For all ROMs $k = 4$ and $l = 3$.

The system is excited by a unit step input and therefore its output decays over time, however, the Krylov based ROM predicts an intermittent growth of the output (Fig. 5.42b). The original Krylov ROMs are stabilized using Alg. 4. The objective in (4.23) is augmented to penalize abrupt changes in the norm of the point-wise system matrices, i.e. $\sum_{n=1}^{N_t} \|\hat{\boldsymbol{A}}_r[n] - \hat{\boldsymbol{A}}_r[n-1]\|_2$. In this formulation of the problem, the norm constraint in (4.23) is set to equal to norm of the state transition matrix of the HFM. The ROM is controlled using rank-3 stabilizer matrices ($l = 3$) with $\Delta n \in \{10, 100\}$. The HFM, the original ROM and the aforementioned stabilized ROMs are compared in Fig. 5.42b and Fig. 5.42c, showing that a small enough Δn can be found to ensure stability and convergence of the ROMs to the HFM, in both the temporal and frequency domains. A more comprehensive study of the sensitivity of the ROM error to the hyper-parameters and convergence of the optimization problem are presented in §5.3.5.

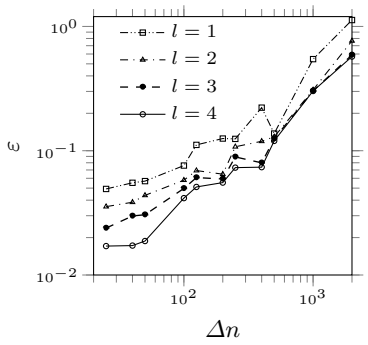


Figure 5.43: The time-varying mechanical vibrations system. The error of the stabilized ROMs ($k = 4$) versus Δn for $l \in \{1, 2, 3, 4\}$.

5.3.5 Hyper-parameter selection and uncertainty quantification

In this section, the sensitivity of the stabilized ROM with respect to its hyper-parameters, system parameters, and inputs is summarized. In particular, we study the influence of the of the hyper-parameters, Δn and l, on the controller/stabilizer matrices, and perform a series of uncertainty quantification experiments to demonstrate the robustness of the proposed approach. For the sake of brevity, we limit this study to the time-varying mechanical vibration system from §5.3.4. The standard Monte-Carlo sampling is continued to converge to 0.1% of change in the median of the probability of the QoI. The QoI is the error of the original Krylov ROM and the stabilized ROM, ε.

Hyper-parameters selection

In this section we study the influence of the hyper-parameters on ROM accuracy. More specifically, we pick a ROM of size $k = 4$, and compute ROM error in the parameter range $l \in \{1, 2, 3, 4\}$ and $\Delta n \in [20, 2000]$. Each ROM is stabilized using the same procedure described in the preceding sections.

The results of this study are illustrated in Fig. 5.43. As expected, increasing intermittency of the activation of the controller (decreasing Δn) and calibrating a larger number of the singular values (increasing l) decreases the ROM error. Moreover, the ROM error decreases almost monotonically for $l > 1$.

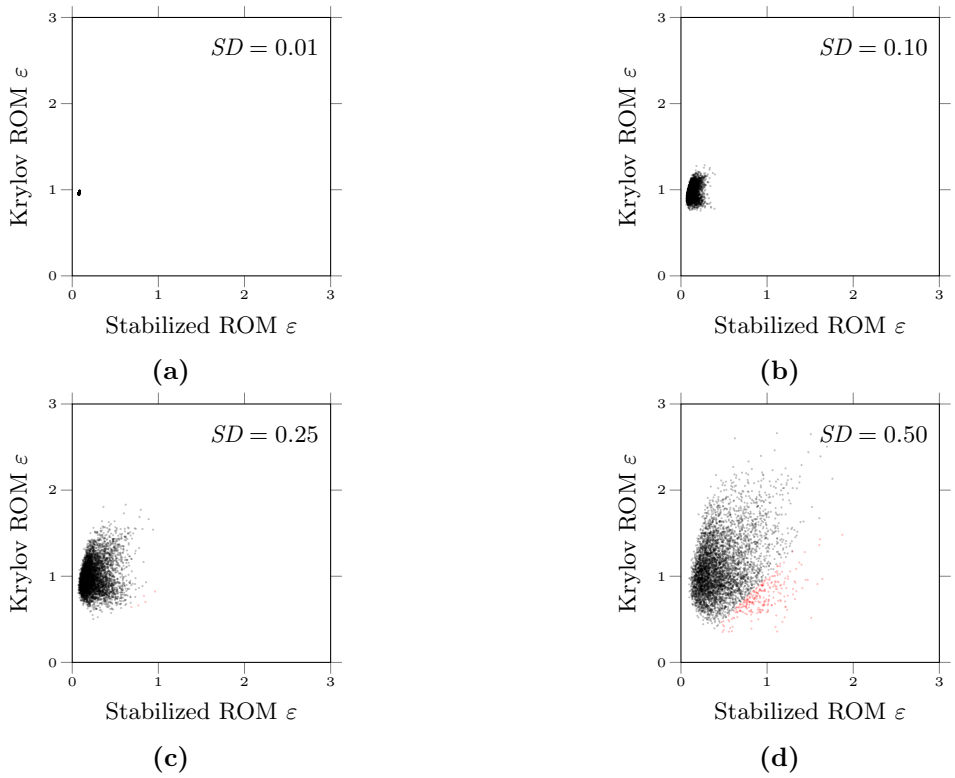


Figure 5.44: The time-varying mechanical vibrations system. Error of the Krylov ROM versus the stabilized ROM in the Monte-Carlo samples given different standard deviation in uncertainty of the input vector, $\boldsymbol{u}[n] \sim \mathcal{N}(\mu = 1, SD)$, where $SD \in \{0.01, 0.10, 0.25, 0.50\}$. The black markers represent the cases where the stabilized ROM is more accurate than the Krylov ROM and red markers represent the opposite.

Input signal uncertainty

In this section, uncertainty is assigned to the input signal acting on k_1. The input is sampled from a normal distribution with a unit mean value, $\mu = 1$, and different values of standard deviation, SD, i.e. $\boldsymbol{u}[n] \sim \mathcal{N}(\mu = 1, SD)$.

The Krylov ROM system subspaces and corresponding matrices are constructed once during the off-line stage. The stabilized ROM system matrices, $\hat{\boldsymbol{A}}_r[n]\ \forall n \in \{0, 1, 2, \cdots, M\}$, are identified for the case of unit step input using Alg. 4, where $\Delta n = 100$ and $l = 3$, as in §5.3.4. Subsequently, the Krylov ROM and stabilized ROM error calculated for each sample of the Monte-Carlo simulation and are plotted in Fig. 5.44. The median and the 25^{th} and 75^{th} percentiles of the samples for a range of the uncertainties in the input vector is summarized in Fig. 5.45, illustrating that the stabilized ROMs are significantly more accurate compared to the Krylov ROMs in a wide range of uncertain input vectors.

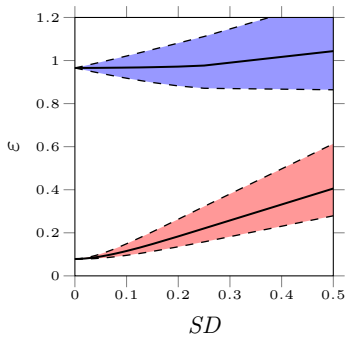


Figure 5.45: The time-varying mechanical vibrations system. Krylov (blue) and stabilized ROM (red) error uncertainty probability distribution propagated from a uncertain input vector. The solid black line indicates the median and dotted lines indicate 25^{th} and 75^{th} percentiles the ROM.

System parameter uncertainty

In this section, a normally distributed uncertainty is assigned to the system parameter $\alpha'[n]$, and therefore k_5, d_5 and k_6 in the LTV system of mechanical vibrations (see Table 5.2), i.e. $\alpha'[n] \sim \alpha[n] + \mathcal{N}(\mu = 0, SD)$. There exist a wide variety of approaches for generating Krylov-based ROMs for parameter varying problems. For the purpose of this study, only two of the simplest ones are considered. In the first approach, the Krylov subspaces are computed only once at a reference system parameter, $\alpha[n]$. While in the second approach, Krylov subspaces are constructed at each of the sampled $\alpha'[n]$.

Therefore, the off-line computational cost of the first approach is significantly lower in the multi-query setting of the UQ experiments. The distributions of the error of the Krylov ROMs and stabilized ROMs are compared in Fig. 5.46. In Fig. 5.47, the Monte-Carlo samples and more detailed distribution of the error are compared for $SD = 0.1$. As expected, the proposed stabilization approach improves significantly both the value and uncertainty of the ROM error, irrespective of the particular approach used to generate the Krylov ROM subspace.

5.3.6 Computational cost and scaling

In this section, the computational complexity and cost of the proposed stabilization method is discussed. An upper bound on the cost of each iteration of the proposed optimization problem is $\mathcal{O}\left(k^3 M^3\right)$ and thus it is bounded cubically with the dimension k of the LTV ROM and the number of time varying system matrices, M. For a general time-varying system $M = N_t$, and for

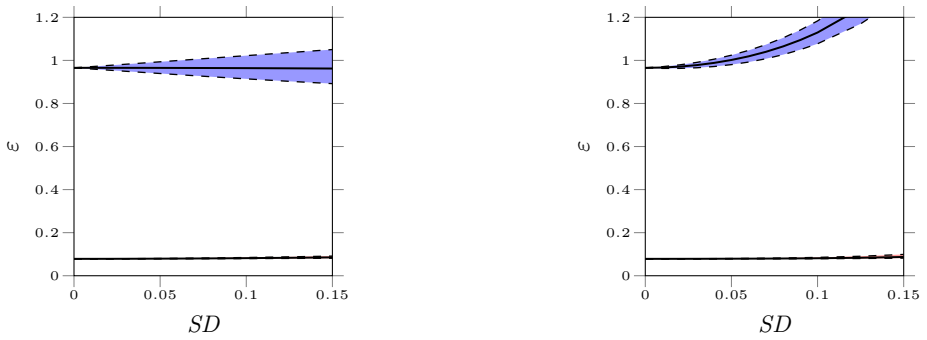


Figure 5.46: The time-varying mechanical vibrations system. Krylov (blue) and stabilized ROM (red) error uncertainty probability distribution propagated from a uncertain input vector. The solid black line indicates the median and dotted lines indicate 25^{th} and 75^{th} percentiles the ROM. (a) The Krylov subspaces are identified once at $\alpha[n]$; (b) The Krylov subspaces are identified at each $\alpha'[n]$.

