Large Eddy Simulation Reduced Order Models

Xuping Xie

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Mathematics

Traian Iliescu, Chair
Jeff T. Borggaard
Serkan Gugercin
Shane D. Ross

April 19, 2017
Blacksburg, Virginia

Keywords: Reduced Order Modeling, Large Eddy Simulation, Approximate Deconvolution, Data-Driven Modeling, Stochastic Reduced Order Model, Spatial Filtering, Finite Element, Numerical Analysis

Large Eddy Simulation Reduced Order Models

Xuping Xie

(ABSTRACT)

This dissertation uses spatial filtering to develop a large eddy simulation reduced order model (LES-ROM) framework for fluid flows. Proper orthogonal decomposition is utilized to extract the dominant spatial structures of the system. Within the general LES-ROM framework, two approaches are proposed to address the celebrated ROM closure problem. No phenomenological arguments (e.g., of eddy viscosity type) are used to develop these new ROM closure models.

The first novel model is the approximate deconvolution ROM (AD-ROM), which uses methods from image processing and inverse problems to solve the ROM closure problem. The AD-ROM is investigated in the numerical simulation of a 3D flow past a circular cylinder at a Reynolds number $Re = 1000$. The AD-ROM generates accurate results without any numerical dissipation mechanism. It also decreases the CPU time of the standard ROM by orders of magnitude.

The second new model is the calibrated-filtered ROM (CF-ROM), which is a data-driven ROM. The available full order model results are used offline in an optimization problem to calibrate the ROM subfilter-scale stress tensor. The resulting CF-ROM is tested numerically in the simulation of the 1D Burgers equation with a small diffusion parameter. The numerical results show that the CF-ROM is more efficient than and as accurate as state-of-the-art ROM closure models.

Large Eddy Simulation Reduced Order Models

Xuping Xie

(GENERAL AUDIENCE ABSTRACT)

Numerical simulation of complex fluid flows is often challenging in many realistic engineering, scientific, and medical applications. Indeed, an accurate numerical approximation of such flows generally requires millions and even billions of degrees of freedom. Furthermore, some design and control applications involve repeated numerical simulations for different parameter values. Reduced order models (ROMs) are an efficient approach to the numerical simulation of fluid flows, since they can reduce the computational time of a brute force computational approach by orders of magnitude while preserving key features of the flow.

Our main contribution to the field is the use of spatial filtering to develop better ROMs. To construct the new spatially filtered ROMs, we use ideas from image processing and inverse problems, as well as data-driven algorithms. The new ROMs are more accurate than standard ROMs in the numerical simulation of challenging three-dimensional flows past a circular cylinder.

Dedicated to my parents, Yichang Xie and Shaozhi Yang.

Dedicated to my wife, Yue Xing.

In memory of my grandparents, Delu Xie and Jiufen Zhang.

Acknowledgments

First of all, I was not sure how far I could go when I started my Ph.D. I am pleased to have this dissertation representing four years of research in model reduction. It was a great, challenging, and rewarding experience in my life. It would have taken much longer, and I would not have gotten this far without the help and support of many whom I am deeply indebted to.

My first and foremost thanks go to my advisor Dr. Traian Iliescu for his encouragement and constant support. It was his enthusiasm, patience and comprehensive knowledge that encouraged me to explore my research interests. Every question he brought up helped me understand better and keep my mind open in research. I learned to draw "big pictures" as well as move step by step. I really enjoyed working with him.

The Math Department at Virginia Tech is a very nice community that closely connects faculty and students to perform research in different disciplines. I am very grateful for having my committee members. I would like to thank Dr. Jeff Borggaard for his advice and insightful comments on the research. Thanks to Dr. Serkan Gugercin for teaching the model reduction class, which I got a deep understanding of ROM from another perspective, and also for the research advice. Thanks to Dr. Shane Ross for sharing his knowledge in Lagrangian coherent structures and helpful discussions in the research collaboration.

Big thanks to Dr. Honghu Liu for his support during my final semester, and his insightful ideas that encouraged me to extend my research to stochastic modeling. I would also like to thank Dr. Tao Lin for his many helpful answers to my questions in the Finite Element class.

I want to express my heartful thanks to my academic family members, Dr. David Wells and Dr. Zhu Wang for their guidance in scientific computing world, both in coding and theorem proving. Without their help, I would probably be stuck somewhere in the "ocean" of numerical simulation.

Staying in Blacksburg would not have been so enjoyable without my fellow graduate students in the Math Department. Thanks to Alan Lattimer for his help and wonderful conversations at ICAM. Thanks to Klajdi Sinani for so many helpful things that he gave me. Thanks to all my personal friends.

Lastly, I would like to give my special thanks to my wife, Yue Xing. I always appreciate her love, understanding, and support.

My years at Virginia Tech would not have been so great and memorable without all of you!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Numerical simulation of complex fluids often puts a heavy burden on the available computational resources in many realistic engineering, scientific, and medical applications, see [59, 60, 121, 144, 157]. Indeed, an accurate numerical approximation of realistic flows generally requires millions and even billions of degrees of freedom. Furthermore, some design and control applications involve repeated numerical simulations for different parameter values. Reduced order models (ROMs) are an efficient approach to the numerical simulation of linear and nonlinear systems [8, 9, 19, 22, 23, 24, 82, 118, 132, 137], since they can reduce the computational time of a direct numerical simulation (DNS) by orders of magnitude while preserving key features of the flow. ROMs have been successfully used in numerous applications, such as fluid flow optimization and control, cardiovascular flows, or geophysical fluid flows (see, e.g., the surveys in [15, 82, 85, 118, 132, 147]). The proper orthogonal decomposition (POD) is one of the most successful methods for ROM development. An accurate numerical simulation is used in POD to extract the dominant structures, which are then used in a Galerkin approximation of the underlying equations [85, 145]. In this work, POD will be exclusively used to construct the ROMs. We emphasize, however, that other ROM bases (e.g., the dynamic mode decomposition (DMS) [102, 137, 142]) could be used in the ROMs that we put forth.

Standard ROMs are extremely efficient and relatively accurate for laminar flows. They generally fail, however, in the numerical simulation of convection-dominated flows [12, 13, 16, 18, 76, 126, 133, 162]. Indeed, to ensure a low computational cost, only the first few POD modes are generally used in the ROM. The resulting low-dimensional ROM, however, generally yields poor results in the numerical simulation of convection-dominated flows, see, e.g., [12, 13, 45, 85, 104, 118, 125, 162]. Fig. 1.1 is the result of a 3D flow past a circular cylinder with $Re = 1000$. It indicates that the standard ROM cannot capture certain spatial structures and its velocity magnitude is too high; more details can be found in Chapter 3 and in [168].