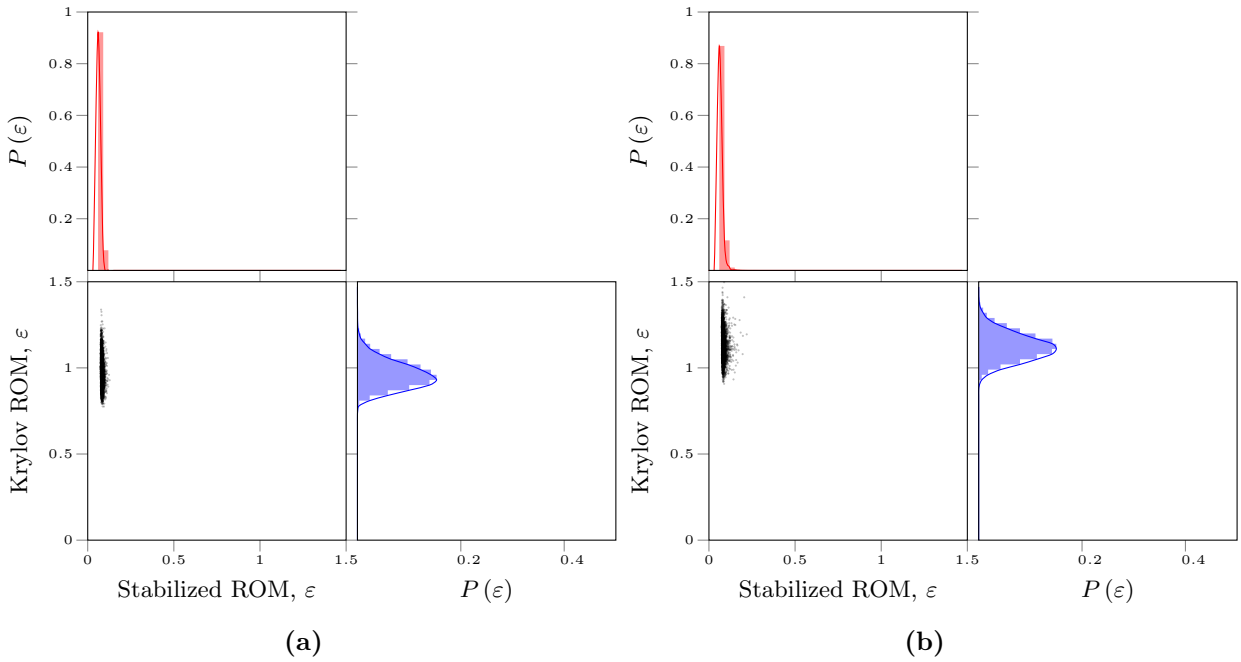


Figure 5.47: The time-varying mechanical vibrations system. (a) The Krylov subspaces are identified once at $\alpha[n]$; (b) The Krylov subspaces are identified at each $\alpha'[n]$. Error of the Krylov ROM versus the stabilized ROM in the Monte-Carlo samples where the system parameter $\alpha'[n] \sim \alpha[n] + \mathcal{N}(0, 0.1)$. Moreover, Krylov (blue) and stabilized ROM (red) error uncertainty probability distribution associated with this simulation are respectively plotted on the right and top axes.

time-periodic and time-invariant systems, $M = p$ and $M = 1$, respectively. However, by introducing the sporadic control (Δn) of the l leading singular values of the reduced point-wise system matrices ($\mathbf{\Sigma_l}[n]$), the cost of the stabilization method is reduced to $\mathcal{O}\left(l^3 M^3/\Delta n^3\right)$. The cost of all other steps in Alg. 4 are independent of the dimension of the HFM. The cost of the proposed algorithm is, therefore, significantly lower compared to competing approaches which typically scale with the size of the HFM. For example, the balanced truncation [147] approach scales as $\mathcal{O}\left(N^3 M\right)$, where N is the size of the HFM, making it prohibitive for realistic large-scale problems.

Table 5.3 summarizes the wall-clock computational costs of evaluating the HFM, the ROM and performing the stabilization procedure. For the wave equation problem considered in §5.3.3, the online computational speed-up is approximately 800, while the off-line stabilization costs are approximately $1/20^{th}$ of the cost of a single evaluation of the HFM. Higher speedups can be expected for more realistic, large-scale engineering applications. The wave equation considered in this paper is one-dimensional and thus, already very efficient to solve even before any model reduction. Moreover, it is emphasized that the target application of our proposed method is multi-query applications, such as optimization, where the costs of the offline stabilization procedure must be weight against potentially hundreds, if not thousands, of queries of the HFM or the ROM. Finally, for the two other problems considered in this work – the ISS and the mechanical vibrations system – the corresponding HFMs are too small to fairly demonstrate the computational advantages of the proposed methodology. These small-scale problems are included to demonstrate the mathematical foundations of the proposed methodology, and to preform an exhaustive UQ study; something that would not be possible with a realistic large-scale HFM.

Table 5.3: Summary of computational costs.

System	**Type**	M	N_t	N	k	l	Δn	**HFM (sec.)**	**Stabilization (sec.)**	**ROM (sec.)**
ISS (§5.3.1)	LTI	1	2.0×10^3	270	10	6	1	7.5	7.8	8.8×10^{-2}
ISS (§5.3.2)	LTP	50	1.0×10^3	270	21	8	8	4.0	4.1×10^2	1.0×10^{-2}
Mechanical vibrations (§5.3.4)	LTV	5.0×10^3	5.0×10^3	12	4	3	100	8.5×10^{-1}	7.2×10^2	4.0×10^{-2}
Wave (§5.3.3)	LTP	2.0×10^4	1.0×10^5	1.5×10^4	12	4	1000	2.8×10^4	1.3×10^3	37

Chapter 6

Conclusions

In this chapter the contributions of this thesis, as well as prospects for future works are summarized.

6.1 Summary

In this thesis, we propose an approach for dimensionality reduction of convection dominated flows, construct the reduced order models (ROMs) and develop a stabilization method with the aim to stabilize the corresponding ROMs.

Most reduced order models are constructed on linear manifolds (subspaces). These manifolds are identified given the snapshots of the solution of high fidelity models or rich experimental data, using approaches such as POD/SVD or DMD. Although these approaches lead to straightforward and fast identifications of the bases, they are inherently incapable of delivering efficient reduced order models. We formally explain the lack of reducibility/compressibility using Kolmogorov n-width and motivate our work using simple problems, e.g. pure convection. A similar irreducibility arises in various systems exhibiting convecting features, traveling wave or shocks, and moving interfaces.

In §2, we propose a low-rank registration-based manifold learning problem with the goal of minimizing the Kolmogorov n-width of the mapped snapshots. The mapping is defined based on a registration problem and low-order interpolation schemes. Further we assume that the identified grid, i.e. a solution of the optimization problem, is of low rank, a characteristics first recognized in [39], and in this thesis we further demonstrate to be sufficient and effective in various systems. We also provide different interpretations of the method, i.e. a change of frame of reference from an Eulerian to an arbitrary Lagrangian Eulerian (ALE) frame, minimizing the Kolmogorov n-width of the snapshots on a learned manifold, and finally an auto-encoder layer based on the physics of the convection dominated problems (low-rank registration-based auto-encoder).

In §3, we derive the ROMs on the learned manifold of §2. In model order reduction of fluid flows, special attention is given to projection-based and more recently to neural network-based ROMs. In the case of projection-based ROMs, the identified bases on the learned manifold are interpolated on the Eulerian grid and the ROMs are constructed on the Eulerian frame. This approach leads to a non-intrusive construction of ROMs capable of being readily incorporated into most codes. We also provide an architecture to incorporate the learned manifold as an auto-encoder layer in a neural network. The proposed auto-encoder reduces the dimensionality of the snapshots and the recurrent neural network (RNN) are trained to reproduce the mapped snapshots.

The projection-based ROMs of §3 are time-varying dynamical systems, lacking a *a priori* stability guarantee. In §4, we develop a *a posteriori* stabilization method for linear time-varying ROMs. The stability of dynamical systems and the eigenvalue reassignment method of stabilizing linear time-invariant ROMs are briefly discussed. The eigenvalue reassignment method is then generalized to stabilize linear time-varying ROMs. The stability criteria and the feedback controller matrices are designed differently to reflect the energy growth of linear time-varying systems.

Finally in §5, different experiments are designed to evaluate and demonstrate efficiency and capabilities of all the proposed methods in §2 to §4.

In §5.1, different parametric and time-dependent problems with traveling waves, moving shocks, and convection/rotating features are designed, which the traditional linear approaches cannot reduce efficiently. These problems include: snapshots of images depicting rotating of a image, reduction of snapshots with different shock location on a parameter-varying nozzle, a wave equation with Gaussian-like solutions traveling in different directions and reflecting from the boundary, one-dimensional Riemann problem with shock and expansion forming and reflecting from the boundaries, two-dimensional Riemann problem depicting traveling shocks and Kelvin–Helmholtz type vortical structures. For all these problems, the proposed dimensionality reduction approach is compared to traditional POD reconstruction, demonstrating at least an order or magnitude reduction in reconstruction error given a set number of bases.

In §5.2, the projection-based and neural network-based ROMs of time-dependent problems introduced in the previous section are constructed. The neural network-based ROMs of two-dimensional problems remain open to further developments. The ROMs error are compared against the traditional ROM where available and a predictive capabilities of the ROMs are investigated. It

is shown the low reconstruction error in the manifold learning stage directly leads to more efficient ROMs.

In §5.3, the proposed stabilization method is evaluated on a wide range of time-invariant, time-periodic and time-varying unstable ROMs. The problems include: a time-invariant model of international space station and its POD Galerkin ROM, a time-periodic model of international space station and its POD Galerkin ROM, the one-dimensional wave equation and it's projection based ROM on the identified time-varying grid representing the optimal manifold identified in §5.1, and a mechanical vibrations problem with time-varying system parameters and its Krylov based time-varying ROM. Finally, the robustness of the stabilized ROMs with respect to the input and parameters space outside of the training range are demonstrated using the uncertainty quantification on the ROM error.

6.2 Future work

Given the potential of the proposed framework, it is worth to develop and explore further applications. In §6.2.1 and §6.2.2 immediate improvements and applications of the methods are discussed. In §6.2.3 to §6.2.5, more speculative and ambitious directions are proposed.

6.2.1 Hyper-reduction for nonlinear time-varying ROMs

In this section, we discuss strategies for developing hyper-reductions algorithms for our proposed nonlinear time-varying MOR approach. For a general nonlinear dynamical system, projection-based MOR does not provide online computational speedup. Hyper-reduction refers to a family of algorithms specifically developed to provide online computational speed-up for general nonlinear system.

To motivate the need for hyper-reduction, consider a first-order dynamical system of the form:

$$\frac{d}{dt}\boldsymbol{w} = \boldsymbol{A}\boldsymbol{w} + \boldsymbol{f}\left(\boldsymbol{w}\right), \tag{6.1}$$

where $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ and $\boldsymbol{f} : \mathbb{R}^N \to \mathbb{R}^N$ are linear and nonlinear operators, respectively. Let the state variable at n^{th} time step be $\boldsymbol{w}^n \approx \boldsymbol{U}\boldsymbol{a}^n$, where $\boldsymbol{U} \in \mathbb{R}^{N\times k}$ and $\boldsymbol{a}^n \in \mathbb{R}^{k\times 1}$ (more details in §3.1.1).

Projecting (6.1) on a test subspace $\boldsymbol{\Phi} \in \mathbb{R}^{N\times k}$ yields the projection-based reduced order model,

$$\boldsymbol{\Phi}^T \boldsymbol{U} \frac{d}{dt}\boldsymbol{a} = \boldsymbol{\Phi}^T \boldsymbol{A}\boldsymbol{U}\boldsymbol{a} + \boldsymbol{\Phi}^T \boldsymbol{f}\left(\boldsymbol{U}\boldsymbol{a}\right). \tag{6.2}$$

Consider a fully discretized form, for example the first order implicit scheme with Galerkin projection,

$$\boldsymbol{a}^n - \boldsymbol{a}^{n-1} = \Delta t \boldsymbol{U}^T \boldsymbol{A}\boldsymbol{U}\boldsymbol{a}^n + \Delta t \boldsymbol{U}^T \boldsymbol{f}\left(\boldsymbol{U}\boldsymbol{a}^n\right). \tag{6.3}$$

In (6.3), the cost of evaluating the linear term scales with the dimensions of the reduced space (k), while the cost for the nonlinear term scales, in the general case, with the size of the high fidelity model (N)[1].

Several methods, often referred to as hyper-reduction, have been developed for reducing the computational cost of projection-based nonlinear ROMs [186–188]. In all of these methods, the main idea involves interpolation of the nonlinear term to reduce the projection cost. For example, Galbally et al. [15] extend the idea of "gappy" POD [189], a method introduced to reconstruct missing/masked data, to only evaluate the nonlinear term at selected indices. Discrete empirical interpolation method (DEIM) avoids the cost of orthogonal projection by specifically selecting interpolation indices to get a near $\mathcal{L}^2$ optimal approximation of the nonlinear term [186]. Carlberg et al. [187] developed the Gauss-Newton with approximated tensors (GNAT) approach to generalize the interpolation procedure.