In the ROM literature, a common explanation for this failure of standard ROMs is that they fail to model the effect of the discarded POD modes, see [57, 85, 12, 13, 162, 125, 131]. Over the years, numerous models for the effect of the neglected POD modes have been

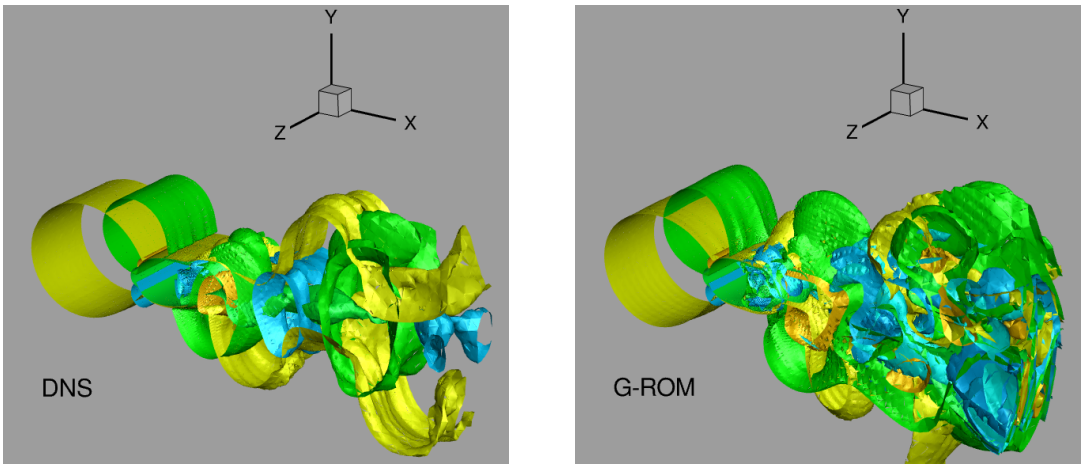


Figure 1.1: 3D flow past a circular cylinder at Reynolds number $Re = 1000$. The standard ROM yields an inaccurate numerical approximation in terms of structure and magnitude.

devised (see, e.g., [7, 12, 13, 16, 26, 131, 133, 162]). Although a survey of these approaches is beyond the scope of this dissertation, some of the most recent developments can be found in, e.g., [13, 121, 124, 133, 162] and references therein.

Most of the current ROM closure models are based on phenomenological arguments, such as the concept of energy cascade [12, 13, 45, 57, 85, 121, 124, 125, 131]. Thus, most of these ROM closure models (e.g., the eddy viscosity ROM closure models [162]) add numerical dissipation. In this dissertation, we use a *fundamentally different approach* to develop ROM closure models. Instead of phenomenological arguments, we use mathematical approximations. The concept of *spatial filtering* is the main tool used in our new framework.

The rest of the dissertation is organized as follows. In the rest of this chapter, we present some background material that will be used in what follows. In Chapter 2, we develop the LES-ROM framework, which we then use to develop the AD-ROM (Chapter 3) and CF-ROM (Chapter 4). In Chapter 5, we prove error estimates for the ROM spatial filter and we apply them to a spatial filtered ROM, the Leray ROM. In Chapter 6, we apply the Leray ROM to a stochastic Burgers equation. Finally, in Chapter 7 we present the conclusions and outline several directions of future research.

1.2 Navier-Stokes Equations (NSE)

In what follows, we use the incompressible Navier-Stokes equations (NSE) as mathematical model for fluid flow:

$$\boldsymbol{u}_t - \frac{1}{Re}\,\Delta \boldsymbol{u} + \boldsymbol{u}\cdot\nabla \boldsymbol{u} + \nabla p = \boldsymbol{f}\,, \tag{1.1}$$

$$\nabla\cdot\boldsymbol{u} = 0\,, \tag{1.2}$$

where $\boldsymbol{u}$ is the velocity, p the pressure, $\boldsymbol{f}$ the forcing, and Re the Reynolds number. In our work, the NSE (1.2)–(1.3) are supplemented with the initial condition $\boldsymbol{u}(\boldsymbol{x}, 0) = \boldsymbol{u}_0(\boldsymbol{x})$ and

steady Dirichlet boundary conditions. In Chapter 3, we use the following version of NSE,

$$\boldsymbol{u}_t - \frac{2}{Re}\nabla\cdot\mathbb{D}(\boldsymbol{u}) + \boldsymbol{u}\cdot\nabla\boldsymbol{u} + \nabla p = \boldsymbol{f}\,, \tag{1.3}$$

where $\mathbb{D}(\boldsymbol{u}) := (\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^\top)/2$ is the deformation tensor of $\boldsymbol{u}$.

In this section, we briefly introduce the necessary mathematical background for the NSE; more details can be found in [66, 106, 154] and Chapter 5. To this end, let

$$\boldsymbol{X} = \boldsymbol{H}_0^1(\Omega) = \Big\{\boldsymbol{v}\in[L^2(\Omega)]^n : \nabla\boldsymbol{v}\in[L^2(\Omega)]^{n\times n} \text{ and } \boldsymbol{v}=\boldsymbol{0} \text{ on } \partial\Omega\Big\}$$

be the Hilbert space of functions with square integrable derivatives that vanish on the boundary, let

$$Q = L_0^2(\Omega) = \Big\{q\in L^2(\Omega) : \int_\Omega q\,d\boldsymbol{x} = 0\Big\}$$

be the space of square integrable functions that vanish on the boundary, and let

$$\boldsymbol{V} = \{\boldsymbol{v}\in\boldsymbol{X} : (\nabla\cdot\boldsymbol{v}, q) = 0, \forall\, q\in Q\}$$

be the space of weakly divergence free functions.

1.3 Finite Element Discretization

In this section, we present the finite element discretization of the NSE, which we exclusively use in what follows.

Definition 1.1. *(strong solution)(see p.153, [106])* $(\boldsymbol{u}, p)$ *is a strong solution of (1.1) if* $\boldsymbol{u}\in L^2(0,T;\boldsymbol{X})\bigcap L^\infty(0,T;\boldsymbol{L}^2(\Omega))$ *and*

(a) $\boldsymbol{u} : [0,T]\to\boldsymbol{X}$ *is differentiable with* $\boldsymbol{u}_t : (0,T]\to\boldsymbol{X}^*$ *integrable, and* $p : (0,T]\to\boldsymbol{X}$ *is continuous.*

(b) For all $t'\in(0,T]$, $(\boldsymbol{u}, p)$ *satisfies*

$$\int_0^{t'}(\boldsymbol{u}_t,\boldsymbol{v}) + (\boldsymbol{u}\cdot\nabla\boldsymbol{u},\boldsymbol{v}) - (p,\nabla\cdot\boldsymbol{v}) + \nu(\nabla\boldsymbol{u},\nabla\boldsymbol{v})dt' = \int_0^{t'}(\boldsymbol{f},\boldsymbol{v})dt'$$

for all $\boldsymbol{v}\in L^2(0,T;H_0^1(\Omega))\bigcap L^\infty(0,T;\boldsymbol{L}^2(\Omega))$.

(c) $\boldsymbol{u}_0\in\boldsymbol{v}$ *and* $\|\boldsymbol{u}(t)-\boldsymbol{u}_0\|\to 0$ *as* $t\to 0$.

(d) $\boldsymbol{u}\in L^4(0,T;\boldsymbol{X})$.

1.3.1 Semi-Discretization

The semidiscrete or continuous in time finite element approximation of the NSE is

$$\begin{aligned}\left(\frac{\partial \boldsymbol{u}_h}{\partial t}, \boldsymbol{v}_h\right) + \nu(\nabla \boldsymbol{u}_h, \nabla \boldsymbol{v}_h) + b^*(\boldsymbol{u}_h, \boldsymbol{u}_h, \boldsymbol{v}_h) - (p_h, \nabla \cdot \boldsymbol{v}_h) = (\boldsymbol{f}, \boldsymbol{v}_h), \qquad & \forall\, \boldsymbol{v}_h \in \boldsymbol{X}^h, \\ (\nabla \cdot \boldsymbol{u}_h, q_h) = 0, & \forall\, q \in Q^h,\end{aligned} \tag{1.4}$$

where $\boldsymbol{X}^h \subset \boldsymbol{X}$, $Q^h \subset Q$, and $b^*(\boldsymbol{u}_h, \boldsymbol{v}_h, \boldsymbol{w}_h) = \frac{1}{2}\left[((\boldsymbol{u}_h \cdot \nabla)\boldsymbol{v}_h, \boldsymbol{w}_h) - ((\boldsymbol{u}_h \cdot \nabla)\boldsymbol{w}_h, \boldsymbol{v}_h)\right]$ is the skew-symmetrized trilinear form. The approximation $(\boldsymbol{u}_h, p_h)$ can be found in the appropriate spaces $(\boldsymbol{X}^h, Q^h)$. Furthermore, we know that if the approximation spaces satisfy the famous inf-sup condition, then the approximation system (1.4) is well defined and has a unique solution. The construction of the appropriate approximation spaces (basis functions) is beyond the scope of this dissertation; the intersted reader is refered to [36, 130, 152, 156].

Proposition 1.2. *(stability)(see p.154, [106]) The solution to (1.4), $\boldsymbol{u}_h$, is stable. For any $t > 0$,*

$$\frac{1}{2}\|\boldsymbol{u}_h\|^2 + \frac{\nu}{2}\int_0^t \|\nabla \boldsymbol{u}_h\|^2 dt' \leq \frac{1}{2}\|\boldsymbol{u}_h(t_0)\|^2 + \frac{1}{2\nu}\int_0^t \|\boldsymbol{f}(t')\|_{-1}^2 dt'.$$

1.3.2 Full Discretization

For illustration purposes, we present the trapezoidal rule, which is a popular time discretization method for the NSE [106]:

Trapezoidal Rule Given a time step $\Delta t > 0$, and $(\boldsymbol{u}_h^n, p_h^n) \in (\boldsymbol{X}^h, Q^h)$ find $(\boldsymbol{u}_h^{n+1}, p_h^{n+1}) \in (\boldsymbol{X}^h, Q^h)$ satisfying

$$\begin{aligned}\left(\frac{\boldsymbol{u}_h^{n+1} - \boldsymbol{u}_h^n}{\Delta t}, \boldsymbol{v}_h\right) + \nu\left(\nabla(\frac{\boldsymbol{u}_h^{n+1} + \boldsymbol{u}_h^n}{2}), \nabla \boldsymbol{v}_h\right) + b^*\left(\tfrac{\boldsymbol{u}_h^{n+1} + \boldsymbol{u}_h^n}{2}, \tfrac{\boldsymbol{u}_h^{n+1} + \boldsymbol{u}_h^n}{2}, \boldsymbol{v}_h\right) \\ - \left(\tfrac{p_{n+1}^h + p_n^h}{2}, \nabla \cdot \boldsymbol{v}_h\right) = \left(\boldsymbol{f}(t^{n+1/2}), \boldsymbol{v}_h\right), \\ (\nabla \cdot \boldsymbol{u}_h^{n+1}, q^h) = 0, \quad \forall (\boldsymbol{v}_h, q_h) \in (\boldsymbol{X}^h, Q^h)\end{aligned} \tag{1.5}$$

The trapezoidal method is A-stable and second order accurate [106]. Further details about full discretizations of the NSE can be found in [11, 35, 41, 154, 155, 169].

1.4 Proper Orthogonal Decomposition

One of the most popular reduced order modeling techniques is the POD, which we exclusively use to generate all the ROMs in this dissertation. We note, however, that other ROM bases (e.g., the dynamic mode decomposition (DMD), see [37, 38, 142, 153]) could be used. We briefly describe POD next. For more details, the reader is referred to, e.g., [85, 118, 145, 159]. The POD starts with the snapshots $\{\boldsymbol{u}_h^0, \ldots, \boldsymbol{u}_h^{N_s}\}$, which are numerical approximations of the NSE at N_s different time instances. For all the ROMs in this dissertation, we use DNS data to build the snapshots. Specifically, we utilize the FE approximations of the NSE

as snapshots (see Section 3.5 for details). We emphasize, however, that other numerical methods can be used instead. The POD seeks a low-dimensional basis that approximates the snapshots optimally with respect to a certain norm. We choose the commonly used L^2-norm.