For the sake of brevity, we restrict out attention to the DEIM hyper-reduction strategy. The goal of DEIM is to approximate $\boldsymbol{f}\left(\boldsymbol{U}\boldsymbol{a}^n\right)$ by evaluating it at only at few indices (entries). Assume that r indices are chosen ($r \ll N$), and the mask matrix is

$$\boldsymbol{P} = [\boldsymbol{e}_{\wp_1}, \cdots, \boldsymbol{e}_{\wp_r}] \in \mathbb{R}^{N\times r}, \tag{6.4}$$

where $\boldsymbol{e}_{\wp_i} \in \mathbb{R}^N$ is the $\wp_i{}^{th}$ column of the identity matrix $\boldsymbol{I}_N \in \mathbb{R}^{N\times N}$.[2] The nonlinear term is approximated on a low-dimensional subspace $\boldsymbol{f}\left(\boldsymbol{U}\boldsymbol{a}^n\right) = \boldsymbol{\Psi}\boldsymbol{c}^n$, where $\boldsymbol{\Psi} \in \mathbb{R}^{N\times q}$ contains the POD

[1] Although some nonlinearities such as polynomials can be precomputed, the costs are intractable in many applications [185].

[2] The algorithm to choose $\boldsymbol{e}_{\wp_i}$ is introduced in [186].

bases.[3] The nonlinear term then can be approximated in this subspace,

$$\boldsymbol{f}\left(\boldsymbol{U}\boldsymbol{a}^n\right) \approx \boldsymbol{\Psi}\left(\boldsymbol{P}^T\boldsymbol{\Psi}\right)^+ \boldsymbol{P}^T\boldsymbol{f}\left(\boldsymbol{U}\boldsymbol{a}^n\right), \tag{6.5}$$

where $^+$ denotes the Moore-Penrose pseudo-inverse, $\boldsymbol{\Psi}\left(\boldsymbol{P}^T\boldsymbol{\Psi}\right)^+ \in \mathbb{R}^{N\times r}$ and $\boldsymbol{P}^T\boldsymbol{f}\left(\boldsymbol{U}\boldsymbol{a}^n\right) \in \mathbb{R}^r$. The evaluation of the masked nonlinear term, $\boldsymbol{P}^T\boldsymbol{f}\left(\boldsymbol{U}\boldsymbol{a}^n\right)$, requires calculations on only selected indices which usually depend on a few neighboring ones, therefore the approximation of $\boldsymbol{f}\left(\boldsymbol{U}\boldsymbol{a}^n\right)$ becomes independent of size of the high fidelity model (N).

Consider the ALE ROM of (6.1), and without loss of generality suppose it is discretized in time using a first-order implicit scheme (more details in §3.2),

$$\boldsymbol{a}^n - \boldsymbol{U}^{n^+}\boldsymbol{U}^{n-1}\boldsymbol{a}^{n-1} = \Delta t\boldsymbol{U}^{n^+}\boldsymbol{A}\boldsymbol{U}^n\boldsymbol{a}^n + \Delta t\boldsymbol{U}^{n^+}\boldsymbol{f}\left(\boldsymbol{U}^n\boldsymbol{a}^n\right), \tag{6.6}$$

where $\mathcal{G}^{-1}(\boldsymbol{U}) = \left\{\boldsymbol{U}^0, \boldsymbol{U}^1, \cdots, \boldsymbol{U}^{N_t}\right\}$, $\boldsymbol{U}^n \in \mathbb{R}^{N\times k}$, and $\boldsymbol{w}^n \approx \boldsymbol{U}^n\boldsymbol{a}^n$ is the approximation of the state variable expressed on the Eulerian grid at the corresponding time step. The time varying coefficient of the linear terms, $\boldsymbol{U}^{n^+}\boldsymbol{U}^{n-1}$ and $\boldsymbol{U}^{n^+}\boldsymbol{A}\boldsymbol{U}^n$, can be precomputed and the evaluation of the linear terms in the ROM scales with k. In contrast, the nonlinear term cannot be precomputed and scales with the dimension of the high fidelity model, N. In the case where a low-order interpolation scheme is used, applying traditional DEIM to the nonlinear term, $\boldsymbol{P}^T\boldsymbol{f}\left(\boldsymbol{U}^n\boldsymbol{a}^n\right)$, is sufficient in reducing the total computation cost. A challenge in this effort will the generalization of this approach to higher-order interpolation schemes which involve a larger number of neighboring cells. In this case, traditional DEIM algorithms may fail to offer speed-up. Consequently, we will also investigate more recent hyper-reduction strategies such as GNAT and Matrix DEIM.

6.2.2 Stabilization of quadratic ROMs of fluid flows

Although the stabilization approach proposed in §4 is, strictly speaking, only applicable to linear, time-varying dynamical systems, there exist a large class of nonlinear systems that may potentially take advantage of some of the techniques developed here. For example, consider a ROM generated via a Galerkin projection of the incompressible Navier-Stokes equations which yields a system of

[3] In DEIM $q = r$.

ordinary differential equations, comprised of constant, linear and quadratic terms, i.e.

$$\frac{d\boldsymbol{a}}{dt} = \boldsymbol{C} + \boldsymbol{L}\boldsymbol{a} + \left[\boldsymbol{a}^T\boldsymbol{Q}^{(1)}\boldsymbol{a}, \boldsymbol{a}^T\boldsymbol{Q}^{(2)}\boldsymbol{a}, \cdots, \boldsymbol{a}^T\boldsymbol{Q}^{(n)}\boldsymbol{a}\right]^T, \tag{6.7}$$

where $\boldsymbol{C} \in \mathbb{R}^k$, $\boldsymbol{L} \in \mathbb{R}^{k\times k}$ and $\boldsymbol{Q}^{(i)} \in \mathbb{R}^{k\times k}$, $\forall i \in \{1, \cdots, k\}$. It is well known that, for a large class of boundary conditions, the nonlinear part of this system is energy preserving: $\boldsymbol{Q}^{(i)}_{j,k} + \boldsymbol{Q}^{(i)}_{k,j} + \boldsymbol{Q}^{(j)}_{i,k} + \boldsymbol{Q}^{(j)}_{k,i} + \boldsymbol{Q}^{(k)}_{i,j} + \boldsymbol{Q}^{(k)}_{j,i} = 0$. Therefore, ROM performance can be improved via appropriate modifications of the linear part of the system; for a more detailed discussion see [36]. It is foreseeable, therefore, that the proposed stabilization approach described in this manuscript can be used to stabilize or calibrate nonlinear ROMs with similar structure.

6.2.3 Low-rank registration-based manifold and Lagrangian coherent structures

The proposed low-rank registration-based manifold provides a framework to further study the nonlinear physical phenomena in convection dominated regimes. As demonstrated in §5.1, the identified low-rank grid *follows* the convective structures present in the flow field. This can be interpreted as a method to capture generalized Lagrangian coherent structures of the flow; for a survey see [190].

Advantages of this realization are twofold. Firstly, the framework provides a numerical approach to realize the path of the traveling structures. This application is of interest especially for geophysical and atmospherics flows [191], where locations and paths of the flow structures, often vortices, are as important as of the statistical properties of the flow. The predictive capabilities the proposed method in a chaotic regime is yet a challenge. Stochastic problems, such as those in [64], are still to be addressed. However, the proposed approach has the potential to provide an efficient yet non-intrusive framework in such regimes. Korteweg–de Vries (KdV) equation, a model of shallow water surfaces, is the stepping stone of such attempts by featuring higher-order nonlinearity, large-scale dispersion, and a non-local dispersions.

Secondly, the snapshots from the moving observer viewpoint are low-rank. Accordingly, from this arbitrary Lagrangian Eulerian (ALE) perspective, the convective features of the flow field are extracted and the remainder of the flow field largely contains the diffusive and shear components

of the flow field and their nonlinear interactions. Many of these low-energy and yet important structures are truncated in the traditional Eulerian approaches of identifying the coherent structures using POD bases. As an example, consider the two-dimensional Riemann problem of §5.1.5 on a fine grid of $N_x = N_y = 1500$ (Fig. 6.1). The Kelvin–Helmholtz type vortical structures are low-energy and therefore are not captured in the first few leading POD bases, from an Eulerian perspective. However, in the proposed framework, these structures remain of significant magnitude. Therefore, the mechanisms of the energy transfer, dynamics and stability of such structures can be investigated in more details.

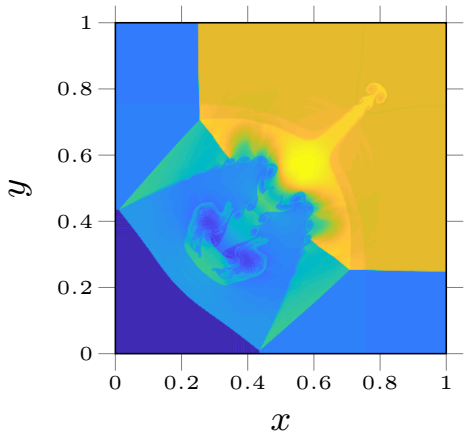


Figure 6.1: Kelvin–Helmholtz type vortical structures in Riemann problem of configuration 3 [2] on a 1500×1500 grid.

6.2.4 Energy decay and stability of convection-dominated ROMs

In §5.2.3, it is shown that the proposed projection-based ROMs can be stable while the traditional projection-based Galerkin ROMs on the Eulerian grid are unstable. This is achieved by breaking the Kolmogorov n-width of the problems,[4] however, the exact mechanism leading to this favorable stability property is not identified.

In traditional ROM construction, two sources of instabilities in convection dominated ROMs have been proposed. In the first, it is argued that the instabilities originate in the truncation of the dissipating range of the energy cascade of the turbulent flows [36]. In the second argument, the instabilities are linked to purely numerical factors [176]. Although both perspectives have provided promising new methods, a consensus on the sources of these instabilities has not been reached.

[4]Not to be confused with Kolmogorov micro-scales and the energy cascades in turbulent flows.

A further comparative study of the energy spectrum of the traditional and the proposed ROMs is suggested to provide deeper insights in regard to the fundamentals of the aforementioned instabilities. Recognition of the source of these instabilities may directly influence the construction and stabilization of fluid reduced order models.

6.2.5 Generalization of the grid bases

In the manifold learning problem proposed in §2, a constant time stepping was assumed. This assumption is consistent with how the high-fidelity model is typically discretized in time. However, clearly, this assumption is not a necessary construct for construction of the manifold. Indeed, significant performance improvements can be expected by generalized the method proposed in this thesis to include a temporal deformation. A schematic of idea is illustrated in Fig. 6.2. Although the addition of this extra degree of freedom complicates the numerical aspects of the temporal evolution of the ROMs, the additional improvement in compressibility of the snapshots maybe worthwhile for certain class of problems.

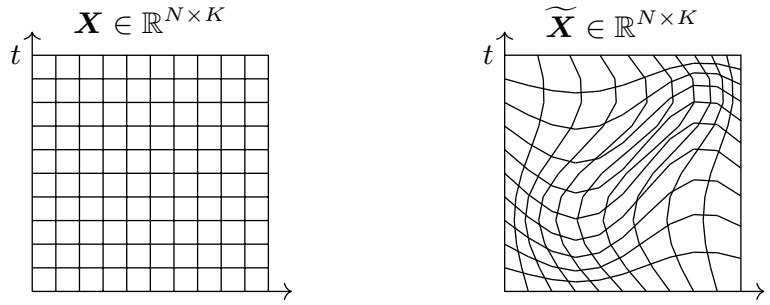


Figure 6.2: Proposed generalization involving additional temporal deformations.