Consider an ensemble of snapshots $\mathcal{R} := \operatorname{span}\{\boldsymbol{u}_h(\cdot,t_0),\ldots,\boldsymbol{u}_h(\cdot,t_M)\}$, which is a collection of velocity data from either numerical simulation results or experimental observations at time $t_i = i\,\Delta t,\ i = 0,\ldots,N_s$. The POD method seeks a low-dimensional basis $\{\boldsymbol{\varphi}_1,\ldots,\boldsymbol{\varphi}_r\}$ in $\boldsymbol{X}$ that optimally approximates the snapshots, i.e., solves the minimization problem: $\min \frac{1}{N_s+1}\sum_{\ell=0}^{N_s}\left\|\boldsymbol{u}_h(\cdot,t_\ell)-\sum_{j=1}^{r}\left(\boldsymbol{u}_h(\cdot,t_\ell),\boldsymbol{\varphi}_j(\cdot)\right)\boldsymbol{\varphi}_j(\cdot)\right\|^2$ subject to the conditions $(\boldsymbol{\varphi}_j,\boldsymbol{\varphi}_i) = \delta_{ij},\ 1\le i,j\le r$, where δ_{ij} is the Kronecker delta. To solve this minimization problem, one can consider the eigenvalue problem $K\,\boldsymbol{z}_j = \lambda_j\,\boldsymbol{z}_j$, for $j = 1,\ldots,r$, where $K\in\mathbb{R}^{(N_s+1)\times(N_s+1)}$ is the snapshot correlation matrix with entries $K_{k\ell} = \dfrac{1}{N_s+1}\left(\boldsymbol{u}_h(\cdot,t_\ell),\boldsymbol{u}_h(\cdot,t_k)\right)$ for $\ell,k = 0,\ldots,N_s$, $\boldsymbol{z}_j$ is the j-th eigenvector, and λ_j is the associated eigenvalue. The eigenvalues are positive and sorted in descending order $\lambda_1\ge\ldots\ge\lambda_d>0$, where d is the rank of $\mathcal{R}$. It can then be shown that the POD basis functions are given by $\boldsymbol{\varphi}_j(\cdot) = \frac{1}{\sqrt{\lambda_j}}\sum_{\ell=0}^{N_s}(\boldsymbol{z}_j)_\ell\,\boldsymbol{u}_h(\cdot,t_\ell),\ 1\le j\le r$, where $(\boldsymbol{z}_j)_\ell$ is the ℓ-th component of the eigenvector $\boldsymbol{z}_j$. It can also be shown that the following error formula holds [85, 100]:

$$\frac{1}{N_s+1}\sum_{\ell=0}^{N_s}\left\|\boldsymbol{u}_h(\cdot,t_\ell)-\sum_{j=1}^{r}\left(\boldsymbol{u}_h(\cdot,t_\ell),\boldsymbol{\varphi}_j(\cdot)\right)_{\mathcal{H}}\boldsymbol{\varphi}_j(\cdot)\right\|^2 = \sum_{j=r+1}^{d}\lambda_j\,. \tag{1.6}$$

We define the ROM space as $\boldsymbol{X}^r := \operatorname{span}\{\boldsymbol{\varphi}_1,\ldots,\boldsymbol{\varphi}_r\}$.

Since the POD basis functions are linear combinations of the snapshots, the POD basis functions satisfy the boundary conditions in (5.1) and are solenoidal. If the FE approximations are used as snapshots, the solution of the minimization problem is equivalent to the solution of the eigenvalue problem

$$\boldsymbol{Y}\,\boldsymbol{Y}^\top\boldsymbol{M}_h\,\boldsymbol{\varphi}_j = \lambda_j\,\boldsymbol{\varphi}_j,\quad j = 1,\ldots,N, \tag{1.7}$$

where $\boldsymbol{\varphi}_j$ and λ_j denote the vector of the FE coefficients of the POD basis functions and the POD eigenvalues, respectively, $\boldsymbol{Y}$ denotes the snapshot matrix, whose columns correspond to the FE coefficients of the snapshots, $\boldsymbol{M}_h$ denotes the FE mass matrix, and N is the dimension of the FE space $\boldsymbol{V}^h$ [42]. The eigenvalues are real and non-negative, so they can be ordered. The POD basis consists of the normalized functions $\{\boldsymbol{\varphi}_j\}_{j=1}^r$, which correspond to the first $r\ll N$ largest eigenvalues. Thus, the POD space is defined as $\boldsymbol{X}^r := \operatorname{span}\{\boldsymbol{\varphi}_1,\ldots,\boldsymbol{\varphi}_r\}$. The POD basis functions belong to $\boldsymbol{V}^h$, which yields $\boldsymbol{X}^r\subset\boldsymbol{V}^h$.

The centering trajectory method is popular in ROM development [85] (see [112] for recent developments). In this approach, the snapshots $\{\boldsymbol{u}_h^1,\ldots,\boldsymbol{u}_h^{N_s}\}$ are replaced by $\{\boldsymbol{u}_h^1-\boldsymbol{U},\ldots,\boldsymbol{u}_h^{N_s}-\boldsymbol{U}\}$, where $\boldsymbol{U} = \frac{1}{N_s}\sum_{i=1}^{N_s}\boldsymbol{u}_h^i$ is the centering trajectory and $\boldsymbol{u}_h^i-\boldsymbol{U}$ are the snapshot fluctuations. Thus, the POD basis functions are computed from the snapshot fluctuations $\boldsymbol{u}_h^i-\boldsymbol{U},\ i = 1,\ldots,N_s$. The centering trajectory method is especially useful for problems that have steady nonhomogeneous Dirichlet boundary conditions, such as the boundary conditions in the 3D flow past a cylinder investigated in Section 3.5. Since in this case the snapshot fluctuations $\boldsymbol{u}_h^i-\boldsymbol{U},\ i = 1,\ldots,N_s$ satisfy homogeneous Dirichlet boundary

conditions, the centering trajectory method avoids the challenges posed by the nonhomogeneous boundary conditions in ROMs (see, however, [78] for alternative approaches). In what follows, we will use the centering trajectory approach in the derivation of our new ROMs. We emphasize, however, that the new ROMs that we develop work equally well when the centering trajectory is not used.

Chapter 2

Large Eddy Simulation Reduced Order Model (LES-ROM)

This chapter proposes a large eddy simulation reduced order model (LES-ROM) framework for the numerical simulation of convection-dominated flows. In this LES-ROM framework, the POD is used to define the ROM basis and a ROM differential filter is used to define the large ROM structures.

We use the LES-ROM framework to construct a structural ROM. We first outline the LES-ROM framework that employs explicit ROM spatial filtering to develop new ROMs in which the term that needs to be modeled (i.e., the ROM stress tensor) is explicitly defined. We emphasize that, in this LES-ROM framework, we carry out explicit spatial filtering at a ROM level, not at the input data level. Thus, the new LES-ROMs can use either DNS input data (as we do in next Chapter 3, see Section 3.3 and Section 3.5) or LES input data (see discussion in Chapter 7).