Appendix A

Lagrangian bases MOR

Consider the following scalar, one-dimensional convection-diffusion equation

$$\frac{\partial w(x,t)}{\partial t} + f_1(x,t,w)\frac{\partial w(x,t)}{\partial x} = f_2(x,t,w)\frac{\partial^2 w(x,t)}{\partial x^2}, \tag{A.1}$$

in the domain $(x,t) \in [x_a, x_b] \times [0,T]$, equipped with initial conditions $w(x,0) = w_0(x)$, and appropriate boundary conditions at x_a, and x_b. It is assumed throughout the reminder of this section that (A.1) is discretized uniformly in space $\boldsymbol{x}[n] = [x_1, \ldots, x_N]^T$ using standard techniques such as finite-volume or finite-elements. For the sake of simplicity, and without any loss of generality, time discretization is performed using the first-order implicit Euler scheme. Hence, if $t^0 = 0 < t^1 < \cdots < t^{N_t} = T$ denotes a discretization of the time interval $[0,T]$ and $w(x,t^n) \approx \boldsymbol{w}^n = [w_1^n, \ldots, w_N^n]^T \in \mathbb{R}^N$, for $n \in \{1, \ldots, N_t\}$, the discrete counterpart of (A.1) at time-step n is

$$\boldsymbol{R}(\boldsymbol{w}^n) = \boldsymbol{w}^n - \boldsymbol{w}^{n-1} + \Delta t \boldsymbol{f}_1^n(\boldsymbol{w}^n) \odot (\boldsymbol{D}_x \boldsymbol{w}^n) - \Delta t \boldsymbol{f}_2^n(\boldsymbol{w}^n) \odot (\boldsymbol{D}_{xx} \boldsymbol{w}^n) = \boldsymbol{0}, \tag{A.2}$$

where $\odot$ denotes the Hadamard product, $\boldsymbol{D}_x \in \mathbb{R}^{N \times N}$, and $\boldsymbol{D}_{xx} \in \mathbb{R}^{N \times N}$ are the discrete approximations of the first and second spatial derivatives, respectively.

In traditional projection-based MOR, the solution is approximated by a global trial subspace

$$\boldsymbol{w}^n \approx \widetilde{\boldsymbol{w}}^n = \boldsymbol{w}_0 + \boldsymbol{U}\boldsymbol{a}^n, \tag{A.3}$$

where the columns of $\boldsymbol{U} \in \mathbb{R}^{N \times k}$ contain the basis for this subspace, and $\boldsymbol{a}^n \in \mathbb{R}^k$ denotes the generalized coordinates of the vectors in these basis. Substituting (A.3) into (A.2) and projecting

onto test basis $\boldsymbol{\Phi} \in \mathbb{R}^{N\times k}$, yields the square system

$$\boldsymbol{\Phi}^T \boldsymbol{R}(\boldsymbol{w}_0 + \boldsymbol{U}\boldsymbol{a}^n) = \boldsymbol{0}, \tag{A.4}$$

where $\boldsymbol{\Phi} = \boldsymbol{U}$ in the case of a Galerkin projection.

For the purpose of the proposed dimensionality reduction approach, the governing equations (A.1) are formulated in the Lagrangian frame of reference

$$\frac{dx}{dt} = f_1(x,t,w), \tag{A.5a}$$

$$\frac{\partial w}{\partial t} = f_2(x,t,w)\frac{\partial^2 w}{\partial x^2}. \tag{A.5b}$$

The discrete counterpart of (A.5) at time-step n is

$$\boldsymbol{R}_x(\boldsymbol{x}^n) = \boldsymbol{x}^n - \boldsymbol{x}^{n-1} - \Delta t \boldsymbol{f}_1^n(\boldsymbol{w}^n) = \boldsymbol{0}, \tag{A.6a}$$

$$\boldsymbol{R}_w(\boldsymbol{w}^n) = \boldsymbol{w}^n - \boldsymbol{w}^{n-1} - \Delta t \boldsymbol{f}_2^n(\boldsymbol{w}^n) \odot (\boldsymbol{D}_{xx}^n \boldsymbol{w}^n) = \boldsymbol{0}, \tag{A.6b}$$

where $\boldsymbol{x}[n] = [x_1^n, \ldots, x_N^n]^T$ denotes the locations of the Lagrangian computational grid at n^{th} time level, and $\boldsymbol{D}_{xx}^n$ denotes the discrete approximation of the second derivative on the Lagrangian grid at time level n.

A.1 Construction of Lagrangian ROM

In the proposed new dimensionality reduction approach, the Lagrangian solution is approximated by a global trial subspace

$$\boldsymbol{x}^n \approx \widetilde{\boldsymbol{x}}^n = \boldsymbol{x}_0 + \boldsymbol{U}_x \boldsymbol{a}_x^n, \tag{A.7a}$$

$$\boldsymbol{w}^n \approx \widetilde{\boldsymbol{w}}^n = \boldsymbol{w}_0 + \boldsymbol{U}_w \boldsymbol{a}_w^n, \tag{A.7b}$$

where the columns of $\boldsymbol{U}_x \in \mathbb{R}^{N\times k}$ and $\boldsymbol{U}_w \in \mathbb{R}^{N\times k}$ contain the basis for the corresponding subspace, and $\boldsymbol{a}_x^n \in \mathbb{R}^k$ and $\boldsymbol{a}_w^n \in \mathbb{R}^k$ denote the generalized coordinates of the vectors in these basis. Substituting (A.7) into (A.6) and projecting onto test basis $\boldsymbol{\Phi}_x \in \mathbb{R}^{N\times k}$ and $\boldsymbol{\Phi}_w \in \mathbb{R}^{N\times k}$, yields

the square system

$$\boldsymbol{\Phi}_x^T \boldsymbol{R}_x(\boldsymbol{x}_0 + \boldsymbol{U}_x \boldsymbol{a}_x^n) = \boldsymbol{0}, \tag{A.8a}$$

$$\boldsymbol{\Phi}_w^T \boldsymbol{R}_w(\boldsymbol{w}_0 + \boldsymbol{U}_w \boldsymbol{a}_w^n) = \boldsymbol{0}, \tag{A.8b}$$

where $\boldsymbol{\Phi}_x = \boldsymbol{U}_x$ and $\boldsymbol{\Phi}_w = \boldsymbol{U}_w$ in the case of a Galerkin projection.

A.2 Construction of Lagrangian global bases

For cases where the HFM is formulated in the Lagrangian frame of reference, that is, when the governing equations are in the form of (A.6), construction of Lagrangian basis follows a procedure very similar to traditional POD. Specifically, we solve the low-rank approximation problem given by (2.5), for a snapshot matrix $\boldsymbol{X} \in \mathbb{R}^{2N \times K}$ containing solution snapshots computed by (A.6). In other words, $[\boldsymbol{X}]_{:,i} = \left[\boldsymbol{x}^i, \boldsymbol{w}^i\right]^T$ for $i = 1, \ldots, K$. Therefore, the optimal Lagrangian basis corresponds to $\boldsymbol{U}_x = [\boldsymbol{U}]_{1:N,1:k}$ and $\boldsymbol{U}_w = [\boldsymbol{U}]_{N+1:2N,1:k}$, where $\boldsymbol{U}$ are the left singular vectors of the snapshot matrix $\boldsymbol{X}$.

For cases where the HFM is formulated in the Eulerian frame of reference, that is, when the governing equations are in the form of (A.1), Lagrangian basis cannot be constructed by solving the standard low-rank approximation problem because Eulerian HFMs typically do not provide the grid deformation $\boldsymbol{x}^i$. Thus, it is not possible to form the snapshot matrix $[\boldsymbol{X}]_{:,i} = \left[\boldsymbol{x}^i, \boldsymbol{w}^i\right]^T$. The present thesis addresses such problems.

A.2.1 Lagrangian grid entanglement

In the proposed Lagrangian MOR approach, the evolution of the Lagrangian spatial grid is approximated in a low-dimensional subspace, $\boldsymbol{x}^i \approx \boldsymbol{U}_x \boldsymbol{a}_x^i$. Unfortunately, this low-dimensional approximation is not guaranteed to preserve the topological properties of the original HFM simulation. Indeed, for some particular cases, the low-dimensional Lagrangian grid becomes severely distorted leading to numerical instabilities. For these cases, particularly those featuring strong shocks, we propose the following modification to the model reduction procedure. Instead of solving the diffusion step in the Lagrangian frame, as in (A.8b), the state basis $\boldsymbol{U}_w$ are interpolated from

the Lagrangian to the stationary Eulerian grid at every time level n and the projection is performed in the Eulerian frame. Instead of solving the diffusion step in the Lagrangian frame, as in (A.8b), the state basis $\boldsymbol{U}_w$ are interpolated from the Lagrangian to the stationary Eulerian grid at every time level n and the projection is performed in the Eulerian frame. Therefore, (A.8b), is replaced with the following

$$\widehat{\boldsymbol{\Phi}}_w^T \widehat{\boldsymbol{R}}_w(\widehat{\boldsymbol{w}}_0 + \widehat{\boldsymbol{U}}_w \boldsymbol{a}_w^n) = \boldsymbol{0}, \tag{A.9}$$

where $\widehat{\boldsymbol{\Phi}}_w = \mathcal{P}^{\boldsymbol{x}}_{\boldsymbol{\Phi}_x \boldsymbol{a}_x^n}(\boldsymbol{U}_w)$, $\widehat{\boldsymbol{U}}_w = \mathcal{P}^{\boldsymbol{x}}_{\boldsymbol{U}_x \boldsymbol{a}_x^n}(\boldsymbol{U}_w)$, and $\widehat{\boldsymbol{w}}_0 = \mathcal{P}^{\boldsymbol{x}}_{\boldsymbol{U}_x \boldsymbol{a}_x^n}(\boldsymbol{w}_0)$, are the interpolated basis and initial conditions and $\widehat{\boldsymbol{R}}_w$ is the diffusion step in the Eulerian frame, defined as

$$\widehat{\boldsymbol{R}}(\boldsymbol{w}^n) = \boldsymbol{w}^n - \boldsymbol{w}^{n-1} - \Delta t \boldsymbol{f}_2^n(\boldsymbol{w}^n) \odot (\boldsymbol{D}_{xx}\boldsymbol{w}^n) = \boldsymbol{0}. \tag{A.10}$$

The interpolation and the ROM of (A.10) is generalized in §3 of the present thesis.

A.3 Numerical experiments

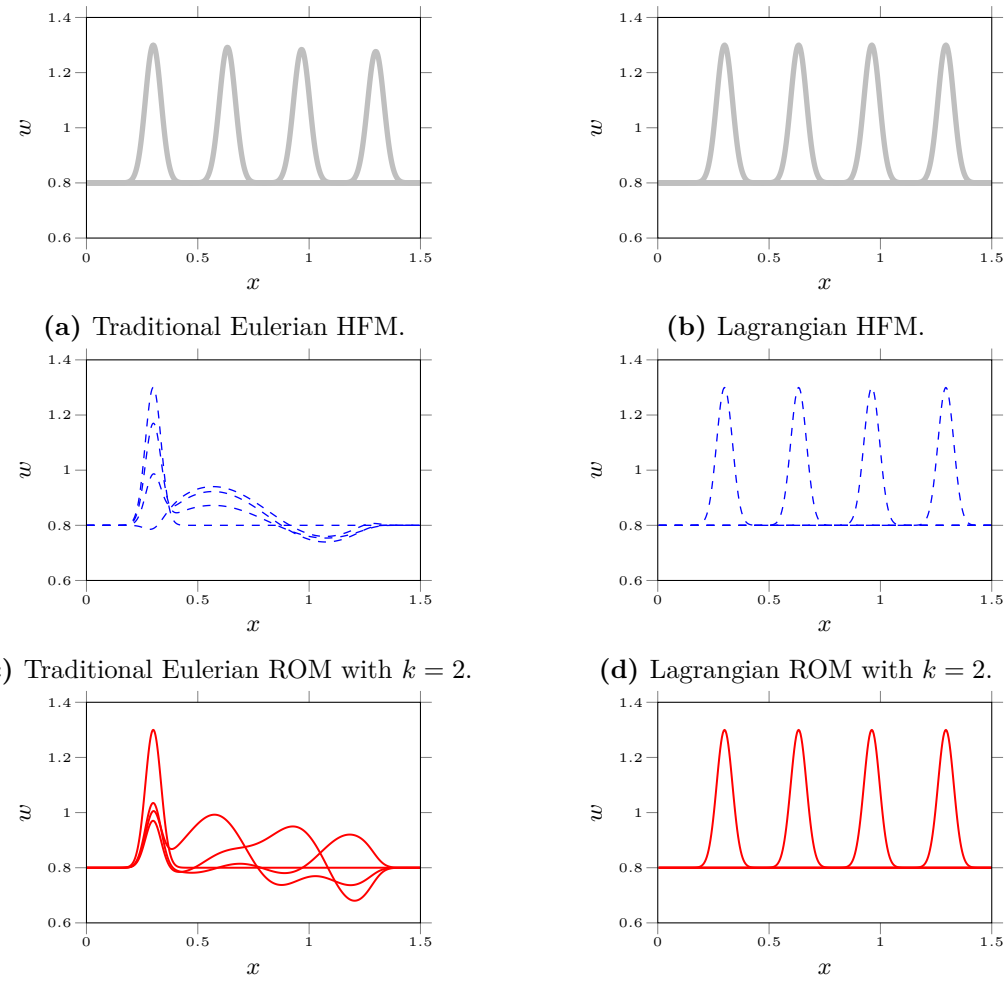
A.3.1 Convection-diffusion equation

The proposed approach is first applied to the reduction of the scalar linear convection equation and a high Péclet number convection-diffusion equation. Specifically, we consider (A.1) with $f_1(x,t,w) = 1$, $f_2(x,t,w) = 1/Pe$, $w(x,0) = 0.5\ e^{-(x-0.3)^2/0.05^2}$, $w(0,t) = 0$, for $(x,t) \in [0,1.5] \times [0,1]$, where $Pe = \infty$ (pure convection) and $Pe = 10^3$.

Two HFMs are constructed for this case; one in the Eulerian frame, as in (A.2), and one in the fully Lagrangian frame, as in (A.6). For both models, a second-order central finite difference discretization is used. $N = 2000$ grid points are used to discretized the domain $0 \leq x \leq 1.5$. A total of $K = 2000$ Eulerian and Lagrangian snapshots are collected. Eulerian and Lagrangian basis are constructed by solving (2.5). Eulerian ROMs are solved in the form of (A.4) and Lagrangian ROMs are solved in the fully Lagrangian frame, as in (A.8a) and (A.8b). Galerkin projection is used in all cases so $\boldsymbol{\Phi} = \boldsymbol{U}$ and $\boldsymbol{\Phi}_x = \boldsymbol{U}_x$, $\boldsymbol{\Phi}_w = \boldsymbol{U}_w$.

ROM solutions for the convection equation and the high Péclet number convection-diffusion equation are illustrated in Fig. A.1 and Fig. A.2, respectively.

Convergence of Eulerian and Lagrangian ROMs of the high Péclet number convection-diffusion



(a) Traditional Eulerian HFM.

(b) Lagrangian HFM.

(c) Traditional Eulerian ROM with $k = 2$.

(d) Lagrangian ROM with $k = 2$.

(e) Traditional Eulerian ROM with $k = 5$.

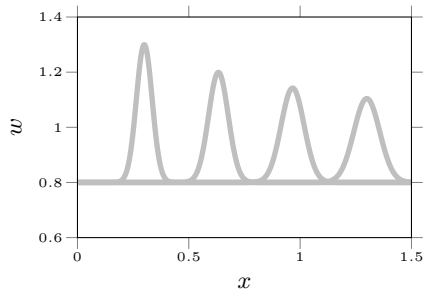
(f) Lagrangian ROM with $k = 5$.

Figure A.1: Model order reduction of scalar convection equation; Solutions are plotted for $t = \{0, 1/3, 2/3, 1\}$.

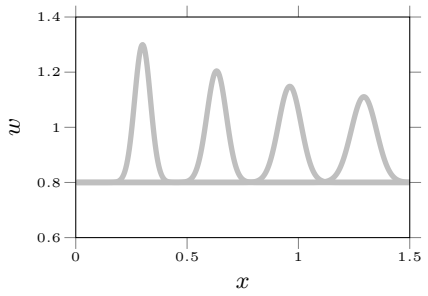
are illustrated in Fig. A.4a, where error is defined as Frobenius distance between HFM and its ROM. For both cases considered, Lagrangian ROMs significantly outperform the Eulerian ROMs in all cases considered.

A.3.2 Burgers' equation

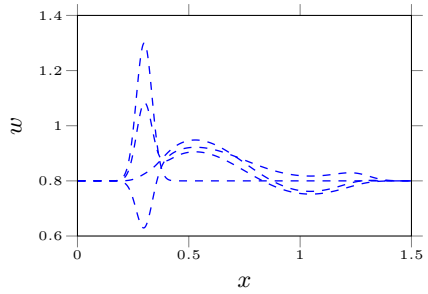
The proposed approach is next applied to the reduction of a convection-dominated Burgers' equation. Specifically, we consider (A.1) with $f_1(x,t,w) = w(x,t)$, $f_2(x,t,w) = \nu$, $w(x,0) = 0.8 + 0.5\ e^{-(x-0.3)^2/0.1^2}$, $w(0,t) = 0$, for $(x,t) \in [0,1.5] \times [0,1]$, where $\nu = 10^{-3}$. As before, two HFM are constructed, one in the Eulerian frame, as in (A.2), and one in the fully Lagrangian frame, as in (A.6). For both models, a second-order central finite difference discretization is used. $N = 2000$ grid



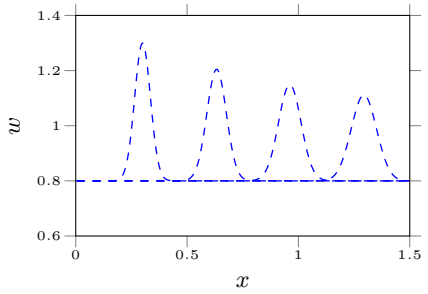
(a) Traditional Eulerian HFM.



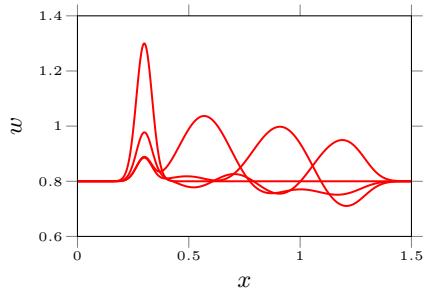
(b) Lagrangian HFM.



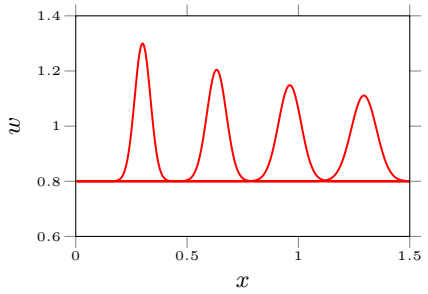
(c) Traditional Eulerian ROM with $k = 2$.



(d) Lagrangian ROM with $k = 2$.



(e) Traditional Eulerian ROM with $k = 5$.



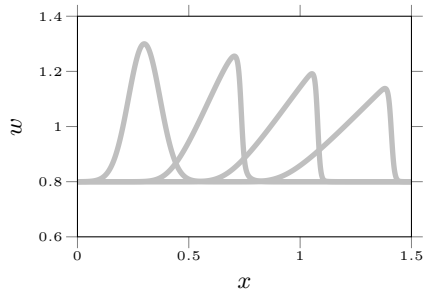
(f) Lagrangian ROM with $k = 5$.

Figure A.2: Model order reduction of scalar convection-diffusion equation with $Pe = 10^3$; Solutions are plotted for $t = \{0, 1/3, 2/3, 1\}$.

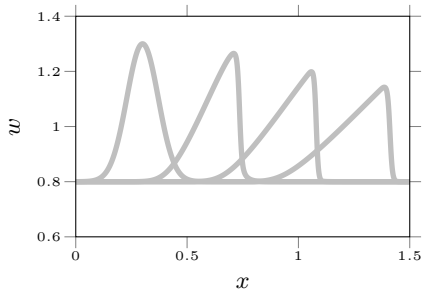
points are used to discretized the domain $0 \leq x \leq 1.5$. Total of $K = 2000$ Eulerian and Lagrangian snapshots are collected. Eulerian and Lagrangian basis are constructed by solving (2.5). Eulerian ROMs are solved in the form of (A.4). Due to the significant Lagrangian grid entanglement caused by the nonlinear convection term in the Burgers' equation, the Lagrangian ROMs are solved using the modified diffusion step; i.e. (A.8b) is replaced with (A.9). Galerkin projection is used in both cases.

Solutions at $t = \{0, 1/3, 2/3, 1\}$ derived using the traditional and the new proposed approach are illustrated in Fig. A.3.

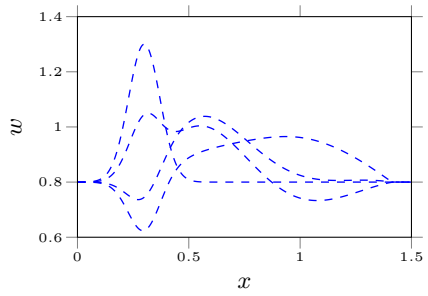
Convergence of the Eulerian and Lagrangian ROMs are illustrated in Fig. A.4b. The Lagrangian ROMs significantly outperform the Eulerian ROMs. For example, a $k = 1$ Lagrangian ROM has approximately the same error as a $k = 20$ Eulerian ROM. Note that Lagrangian ROMs only up to $k = 5$ are considered. After $k = 5$, some of the interpolated Lagrangian basis $\widehat{\boldsymbol{U}}_w$ become linearly dependent and thus, no further performance gain can be expected.



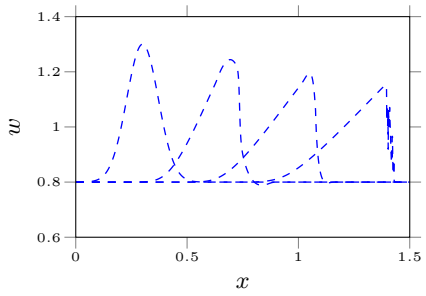
(a) Traditional Eulerian HFM.



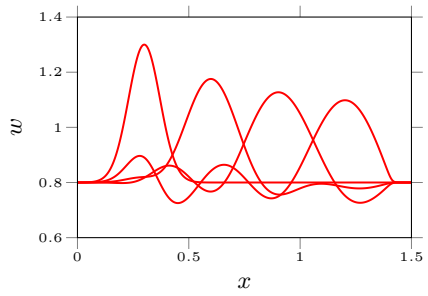
(b) Lagrangian HFM.



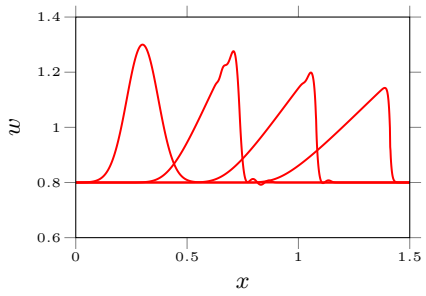
(c) Traditional Eulerian ROM with $k = 2$.



(d) Lagrangian ROM with $k = 2$.

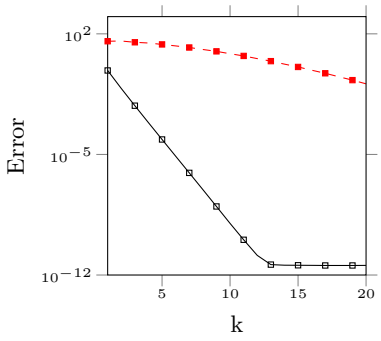


(e) Traditional Eulerian ROM with $k = 5$.

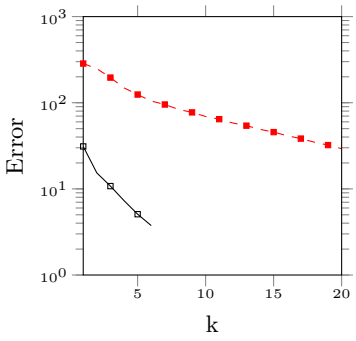


(f) Lagrangian ROM with $k = 5$.

Figure A.3: Model order reduction of scalar Burgers' equation with $\nu = 10^{-3}$; Solutions are plotted for $t = \{0, 1/3, 2/3, 1\}$.



(a) $Pe = 10^3$



(b) $Re = 10^3$

Figure A.4: ROM convergence for scalar convection-diffusion equation and Burgers' equation. Traditional Eulerian ROMs (dashed red lines with filled markers) and Lagrangian ROMs (solid black lines with empty markers).