2.1 Introduction

In this section, the POD, the standard Galerkin ROM and the centering trajectory are briefly presented. The NSE (1.3) are used as mathematical model. In our work, the NSE (1.3) are supplemented with the initial condition $\boldsymbol{u}(\boldsymbol{x},0) = \boldsymbol{u}_0(\boldsymbol{x})$ and steady Dirichlet boundary conditions.

Remark 2.1 (Viscous Term Formulation)**.** *In the NSE* (1.3)*, we used the viscous term formulation*

$$\frac{2}{Re}\,\nabla\cdot\mathbb{D}(\boldsymbol{u}) = \frac{1}{Re}\,\nabla\cdot\left[\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^{\top}\right]. \tag{2.1}$$

When $\nabla\cdot\boldsymbol{u} = 0$*, the viscous term formulation* (2.1) *is identical to the standard formulation*

$$\frac{1}{Re}\,\nabla\cdot\nabla\boldsymbol{u} = \frac{1}{Re}\,\Delta\boldsymbol{u}\,. \tag{2.2}$$

As noted in Section 3.3.1 of [79], however, the FE discretizations of the two formulations are usually not identical.

Both the continuous version and the FE discretization of the alternative viscous term formulation (2.1) *are carefully discussed in Section 3.11.3, Section 3.12.1 and Section 3.13.1 of [79]. We only note that, since* $\mathbb{D}(\boldsymbol{u}) = \mathbb{D}(\boldsymbol{u})^\top$*, the following inequality (which is equation (3.11-11) of [79]) holds:*

$$\Big(2\,\mathbb{D}(\boldsymbol{u}), \nabla \boldsymbol{u}\Big) = \Big(\mathbb{D}(\boldsymbol{u}), \nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^\top\Big) = 2\,\Big(\mathbb{D}(\boldsymbol{u}), \mathbb{D}(\boldsymbol{u})\Big) \geq 0\,. \tag{2.3}$$

Inequality (2.3) *and Korn's inequality (see Theorem 3.77 in [63]) imply that the matrix corresponding to the FE discretization of* (2.1) *in* $\boldsymbol{H}^1_0$ *is positive definite.*

In the remainder of the Chapter, we use the alternative viscous term formulation (2.1)*.*

To develop the standard Galerkin ROM, we start by considering the ROM velocity approximation, $\boldsymbol{U}_r$, defined as follows:

$$\boldsymbol{U}_r(\boldsymbol{x}, t) = \boldsymbol{U}(\boldsymbol{x}) + \sum_{j=1}^{r} a_j(t)\boldsymbol{\varphi}_j(\boldsymbol{x}) = \boldsymbol{U}(\boldsymbol{x}) + \boldsymbol{u}_r(\boldsymbol{x}, t)\,, \tag{2.4}$$

where $\{a_j(t)\}_{j=1}^r$ are the sought time-varying POD coefficients, $\boldsymbol{U}$ is the centering trajectory defined in Section 1.4, and $\boldsymbol{u}_r(\boldsymbol{x}, t) = \sum_{j=1}^r a_j(t)\boldsymbol{\varphi}_j(\boldsymbol{x})$ is the ROM velocity fluctuation. We use the POD basis, the fact that the POD modes are weakly divergence-free (just as the snapshots), and a Galerkin approximation of the NSE to obtain the standard *Galerkin ROM (G-ROM)* [85]: $\forall\, i = 1, \ldots, r$,

$$\left(\frac{\partial \boldsymbol{U}_r}{\partial t}, \boldsymbol{\varphi}_i\right) + \frac{2}{Re}\left(\mathbb{D}(\boldsymbol{U}_r), \nabla \boldsymbol{\varphi}_i\right) + \left(\boldsymbol{U}_r \cdot \nabla \boldsymbol{U}_r, \boldsymbol{\varphi}_i\right) = 0\,. \tag{2.5}$$

The G-ROM (2.5) yields the following autonomous dynamical system for the vector of time coefficients, $\mathbf{a}(t)$:

$$\dot{\mathbf{a}} = \mathbf{b} + \mathbf{A}\,\mathbf{a} + \mathbf{a}^\top \mathbf{B}\,\mathbf{a}, \tag{2.6}$$

where $\mathbf{b}$, $\mathbf{A}$, and $\mathbf{B}$ correspond to the constant, linear, and quadratic terms in the numerical discretization of the NSE (1.3)–(1.2), respectively. The initial conditions are obtained by projection:

$$a_j(0) = \Big(\boldsymbol{u}_0 - \mathbf{U}, \boldsymbol{\varphi}_j\Big)\,, \quad j = 1, \ldots, r. \tag{2.7}$$

The finite dimensional system (2.6) can be written componentwise as follows: For all $i = 1, \ldots, r$,

$$\dot{a}_i(t) = b_i + \sum_{m=1}^{r} A_{im} a_m(t) + \sum_{m=1}^{r}\sum_{n=1}^{r} B_{imn}\, a_m(t)\, a_n(t), \tag{2.8}$$

where

$$b_i = -\Big(\mathbf{U} \cdot \nabla \mathbf{U}, \boldsymbol{\varphi}_i\Big) - \frac{2}{Re}\left(\frac{\nabla \mathbf{U} + \nabla \mathbf{U}^\top}{2}, \nabla \boldsymbol{\varphi}_i\right), \tag{2.9}$$

$$A_{im} = -\Big(\mathbf{U} \cdot \nabla \boldsymbol{\varphi}_m, \boldsymbol{\varphi}_i\Big) - \Big(\boldsymbol{\varphi}_m \cdot \nabla \mathbf{U}, \boldsymbol{\varphi}_i\Big) - \frac{2}{Re}\left(\frac{\nabla \boldsymbol{\varphi}_m + \nabla \boldsymbol{\varphi}_m{}^\top}{2}, \nabla \boldsymbol{\varphi}_i\right), \tag{2.10}$$

$$B_{imn} = -\Big(\boldsymbol{\varphi}_m \cdot \nabla \boldsymbol{\varphi}_n, \boldsymbol{\varphi}_i\Big)\,. \tag{2.11}$$