References

[1] B. Lohmann, R. Eid, Efficient order reduction of parametric and nonlinear models by superposition of locallly reduced models, Methoden und Anwendungen der Regelung- stechnik, Achen, 2009.

[2] P. D. Lax, X.-d. Liu, Solution of two-dimensional Riemann problems of gas dynamics by positive schemes, SIAM Journal on Scientific Computing 19 (1998) 319–340.

[3] F. Chinesta, R. Keunings, A. Leygue, The Proper Generalized Decomposition for Advanced Numerical Simulations, Springer Briefs in Applied Sciences and Technology, Springer International Publishing, 2014.

[4] P. Benner, S. Gugercin, K. Willcox, A survey of projection-based model reduction methods for parametric dynamical systems, SIAM Review 57 (2015) 1–49.

[5] G. Rozza, H. Malik, N. Demo, M. Tezzele, M. Girfoglio, G. Stabile, A. Mola, Advances in reduced order methods for parametric industrial problems in computational fluid dynamics, 6th European Conference on Computational Mechanics (ECCM 6), Glasgow, UK, 2018.

[6] P. LeGresley, J. Alonso, Airfoil design optimization using reduced order models based on proper orthogonal decomposition, Fluids 2000 Conference and Exhibit, American Institute of Aeronautics and Astronautics, Denver, CO, 2000.

[7] T. Lieu, C. Farhat, Adaptation of aeroelastic reduced-order models and application to an F-16 configuration, AIAA Journal 45 (2007) 1244–1257.

[8] D. Amsallem, C. Farhat, Interpolation method for adapting reduced-order models and application to aeroelasticity, AIAA Journal 46 (2008) 1803–1813.

[9] A. J. Sinclair, G. T. Flowers, Coupled flight-dynamic and low-order aeroelastic model for a slender launch vehicle, Journal of Vibration and Acoustics 135 (2013) 021002.

[10] M. Fahl, E. W. Sachs, Reduced order modelling approaches to PDE-constrained optimization based on proper orthogonal decomposition, in: L. T. Biegler, M. Heinkenschloss, O. Ghattas, B. van Bloemen Waanders (Eds.), Large-Scale PDE-Constrained Optimization, volume 30 of *Lecture Notes in Computational Science and Engineering*, Springer, Berlin, Heidelberg, 2003, pp. 268–280.

[11] D. Amsallem, M. J. Zahr, Y. Choi, C. Farhat, Design optimization using hyper-reduced order models, Structural and Multidisciplinary Optimization 51 (2015) 919–940.

[12] L. Mathelin, M. Y. Hussaini, T. A. Zang, Stochastic approaches to uncertainty quantification in CFD simulations, Numerical Algorithms 38 (2005) 209–236.

[13] O. Knio, O. Le Maître, Uncertainty propagation in CFD using polynomial chaos decomposition, Fluid Dynamics Research 38 (2006) 616–640.

[14] T. Bui-Thanh, K. Willcox, O. Ghattas, Parametric reduced-order models for probabilistic analysis of unsteady aerodynamic applications, AIAA Journal 46 (2008) 2520–2529.

[15] D. Galbally, K. Fidkowski, K. Willcox, O. Ghattas, Non-linear model reduction for uncertainty quantification in large-scale inverse problems, International Journal for Numerical Methods in Engineering 81 (2009) 1581–1608.

[16] T. Lieu, C. Farhat, M. Lesoinne, Reduced-order fluid/structure modeling of a complete aircraft configuration, Computer Methods in Applied Mechanics and Engineering 195 (2006) 5730–5742.

[17] M. Ghoreyshi, A. Jirásek, R. M. Cummings, Reduced order unsteady aerodynamic modeling for stability and control analysis using computational fluid dynamics, Progress in Aerospace Sciences 71 (2014) 167–217.

[18] A. C. Antoulas, D. C. Sorensen, Approximation of large-scale dynamical systems: An overview, International Journal of Applied Mathematics and Computer Science 11 (2001) 1093–1121.

[19] W. Dahmen, C. Plesken, G. Welper, Double greedy algorithms: Reduced basis methods for transport dominated problems, ESAIM: Mathematical Modelling and Numerical Analysis 48 (2014) 623–663.

[20] A. Iollo, D. Lombardi, Advection modes by optimal mass transfer, Physical Review E 89 (2014) 022923.

[21] J. Sesterhenn, A. Shahirpour, A Lagrangian dynamic mode decomposition, arXiv preprint arXiv:1603.02539 (2016). `arXiv:1603.02539`.

[22] D. Rim, S. Moe, R. J. LeVeque, Transport reversal for model reduction of hyperbolic partial differential equations, arXiv preprint arXiv:1701.07529 (2017). `arXiv:1701.07529`.

[23] J. Reiss, P. Schulze, J. Sesterhenn, V. Mehrmann, The shifted proper orthogonal decomposition: A mode decomposition for multiple transport phenomena, SIAM Journal on Scientific Computing 40 (2018) A1322–A1344.

[24] K. Lee, K. T. Carlberg, Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders, arXiv preprint arXiv:1812.08373v1 (2018). `arXiv:1812.08373v1`.

[25] N. Cagniart, Y. Maday, B. Stamm, Model order reduction for problems with large convection effects, in: B. N. Chetverushkin, W. Fitzgibbon, Y. Kuznetsov, P. Neittaanmäki, J. Periaux, O. Pironneau (Eds.), Contributions to Partial Differential Equations and Applications, Computational Methods in Applied Sciences, Springer International Publishing, 2019, pp. 131–150.

[26] A. Mendible, A. Aravkin, W. Lowrie, S. Brunton, J. Nathan Kutz, Dimensionality reduction and reduced order modeling for traveling wave physics, arXiv preprint arXiv:1911.00565v1 (2019). `arXiv:1911.00565v1`.

[27] K. Kunisch, S. Volkwein, Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics, SIAM Journal on Numerical Analysis 40 (2002) 492–515.

[28] K. Veroy, C. Prud'homme, D. Rovas, A. Patera, A posteriori error bounds for reduced-basis approximation of parametrized noncoercive and nonlinear elliptic partial differential equations, in: 16th AIAA Computational Fluid Dynamics Conference, Orlando, Florida, 2003.

[29] M. Kahlbacher, S. Volkwein, Galerkin proper orthogonal decomposition methods for parameter dependent elliptic systems, Discussiones Mathematicae. Differential Inclusions, Control and Optimization 27 (2007) 95–117.

[30] M. Balajewicz, J. Toivanen, Reduced order models for pricing american options under stochastic volatility and jump-diffusion models, Procedia Computer Science 80 (2016) 734–743.

[31] J. Ge, K. I. Kim, Z. Lin, H. Zhu, A SIS reaction–diffusion–advection model in a low-risk and high-risk domain, Journal of Differential Equations 259 (2015) 5486–5509.

[32] L.-P. Morency, A. Rahimi, N. Checka, T. Darrell, Fast stereo-based head tracking for interactive environments, in: Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition, Washington, DC, 2002, pp. 390–395.

[33] M. Balajewicz, E. Dowell, Reduced-order modeling of flutter and limit-cycle oscillations using the sparse Volterra series, Journal of Aircraft 49 (2012) 1803–1812.

[34] J. Du, F. Fang, C. C. Pain, I. Navon, J. Zhu, D. A. Ham, POD reduced-order unstructured mesh modeling applied to 2D and 3D fluid flow, Computers & Mathematics with Applications 65 (2013) 362–379.

[35] M. Balajewicz, E. Dowell, Stabilization of projection-based reduced-order models of the Navier-Stokes equations, Nonlinear Dynamics 70 (2012) 1619–1632.

[36] M. Balajewicz, E. Dowell, B. Noack, Low-dimensional modelling of high-Reynolds-number shear flows incorporating constraints from the Navier-Stokes equation, Journal of Fluid Mechanics 729 (2013) 285–308.

[37] M. Balajewicz, F. Charbel, Reduction of nonlinear embedded boundary models for problems with evolving interfaces, Journal of Computational Physics 274 (2014) 489–504.

[38] D. Xiao, F. Fang, A. G. Buchan, C. C. Pain, I. M. Navon, J. Du, G. Hu, Non-linear model reduction for the Navier-Stokes equations using residual DEIM method, Journal of Computational Physics 263 (2014) 1–18.

[39] R. Mojgani, M. Balajewicz, Lagrangian basis method for dimensionality reduction of convection dominated nonlinear flows, arXiv preprint arXiv:1701.04343 (2017). arXiv:1701.04343.

[40] N. J. Nair, M. Balajewicz, Transported snapshot model order reduction approach for parametric, steady-state fluid flows containing parameter-dependent shocks, International Journal for Numerical Methods in Engineering 117 (2019) 1234–1262.

[41] L. Sirovich, Turbulence and the dynamics of coherent structures. part I: Coherent structures, Quarterly of Applied Mathematics 45 (1987) 561–571.

[42] J. L. Lumley, Stochastic tools in turbulence, Academic Press, 1970.

[43] D. J. Lucia, P. S. Beran, W. A. Silva, Aeroelastic system development using proper orthogonal decomposition and Volterra theory, Journal of Aircraft 42 (2005) 509–518.

[44] L. Perret, E. Collin, J. Delville, Polynomial identification of POD based low-order dynamical system, Journal of Turbulence 7 (2006) 1–15.

[45] M.-L. Rapun, J. M. Vega, Reduced-order models based on local POD plus Galerkin projection, Journal of Computational Physics 229 (2010) 3046–3063.

[46] B. R. Noack, M. Morzynski, G. Tadmor, Reduced-order modelling for flow control, volume 528 of *CISM International Centre for Mechanical Sciences*, Springer Science & Business Media, 2011.

[47] P. Holmes, J. L. Lumley, G. Berkooz, C. W. Rowley, Turbulence, coherent structures, dynamical systems and symmetry, 2nd ed., Cambridge University Press, 2012.

[48] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, D. S. Henningson, Spectral analysis of nonlinear flows, Journal of Fluid Mechanics 641 (2009) 115–127.

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

[50] K. K. Chen, J. H. Tu, C. W. Rowley, Variants of dynamic mode decomposition: Boundary condition, Koopman, and Fourier analyses, Journal of Nonlinear Science 22 (2012) 887–915.

[51] M. O. Williams, I. G. Kevrekidis, C. W. Rowley, A data-driven approximation of the Koopman operator: Extending dynamic mode decomposition, Journal of Nonlinear Science 25 (2015) 1307–1346.

[52] A. Wynn, D. Pearson, B. Ganapathisubramani, P. J. Goulart, Optimal mode decomposition for unsteady flows, Journal of Fluid Mechanics 733 (2013) 473.

[53] J. N. Kutz, X. Fu, S. L. Brunton, Multiresolution dynamic mode decomposition, SIAM Journal on Applied Dynamical Systems 15 (2016) 713–735.

[54] J. L. Proctor, S. L. Brunton, J. N. Kutz, Dynamic mode decomposition with control, SIAM Journal on Applied Dynamical Systems 15 (2016) 142–161.

[55] S. L. Brunton, J. L. Proctor, J. H. Tu, J. N. Kutz, Compressed sensing and dynamic mode decomposition, Journal of Computational Dynamics 2 (2015) 165–191.

[56] K. Willcox, J. Peraire, Balanced model reduction via the proper orthogonal decomposition, AIAA journal 40 (2002) 2323–2330.

[57] C. W. Rowley, Model reduction for fluids, using balanced proper orthogonal decomposition, International Journal of Bifurcation and Chaos 15 (2005) 997–1013.

[58] B. C. Moore, Principal component analysis in linear systems: Controllability, observability, and model reduction, IEEE Transactions on Automatic Control 26 (1981) 17–32.

[59] L. Machiels, Y. Maday, I. B. Oliveira, A. T. Patera, D. V. Rovas, Output bounds for reduced-basis approximations of symmetric positive definite eigenvalue problems, Comptes Rendus de l'Académie des Sciences - Series I - Mathematics 331 (2000) 153–158.