Next, we present the component-wise formulation of equations (2.9)–(2.11), which is used in the computational implementation. To this end, let $(\cdot)_x$, $(\cdot)_y$, and $(\cdot)_z$ denote the partial derivatives with respect to x, y, and z, respectively. Furthermore, let $\boldsymbol{\varphi}_k^u$, $\boldsymbol{\varphi}_k^v$, and $\boldsymbol{\varphi}_k^w$ denote the $x-$, $y-$, and $z-$components of the POD basis function $\boldsymbol{\varphi}_k$.

$$\boldsymbol{U}\cdot\nabla\boldsymbol{U} = \begin{bmatrix} U & V & W \end{bmatrix}\cdot\begin{bmatrix} U_x & V_x & W_x \\ U_y & V_y & W_y \\ U_z & V_z & W_z \end{bmatrix} = \begin{bmatrix} UU_x+VU_y+WU_z \\ UV_x+VV_y+WV_z \\ UW_x+VW_y+WW_z \end{bmatrix}^T, \tag{2.12}$$

$$\begin{aligned} (\boldsymbol{U}\cdot\nabla\boldsymbol{U},\boldsymbol{\varphi}_k) &= (UU_x,\boldsymbol{\varphi}_k^u)+(VU_y,\boldsymbol{\varphi}_k^u)+(WU_z,\boldsymbol{\varphi}_k^u) \\ &+ (UV_x,\boldsymbol{\varphi}_k^v)+(VV_y,\boldsymbol{\varphi}_k^v)+(WV_z,\boldsymbol{\varphi}_k^v) \\ &+ (UW_x,\boldsymbol{\varphi}_k^w)+(VW_y,\boldsymbol{\varphi}_k^w)+(WW_z,\boldsymbol{\varphi}_k^w), \end{aligned} \tag{2.13}$$

$$\nabla\boldsymbol{U} = \begin{bmatrix} U_x & V_x & W_x \\ U_y & V_y & W_y \\ U_z & V_z & W_z \end{bmatrix}, \nabla\boldsymbol{\varphi}_k = \begin{bmatrix} \boldsymbol{\varphi}_x^u & \boldsymbol{\varphi}_x^v & \boldsymbol{\varphi}_x^w \\ \boldsymbol{\varphi}_y^u & \boldsymbol{\varphi}_y^v & \boldsymbol{\varphi}_y^w \\ \boldsymbol{\varphi}_z^u & \boldsymbol{\varphi}_z^v & \boldsymbol{\varphi}_z^w \end{bmatrix}, \nabla\boldsymbol{U}^T = \begin{bmatrix} U_x & U_y & U_z \\ V_x & V_y & V_z \\ W_x & W_y & W_z \end{bmatrix},$$

$$\begin{aligned} (\nabla\boldsymbol{U},\nabla\boldsymbol{\varphi}_k) &= (U_x,\boldsymbol{\varphi}_{kx}^u)+(V_x,\boldsymbol{\varphi}_{ky}^u)+(W_x,\boldsymbol{\varphi}_{kz}^u) \\ &+ (U_y,\boldsymbol{\varphi}_{kx}^v)+(V_y,\boldsymbol{\varphi}_{ky}^v)+(W_y,\boldsymbol{\varphi}_{kz}^v) \\ &+ (U_z,\boldsymbol{\varphi}_{kx}^w)+(V_z,\boldsymbol{\varphi}_{ky}^w)+(W_z,\boldsymbol{\varphi}_{kz}^w), \end{aligned}$$

$$\begin{aligned} (\nabla\boldsymbol{U}^T,\nabla\boldsymbol{\varphi}_k) &= (U_x,\boldsymbol{\varphi}_{kx}^u)+(U_y,\boldsymbol{\varphi}_{ky}^u)+(U_z,\boldsymbol{\varphi}_{kz}^u) \\ &+ (V_x,\boldsymbol{\varphi}_{kx}^v)+(V_y,\boldsymbol{\varphi}_{ky}^v)+(V_z,\boldsymbol{\varphi}_{kz}^v) \\ &+ (W_x,\boldsymbol{\varphi}_{kx}^w)+(W_y,\boldsymbol{\varphi}_{ky}^w)+(W_z,\boldsymbol{\varphi}_{kz}^w), \end{aligned}$$

$$\begin{aligned} (\boldsymbol{U}\cdot\nabla\boldsymbol{\varphi}_m,\boldsymbol{\varphi}_k) &= \left(\begin{bmatrix} U & V & W \end{bmatrix}\cdot\begin{bmatrix} \boldsymbol{\varphi}_{mx}^u & \boldsymbol{\varphi}_{mx}^v & \boldsymbol{\varphi}_{mx}^w \\ \boldsymbol{\varphi}_{my}^u & \boldsymbol{\varphi}_{my}^v & \boldsymbol{\varphi}_{my}^w \\ \boldsymbol{\varphi}_{mz}^u & \boldsymbol{\varphi}_{mz}^v & \boldsymbol{\varphi}_{mz}^w \end{bmatrix},\boldsymbol{\varphi}_k\right) \\ &= (U\boldsymbol{\varphi}_{mx}^u,\boldsymbol{\varphi}_k^u)+(V\boldsymbol{\varphi}_{my}^u,\boldsymbol{\varphi}_k^u)+(W\boldsymbol{\varphi}_{mz}^u,\boldsymbol{\varphi}_k^u) \\ &+ (U\boldsymbol{\varphi}_{mx}^v,\boldsymbol{\varphi}_k^v)+(V\boldsymbol{\varphi}_{my}^v,\boldsymbol{\varphi}_k^v)+(W\boldsymbol{\varphi}_{mz}^v,\boldsymbol{\varphi}_k^v) \\ &+ (U\boldsymbol{\varphi}_{mx}^w,\boldsymbol{\varphi}_k^w)+(V\boldsymbol{\varphi}_{my}^w,\boldsymbol{\varphi}_k^w)+(W\boldsymbol{\varphi}_{mz}^w,\boldsymbol{\varphi}_k^w), \end{aligned} \tag{2.14}$$