[60] C. W. Rowley, J. E. Marsden, Reconstruction equations and the Karhunen–Loève expansion for systems with symmetry, Physica D: Nonlinear Phenomena 142 (2000) 1–19.

[61] C. W. Rowley, I. G. Kevrekidis, J. E. Marsden, K. Lust, Reduction and reconstruction for self-similar dynamical systems, Nonlinearity 16 (2003) 1257–1275.

[62] M. Kavousanakis, R. Erban, A. G. Boudouvis, C. W. Gear, I. G. Kevrekidis, Projective and coarse projective integration for problems with continuous symmetries, Journal of Computational Physics 225 (2007) 382–407.

[63] J.-F. Gerbeau, D. Lombardi, Approximated Lax pairs for the reduced order integration of nonlinear evolution equations, Journal of Computational Physics 265 (2014) 246–269.

[64] S. Mowlavi, T. P. Sapsis, Model order reduction for stochastic dynamical systems with continuous symmetries, SIAM Journal on Scientific Computing 40 (2018) A1669–A1695.

[65] D. Rim, K. T. Mandli, Model reduction of a parametrized scalar hyperbolic conservation law using displacement interpolation, arXiv preprint arXiv:1805.05938v1 (2018). arXiv:1805.05938v1.

[66] M. Kirby, D. Armbruster, Reconstructing phase space from PDE simulations, ZAMP Zeitschrift für angewandte Mathematik und Physik 43 (1992) 999–1022.

[67] S. Glavaski, J. E. Marsden, R. M. Murray, Model reduction, centering, and the Karhunen-Loeve expansion, in: Proceedings of the 37th IEEE Conference on Decision and Control, volume 2, Tampa, FL, 1998, pp. 2071–2076.

[68] S. Helgason, The radon transform on R^n, in: Integral Geometry and Radon Transforms, Springer New York, New York, NY, 2011.

[69] J. Reiss, P. Schulze, J. Sesterhenn, The shifted proper orthogonal decomposition: A mode decomposition for multiple transport phenomena, arXiv preprint arXiv:1512.01985 (2015). arXiv:1512.01985.

[70] P. Schulze, J. Reiss, V. Mehrmann, Model reduction for a pulsed detonation combuster via shifted proper orthogonal decomposition, arXiv preprint arXiv:1803.01805 (2018). arXiv:1803.01805.

[71] N. Sarna, S. Grundel, Model reduction of time-dependent hyperbolic equations using collocated residual minimisation and shifted snapshots, arXiv preprint arXiv:2003.06362 (2020). arXiv:2003.06362.

[72] D. Torlo, Model reduction for advection dominated hyperbolic problems in an ALE framework: Offline and online phases, arXiv preprint arXiv:2003.13735 (2020). arXiv:2003.13735.

[73] T. Taddei, L. Zhang, Space-time registration-based model reduction of parameterized one-dimensional hyperbolic pdes, arXiv preprint arXiv:2004.06693 (2020). arXiv:2004.06693.

[74] D. Amsallem, M. J. Zahr, C. Farhat, Nonlinear model order reduction based on local reduced-order bases, International Journal for Numerical Methods in Engineering 92 (2012) 891–916.

[75] D. J. Lucia, Reduced order modeling for high speed flows with moving shocks, Ph.D. thesis, Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio, 2001.

[76] D. Rim, B. Peherstorfer, K. T. Mandli, Manifold approximations via transported subspaces: Model reduction for transport-dominated problems, arXiv preprint arXiv:1912.13024 (2019). arXiv:1912.13024.

[77] K. Carlberg, Adaptive h-refinement for reduced-order models, International Journal for Numerical Methods in Engineering 102 (2015) 1192–1210.

[78] H. Babaee, M. Farazmand, G. Haller, T. P. Sapsis, Reduced-order description of transient instabilities and computation of finite-time Lyapunov exponents, Chaos: An Interdisciplinary Journal of Nonlinear Science 27 (2017) 063103.

[79] F. J. Gonzalez, M. Balajewicz, Deep convolutional recurrent autoencoders for learning low-dimensional feature dynamics of fluid systems, arXiv preprint arXiv:1808.01346 (2018). arXiv:1808.01346.

[80] E. J. Parish, K. T. Carlberg, Time-series machine-learning error models for approximate solutions to parameterized dynamical systems, Computer Methods in Applied Mechanics and Engineering 365 (2020) 112990.

[81] I. Kalashnikova, B. van Bloemen Waanders, S. Arunajatesan, M. Barone, Stabilization of projection-based reduced order models for linear time-invariant systems via optimization-based eigenvalue reassignment, Computer Methods in Applied Mechanics and Engineering 272 (2014) 251–270.

[82] G. E. Hinton, M. Revow, P. Dayan, Recognizing handwritten digits using mixtures of linear models, in: G. Tesauro, D. S. Touretzky, T. K. Leen (Eds.), Advances in Neural Information Processing Systems 7, MIT Press, 1995, pp. 1015–1022.

[83] M. A. A. Cox, T. F. Cox, Multidimensional scaling, in: Handbook of Data Visualization, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 315–347.

[84] J. Friedman, T. Hastie, R. Tibshirani, The elements of statistical learning, number 10 in 1, Springer series in statistics New York, 2001.

[85] J. B. Tenenbaum, Mapping a manifold of perceptual observations, in: M. I. Jordan, M. J. Kearns, S. A. Solla (Eds.), Advances in Neural Information Processing Systems 10, MIT Press, 1998, pp. 682–688.

[86] S. Mika, B. Schölkopf, A. J. Smola, K.-R. Müller, M. Scholz, G. Rätsch, Kernel pca and de-noising in feature spaces, in: M. J. Kearns, S. A. Solla, D. A. Cohn (Eds.), Advances in Neural Information Processing Systems 11, MIT Press, 1999, pp. 536–542.

[87] S. T. Roweis, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (2000) 2323–2326.

[88] M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, Neural Computation 15 (2003) 1373–1396.

[89] K. Q. Weinberger, F. Sha, L. K. Saul, Learning a kernel matrix for nonlinear dimensionality reduction, in: Twenty-first international conference on Machine learning - ICML '04, ACM Press, New York, New York, USA, 2004, p. 106.

[90] H. G.E, S. R.R, Reducing the dimensionality of data with neural networks, Science 313 (2006) 504–507.

[91] L. v. d. Maaten, G. Hinton, Visualizing data using t-SNE, Journal of machine learning research 9 (2008) 2579–2605.

[92] C. Walder, B. Schölkopf, Diffeomorphic dimensionality reduction, in: D. Koller, D. Schuurmans, Y. Bengio, L. Bottou (Eds.), Advances in Neural Information Processing Systems 21, Curran Associates, Inc., 2009, pp. 1713–1720.

[93] K. Lee, K. T. Carlberg, Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders, Journal of Computational Physics 404 (2020) 108973.

[94] J. Masci, U. Meier, D. Cireşan, J. Schmidhuber, Stacked convolutional auto-encoders for hierarchical feature extraction, in: International conference on artificial neural networks, Springer, 2011, pp. 52–59.

[95] A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, CoRR abs/1511.06434 (2015).

[96] M. Cheng, F. Fang, C. Pain, I. Navon, Data-driven modelling of nonlinear spatio-temporal fluid flows using a deep convolutional generative adversarial network, Computer Methods in Applied Mechanics and Engineering 365 (2020) 113000.

[97] J. N. Kutz, S. L. Brunton, B. W. Brunton, J. L. Proctor, Dynamic mode decomposition: data-driven modeling of complex systems, SIAM, 2016.

[98] C. Gin, B. Lusch, S. L. Brunton, J. N. Kutz, Deep learning models for global coordinate transformations that linearize pdes, arXiv preprint arXiv:1911.02710 (2019). `arXiv:1911.02710`.

[99] T. Taddei, A registration method for model order reduction: Data compression and geometry reduction, SIAM Journal on Scientific Computing 42 (2020) A997–A1027.

[100] A. Quarteroni, A. Manzoni, F. Negri, Reduced Basis Methods for Partial Differential Equations, volume 92 of *UNITEXT*, Springer International Publishing, 2016.

[101] T. Taddei, A registration method for model order reduction: data compression and geometry reduction, arXiv preprint arXiv:1906.11008 (2019). `arXiv:1906.11008`.

[102] M. Bachmayr, A. Cohen, Kolmogorov widths and low-rank approximations of parametric elliptic PDEs, Mathematics of Computation 86 (2016) 701–724.

[103] A. Pinkus, n-Widths in Approximation Theory, Springer, Berlin, Heidelberg, 1985.

[104] J. M. Melenk, On n-widths for elliptic problems, Journal of Mathematical Analysis and Applications 247 (2000) 272–289.

[105] S. M. Djouadi, On the optimality of the proper orthogonal decomposition and balanced truncation, Proceedings of the IEEE Conference on Decision and Control (2008) 4221–4226.

[106] S. M. Djouadi, On the connection between balanced proper orthogonal decomposition, balanced truncation, and metric complexity theory for infinite dimensional systems, Proceedings of the 2010 American Control Conference, ACC 2010 (2010) 4911–4916.

[107] B. Unger, S. Gugercin, Kolmogorov n-widths for linear dynamical systems, Advances in Computational Mathematics (2019).

[108] C. Fyfe, A neural network for pca and beyond, Neural Processing Letters 6 (1997) 33–41.

[109] M. P. Holmes, A. G. Gray, C. Lee Isbell, QUIC-SVD: Fast SVD using cosine trees, Advances in Neural Information Processing Systems 21 - Proceedings of the 2008 Conference (2009) 673–680.

[110] R. Maulik, B. Lusch, P. Balaprakash, Reduced-order modeling of advection-dominated systems with recurrent neural networks and convolutional autoencoders, arXiv preprint arXiv:2002.00470 (2020).

[111] Y. Bengio, A. Courville, P. Vincent, Representation learning: A review and new perspectives, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (2013) 1798–1828.

[112] H. Bourlard, Y. Kamp, Auto-association by multilayer perceptrons and singular value decomposition, Biological Cybernetics 59 (1988) 291–294.

[113] E. Plaut, From principal subspaces to principal components with linear autoencoders, arXiv preprint arXiv:1804.10253 (2019). arXiv:1804.10253.

[114] J. Modersitzki, FAIR: Flexible Algorithms for Image Registration, Society for Industrial and Applied Mathematics., 2009.

[115] A. Mang, A. Gholami, C. Davatzikos, G. Biros, PDE-constrained optimization in medical image analysis, Optimization and Engineering 19 (2018) 765–812. arXiv:1803.00058.

[116] H. Lu, D. M. Tartakovsky, Lagrangian dynamic mode decomposition for construction of reduced-order models of advection-dominated phenomena, Journal of Computational Physics 407 (2020) 109229.

[117] H. Lu, D. M. Tartakovsky, Dynamic mode decomposition for construction of reduced-order models of hyperbolic problems with shocks, arXiv preprint arXiv:2003.12165 (2020). arXiv:2003.12165.

[118] P. C. Hansen, D. P. O'Leary, The use of the l-curve in the regularization of discrete ill-posed problems, SIAM Journal on Scientific Computing 14 (1993) 1487–1503.

[119] Q. Tan, L. Gao, Y.-K. Lai, S. Xia, Variational autoencoders for deforming 3D mesh models, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, 2018, pp. 5841–5850.

[120] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Computation 9 (1997) 1735–1780.

[121] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using rnn encoder-decoder for statistical machine translation, arXiv preprint arXiv:1406.1078 (2014). arXiv:1406.1078.

[122] E. J. Parish, K. T. Carlberg, Time-series machine-learning error models for approximate solutions to parameterized dynamical systems, arXiv preprint arXiv:1907.11822 (2019). arXiv:1907.11822.

[123] C. W. Rowley, T. Colonius, R. M. Murray, Model reduction for compressible flows using POD and Galerkin projection, Physica D: Nonlinear Phenomena 189 (2004) 115–129.

[124] I. Kalashnikova, M. F. Barone, S. Arunajatesan, B. G. van Bloemen Waanders, Construction of energy-stable projection-based reduced order models, Applied Mathematics and Computation 249 (2014) 569–596.