$$\begin{aligned} (\boldsymbol{\varphi}_m\cdot\nabla\boldsymbol{U},\boldsymbol{\varphi}_k) &= \left(\begin{bmatrix} \boldsymbol{\varphi}_m^u & \boldsymbol{\varphi}_m^v & \boldsymbol{\varphi}_m^w \end{bmatrix}\cdot\begin{bmatrix} U_x & V_x & W_x \\ U_y & V_y & W_y \\ U_z & V_z & W_z \end{bmatrix},\boldsymbol{\varphi}_k\right) \\ &= (\boldsymbol{\varphi}_m^u U_x,\boldsymbol{\varphi}_k^u)+(\boldsymbol{\varphi}_m^v U_y,\boldsymbol{\varphi}_k^u)+(\boldsymbol{\varphi}_m^w U_z,\boldsymbol{\varphi}_k^u) \\ &+ (\boldsymbol{\varphi}_m^u V_x,\boldsymbol{\varphi}_k^v)+(\boldsymbol{\varphi}_m^v V_y,\boldsymbol{\varphi}_k^v)+(\boldsymbol{\varphi}_m^w V_z,\boldsymbol{\varphi}_k^v) \\ &+ (\boldsymbol{\varphi}_m^u W_x,\boldsymbol{\varphi}_k^w)+(\boldsymbol{\varphi}_m^v W_y,\boldsymbol{\varphi}_k^w)+(\boldsymbol{\varphi}_m^w W_z,\boldsymbol{\varphi}_k^w), \end{aligned} \tag{2.15}$$

$$
\begin{aligned}
(\nabla\boldsymbol{\varphi}_m, \nabla\boldsymbol{\varphi}_k) &= (\boldsymbol{\varphi}^u_{mx}, \boldsymbol{\varphi}^u_{kx}) + (\boldsymbol{\varphi}^v_{mx}, \boldsymbol{\varphi}^u_{ky}) + (\boldsymbol{\varphi}^w_{mx}, \boldsymbol{\varphi}^u_{kz}) \\
&+ (\boldsymbol{\varphi}^u_{my}, \boldsymbol{\varphi}^v_{kx}) + (\boldsymbol{\varphi}^v_{my}, \boldsymbol{\varphi}^v_{ky}) + (\boldsymbol{\varphi}^w_{my}, \boldsymbol{\varphi}^v_{kz}) \\
&+ (\boldsymbol{\varphi}^u_{mz}, \boldsymbol{\varphi}^w_{kx}) + (\boldsymbol{\varphi}^v_{mz}, \boldsymbol{\varphi}^w_{ky}) + (\boldsymbol{\varphi}^w_{mz}, \boldsymbol{\varphi}^w_{kz}) \qquad (2.16)
\end{aligned}
$$

$$
\begin{aligned}
(\nabla\boldsymbol{\varphi}^T_m, \nabla\boldsymbol{\varphi}_k) &= (\boldsymbol{\varphi}^u_{mx}, \boldsymbol{\varphi}^u_{kx}) + (\boldsymbol{\varphi}^u_{my}, \boldsymbol{\varphi}^u_{ky}) + (\boldsymbol{\varphi}^u_{mz}, \boldsymbol{\varphi}^u_{kz}) \\
&+ (\boldsymbol{\varphi}^v_{mx}, \boldsymbol{\varphi}^v_{kx}) + (\boldsymbol{\varphi}^v_{my}, \boldsymbol{\varphi}^v_{ky}) + (\boldsymbol{\varphi}^v_{mz}, \boldsymbol{\varphi}^v_{kz}) \\
&+ (\boldsymbol{\varphi}^w_{mx}, \boldsymbol{\varphi}^w_{kx}) + (\boldsymbol{\varphi}^w_{my}, \boldsymbol{\varphi}^w_{ky}) + (\boldsymbol{\varphi}^w_{mz}, \boldsymbol{\varphi}^w_{kz}) \qquad (2.17)
\end{aligned}
$$

$$
\begin{aligned}
(\boldsymbol{\varphi}_m \cdot \nabla\boldsymbol{\varphi}_n, \boldsymbol{\varphi}_k) &= \left(\begin{bmatrix} \boldsymbol{\varphi}^u_m, & \boldsymbol{\varphi}^v_m, & \boldsymbol{\varphi}^w_m \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{\varphi}^u_{nx} & \boldsymbol{\varphi}^v_{nx} & \boldsymbol{\varphi}^w_{nx} \\ \boldsymbol{\varphi}^u_{ny} & \boldsymbol{\varphi}^v_{ny} & \boldsymbol{\varphi}^w_{ny} \\ \boldsymbol{\varphi}^u_{nz} & \boldsymbol{\varphi}^v_{nz} & \boldsymbol{\varphi}^w_{nz} \end{bmatrix}, \boldsymbol{\varphi}_k \right) \\
&= (\boldsymbol{\varphi}^u_m \boldsymbol{\varphi}^u_{nx}, \boldsymbol{\varphi}^u_k) + (\boldsymbol{\varphi}^v_m \boldsymbol{\varphi}^u_{ny}, \boldsymbol{\varphi}^u_k) + (\boldsymbol{\varphi}^w_m \boldsymbol{\varphi}^u_{nz}, \boldsymbol{\varphi}^u_k) \\
&+ (\boldsymbol{\varphi}^u_m \boldsymbol{\varphi}^v_{nx}, \boldsymbol{\varphi}^v_k) + (\boldsymbol{\varphi}^v_m \boldsymbol{\varphi}^v_{ny}, \boldsymbol{\varphi}^v_k) + (\boldsymbol{\varphi}^w_m \boldsymbol{\varphi}^v_{nz}, \boldsymbol{\varphi}^v_k) \\
&+ (\boldsymbol{\varphi}^u_m \boldsymbol{\varphi}^w_{nx}, \boldsymbol{\varphi}^w_k) + (\boldsymbol{\varphi}^v_m \boldsymbol{\varphi}^w_{ny}, \boldsymbol{\varphi}^w_k) + (\boldsymbol{\varphi}^w_m \boldsymbol{\varphi}^w_{nz}, \boldsymbol{\varphi}^w_k), \qquad (2.18)
\end{aligned}
$$

$$
\begin{aligned}
(\boldsymbol{\varphi}_m, \boldsymbol{\varphi}_k) &= \left((\boldsymbol{\varphi}^u_m, \boldsymbol{\varphi}^v_m, \boldsymbol{\varphi}^w_m), (\boldsymbol{\varphi}^u_k, \boldsymbol{\varphi}^v_k, \boldsymbol{\varphi}^w_k) \right) \\
&= (\boldsymbol{\varphi}^u_m, \boldsymbol{\varphi}^u_k) + (\boldsymbol{\varphi}^v_m, \boldsymbol{\varphi}^v_m) + (\boldsymbol{\varphi}^w_m, \boldsymbol{\varphi}^w_k). \qquad (2.19)
\end{aligned}
$$

2.2 LES-ROM Framework

Previous Relevant Work The LES-ROM framework that we develop in this chapter has been also utilized in, e.g., [12, 85, 162]. We emphasize that, to our knowledge, the LES-ROM framework has only been used *implicitly*, to motivate various functional ROM closure models [12, 85]. In this dissertation, we take a completely different approach and *explicitly* use the LES-ROM framework (specifically, the ROM stress tensor) to develop a structural ROM closure model, the AD-ROM.