[125] D. Amsallem, C. Farhat, Stabilization of projection-based reduced-order models, International Journal for Numerical Methods in Engineering 91 (2012) 358–377.

[126] E. Rezaian, M. Wei, A hybrid stabilization approach for reduced-order models of compressible flows with shock-vortex interaction, International Journal for Numerical Methods in Engineering (2019).

[127] R. Mojgani, M. Balajewicz, Stabilization of linear time-varying reduced order models, a feedback controller approach, International Journal for Numerical Methods in Engineering (2020).

[128] D. Kirk, Optimal Control Theory: An Introduction, Dover Books on Electrical Engineering, Dover Publications, 1970.

[129] S. Bagheri, D. S. Henningson, J. Hœpffner, P. J. Schmid, Input-output analysis and control design applied to a linear model of spatially developing flows, Applied Mechanics Reviews 62 (2009) 1–27.

[130] K. K. Chen, C. W. Rowley, H_2 optimal actuator and sensor placement in the linearised complex Ginzburg-Landau system, Journal of Fluid Mechanics 681 (2011) 241–260.

[131] P. Sashittal, D. J. Bodony, Reduced-order control using low-rank dynamic mode decomposition, Theoretical and Computational Fluid Dynamics 33 (2019) 603–623.

[132] M. Tomás-Rodríguez, S. P. Banks, Linear, Time-varying Approximations to Nonlinear Dynamical Systems, volume 411 of *Lecture Notes in Control and Information Sciences*, Springer London, London, 2010.

[133] T. Bui-Thanh, K. Willcox, O. Ghattas, van Bloemen Waanders, Goal-oriented, model-constrained optimization for reduction of large-scale systems, Journal of Computational Physics 224 (2007) 880–896.

[134] S. Gugercin, A. C. Antoulas, A survey of model reduction by balanced truncation and some new results, International Journal of Control 77 (2004) 748–766.

[135] P. Benner, J. Saak, Numerical solution of large and sparse continuous time algebraic matrix Riccati and Lyapunov equations: a state of the art survey, GAMM-Mitteilungen 36 (2013) 32–52.

[136] M. Hached, K. Jbilou, Numerical solutions to large-scale differential Lyapunov matrix equations, Numerical Algorithms 79 (2017) 741–757.

[137] P. Benner, Z. Bujanović, P. Kürschner, J. Saak, RADI: a low-rank ADI-type algorithm for large scale algebraic Riccati equations, Numerische Mathematik 138 (2018) 301–330.

[138] A. Schmidt, B. Haasdonk, Reduced basis approximation of large scale parametric algebraic Riccati equations, ESAIM: Control, Optimisation and Calculus of Variations 24 (2018) 129–151.

[139] D. Palitta, V. Simoncini, Numerical methods for large-scale Lyapunov equations with symmetric banded data, SIAM Journal on Scientific Computing 40 (2018) A3581–A3608.

[140] D. Amsallem, C. Farhat, On the stability of reduced-order linearized computational fluid dynamics models based on POD and Galerkin projection: Descriptor vs non-descriptor forms, in: Reduced Order Methods for Modeling and Computational Reduction, Springer International Publishing, 2014, pp. 215–233.

[141] M. Bergmann, C.-H. Bruneau, A. Iollo, Enablers for robust POD models, Journal of Computational Physics 228 (2009) 516–538.

[142] M. Balajewicz, I. Tezaur, E. Dowell, Minimal subspace rotation on the Stiefel manifold for stabilization and enhancement of projection-based reduced-order models for the compressible Navier-Stokes equations, Journal of Computational Physics 321 (2016) 224–241.

[143] M.-s. Hossain, Numerical Methods for Model Reduction of Time-Varying Descriptor Systems, Ph.D. thesis, Chemnitz University of Technology, Chemnitz, 2011.

[144] S. Shokoohi, L. M. Silverman, P. Van Dooren, Linear time-variable systems: Balancing and model reduction, IEEE Transactions on Automatic Control 28 (1983) 810–822.

[145] H. Sandberg, A. Rantzer, Balanced truncation of linear time-varying systems, IEEE Transactions on Automatic Control 49 (2004) 217–229.

[146] Y. Chahlaoui, P. Van Dooren, Estimating gramians of large-scale time-varying systems, IFAC Proceedings Volumes 35 (2002) 325–330.

[147] N. Lang, J. Saak, T. Stykel, Balanced truncation model reduction for linear time-varying systems, Mathematical and Computer Modelling of Dynamical Systems 22 (2016) 267–281.

[148] W. J. Rugh, Linear system theory, Prentice Hall, Upper Saddle River, NJ, 1996.

[149] H. Rosenbrock, The stability of linear time-dependent control systems, Journal of Electronics and Control 15 (1963) 73–80.

[150] M. Malek-Zavarei, The stability of linear time-varying systems, International Journal of Control 27 (1978) 809–815.

[151] P. Mullhaupt, D. Buccieri, D. Bonvin, A numerical sufficiency test for the asymptotic stability of linear time-varying systems, Automatica 43 (2007) 631–638.

[152] Hiroaki Shiobara, N. Hori, Numerical exact discrete-time-model of linear time-varying systems, in: 2008 International Conference on Control, Automation and Systems, IEEE, 2008, pp. 2314–2318.

[153] A. Szyda, A numerical sufficiency test for the asymptotic stability of linear time-varying systems, Pomiary, Automatyka, Kontrola 56 (210) 1364–1367.

[154] A. Baccouche, H. Sayem, A numerical tool for stability analysis of linear time-varying systems, in: 2017 International Conference on Advanced Systems and Electric Technologies (IC_ASET), IEEE, 2017, pp. 323–328.

[155] B. Zhou, T. Zhao, On asymptotic stability of discrete-time linear time-varying systems, IEEE Transactions on Automatic Control 62 (2017) 4274–4281.

[156] J. Roychowdhury, Reduced-order modelling of linear time-varying systems, in: Proceedings of the 1998 IEEE/ACM International Conference on Computer-aided Design, ICCAD '98, ACM, New York, NY, 1998, pp. 92–95.

[157] J. R. Phillips, Projection-based approaches for model reduction of weakly nonlinear, time-varying systems, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 22 (2003) 171–187.

[158] Y. Chahlaoui, P. Van Dooren, Model reduction of time-varying systems, in: P. Benner, D. C. Sorensen, V. Mehrmann (Eds.), Dimension Reduction of Large-Scale Systems, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 131–148.

[159] P. Astrid, S. Weiland, K. Willcox, T. Backx, Missing point estimation in models described by proper orthogonal decomposition, IEEE Transactions on Automatic Control 53 (2008) 2237–2251.

[160] H. Mohammad-Sahadet, Projection-based model reduction for time-varying descriptor systems: New results, Numerical Algebra, Control and Optimization 6 (2016) 73.

[161] W. A. Coppel, Dichotomies in stability theory, Lecture Notes in Mathematics 629 (1978).

[162] A. Ilchmann, D. H. Owens, D. Prätzel-Wolters, Sufficient conditions for stability of linear time-varying systems, Systems & Control Letters 9 (1987) 157–163.

[163] P. Van Dooren, J. Sreedhar, When is a periodic discrete-time system equivalent to a time-invariant one?, Linear Algebra and its Applications 212-213 (1994) 131–151.

[164] S. Bittanti, P. Colaneri, Periodic Systems, Communications and Control Engineering, Springer London, London, UK, 2009.

[165] E. W. Kamen, The poles and zeros of a linear time-varying system, Linear Algebra and its Applications 98 (1988) 263–289.

[166] Y. Saad, Numerical Methods for Large Eigenvalue Problems, Society for Industrial and Applied Mathematics, 2011.

[167] G. H. Golub, V. C. Klema, G. W. Stewart, Rank Degeneracy and Least Squares Problems, Technical Report, Stanford University, Stanford, CA, USA, 1976.

[168] C. Canuto, M. Y. Hussaini, A. Quarteroni, T. A. Zang, Spectral Methods, Scientific Computation, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[169] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, J. N. Kutz, On dynamic mode decomposition: Theory and applications, Journal of Computational Dynamics 1 (2014) 391–421. arXiv:1312.0041.

[170] H. Babaee, T. P. Sapsis, A minimization principle for the description of modes associated with finite-time instabilities, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 472 (2016).

[171] M. Cruz Varona, B. Lohmann, Model Reduction of Linear Time-Varying Systems with Applications for Moving Loads, Springer International Publishing, Cham, 2017, pp. 367–386.

[172] P. Benner, S. Görner, J. Saak, Numerical solution of optimal control problems for parabolic systems, Parallel Algorithms and Cluster Computing, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 151–169.

[173] K. S. Tsakalis, P. A. Ionnou, Linear time-varying systems, control and adaptation, Prentice Hall, Englewood Cliffs, New Jersey, 1993.

[174] G. A. Sod, A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws, Journal of Computational Physics 27 (1978) 1–31.

[175] O. San, K. Kara, Numerical assessments of high-order accurate shock capturing schemes: Kelvin–Helmholtz type vortical structures in high-resolutions, Computers & Fluids 89 (2014) 254–276.

[176] S. Grimberg, C. Farhat, N. Youkilis, On the stability of projection-based model order reduction for convection-dominated laminar and turbulent flows, 2020.

[177] F. Chollet, et al., Keras, `https://keras.io`, 2015.

[178] MATLAB, version 9.0.0 (R2016a), The MathWorks Inc., Natick, Massachusetts, 2016.

[179] S. Gugercin, A. C. Antoulas, N. Bedrossian, Approximation of the international space station 1R and 12A models, Proceedings of the IEEE Conference on Decision and Control 2 (2001) 1515–1516.

[180] Y. Chahlaoui, P. Van Dooren, Benchmark examples for model reduction of linear time-invariant dynamical systems, Springer, Berlin, Heidelberg, 2005, pp. 379–392.

[181] I. Kalashnikova, S. Arunajatesan, M. F. Barone, B. G. V. B. Waanders, J. A. Fike, Reduced Order Modeling for Prediction and Control of Large-Scale Systems, Technical Report, Sandia National Laboratories, Albuquerque, New Mexico, 2014.

[182] W. Stankiewicz, R. Roszak, M. Morzyński, Arbitrary Lagrangian-Eulerian approach in reduced order modeling of a flow with a moving boundary, Progress in Flight Physics, EDP Sciences, Les Ulis, France, 2013, pp. 109–124.

[183] D. Amsallem, C. Farhat, An online method for interpolating linear parametric reduced-order models, SIAM Journal on Scientific Computing 33 (2011) 2169–2198.

[184] Y. Saad, Iterative Methods for Sparse Linear Systems, Society for Industrial and Applied Mathematics, 2003.

[185] S. Jain, P. Tiso, Simulation-free hyper-reduction for geometrically nonlinear structural dynamics: A quadratic manifold lifting approach, Journal of Computational and Nonlinear Dynamics 13 (2018) 071003.

[186] S. Chaturantabut, D. C. Sorensen, Nonlinear model reduction via discrete empirical interpolation, SIAM Journal on Scientific Computing 32 (2010) 2737–2764.

[187] K. Carlberg, C. Farhat, J. Cortial, D. Amsallem, The GNAT method for nonlinear model reduction: effective implementation and application to computational fluid dynamics and turbulent flows, Journal of Computational Physics 242 (2013) 623–647.

[188] C. Farhat, T. Chapman, P. Avery, Structure-preserving, stability, and accuracy properties of the energy-conserving sampling and weighting method for the hyper reduction of nonlinear finite element dynamic models, International Journal for Numerical Methods in Engineering 102 (2015) 1077–1110.

[189] R. Everson, L. Sirovich, Karhunen–Loève procedure for gappy data, Journal of the Optical Society of America A 12 (1995) 1657–1664.

[190] S. Balasuriya, N. T. Ouellette, I. I. Rypina, Generalized lagrangian coherent structures, Physica D: Nonlinear Phenomena 372 (2018) 31–51.

[191] M. R. Allshouse, J. L. Thiffeault, Detecting coherent structures using braids, Physica D: Nonlinear Phenomena 241 (2012) 95–105.