In ROMs, spatial filtering has been used as a preprocessing step, to filter out the noise in the snapshot data, i.e., in the generation of the POD modes (see, e.g., Section 5 in [10] for a survey of relevant work). We emphasize, however, that our approach is *fundamentally different*, since we *explicitly* use spatial filters in the *actual ROMs*, i.e., we modify the mathematical model, not the input data. To our knowledge, explicit ROM spatial filtering has only be used in [138, 162, 165].

In this section, we exclusively use ROM spatial filtering to develop the LES-ROM framework and the AD-ROM. A different filtering approach was used in [121]: Instead of ROM spatial filtering, the authors used *ROM ensemble averaging* to develop the ensemble averaged ROM, called the Galerkin-Reynolds equation (see equation (7) in [121]). The ensemble averaged ROM proposed in [121] has, in our opinion, great potential. We emphasize, however, that the LES-ROM that we propose in this chapter and the ensemble averaged ROM proposed in [121] are *fundamentally different*: The LES-ROM employs the LES methodology

to develop a dynamical system for the spatial averages of the ROM velocity. The ensemble averaged ROM, on the other hand, is reminiscent of RANS formalism and consists of a dynamical system for the ensemble average of the ROM coefficients.

In this section, we develop the LES-ROM framework. In section 2.2.1, we present the explicit spatial filter. In section 2.2, we use this filter to develop the LES-ROM framework. In section 2.3, we present standard ROM closure models, which are based on phenomenological arguments. Finally, in section 2.3, we summarize other approaches that are similar to the LES-ROM framework.

2.2.1 Explicit ROM Spatial Filtering

To develop the new LES-ROM framework, we use the explicit *differential filter (DF)*. We present the continuous formulation of the DF, and two of its discrete formulations.

The DF has been pioneered in LES by Germano [71, 72]. Later, it was also used in a ROM context to develop regularized ROMs: The ROM-DF was first used in [138] for the 1D Kuramoto–Sivashinsky equation in a periodic setting. The ROM-DF was subsequently used in [165] for the 3D NSE in a general non-periodic setting.

Continuous Differential Filter (C-DF)

The *continuous differential filter (C-DF)* has been extensively used to develop LES models [107]. It is defined as follows (see, e.g., Section 2.2.6 in [107]): Let δ be the radius of the C-DF. For all $\boldsymbol{\Phi} \in \boldsymbol{L}^2(\Omega)$, find $\overline{\boldsymbol{\Phi}} \in \boldsymbol{L}^2(\Omega)$ such that

$$-\delta^2 \Delta\overline{\boldsymbol{\Phi}} + \overline{\boldsymbol{\Phi}} = \boldsymbol{\Phi} \qquad \text{in } \Omega \tag{2.20}$$

$$\overline{\boldsymbol{\Phi}} = \boldsymbol{0} \qquad \text{on } \partial\Omega\,. \tag{2.21}$$

The C-DF (2.20)–(2.21) is a linear spatial filter, which employs an elliptic operator to smooth the input variables and eliminate the small scales (i.e., high frequencies) [27].

The following lemma was proved in [95]:

Lemma 2.2. *The C-DF* (2.20)–(2.21) *is a self-adjoint operator on* $\boldsymbol{L}^2(\Omega)$.

Proof. Consider $\boldsymbol{\Phi}, \boldsymbol{\Psi} \in \boldsymbol{L}^2(\Omega)$. By the definition of the C-DF, we have

$$-\delta^2 \Delta\overline{\boldsymbol{\Psi}} + \overline{\boldsymbol{\Psi}} = \boldsymbol{\Psi} \qquad \text{in } \Omega \tag{2.22}$$

$$\overline{\boldsymbol{\Psi}} = \boldsymbol{0} \qquad \text{on } \partial\Omega\,. \tag{2.23}$$

Multiplying (2.22) by $\overline{\boldsymbol{\Phi}}$, integrating by parts, and using (2.21), we obtain

$$\delta^2 \left(\nabla\overline{\boldsymbol{\Psi}}, \nabla\overline{\boldsymbol{\Phi}}\right) + \left(\overline{\boldsymbol{\Psi}}, \overline{\boldsymbol{\Phi}}\right) = \left(\boldsymbol{\Psi}, \overline{\boldsymbol{\Phi}}\right) \qquad \text{in } \Omega\,. \tag{2.24}$$

Multiplying (2.20) by $\overline{\boldsymbol{\Psi}}$, integrating by parts, and using (2.23), we obtain

$$\delta^2 \left(\nabla\overline{\boldsymbol{\Phi}}, \nabla\overline{\boldsymbol{\Psi}}\right) + \left(\overline{\boldsymbol{\Phi}}, \overline{\boldsymbol{\Psi}}\right) = \left(\boldsymbol{\Phi}, \overline{\boldsymbol{\Psi}}\right) \qquad \text{in } \Omega\,. \tag{2.25}$$

Equations (2.24) and (2.25) prove the lemma. □ □

FE Differential Filter (FE-DF)

The *FE differential filter (FE-DF)* is defined as follows [27, 107]: Let δ be the radius of the FE-DF. For all $\boldsymbol{u}^h \in \boldsymbol{X}^h$ (where $\boldsymbol{X}^h$ is the FE space), find $\overline{\boldsymbol{u}^h}^h \in \boldsymbol{X}^h$ such that

$$-\delta^2 \Delta \overline{\boldsymbol{u}^h}^h + \overline{\boldsymbol{u}^h}^h = \boldsymbol{u}^h \qquad \text{in } \Omega \tag{2.26}$$

$$\overline{\boldsymbol{u}^h}^h = \boldsymbol{0} \qquad \text{on } \partial\Omega\,, \tag{2.27}$$

which yields the following linear system:

$$\left(\boldsymbol{M}_h + \delta^2\, \boldsymbol{S}_h\right) \overline{\boldsymbol{U}^h}^h = \boldsymbol{M}_h\, \boldsymbol{U}^h\,, \tag{2.28}$$

where $\boldsymbol{M}_h \in \mathbb{R}^{N\times N}$ is the FE mass matrix, $\boldsymbol{S}_h \in \mathbb{R}^{N\times N}$ is the FE stiffness matrix, $\overline{\boldsymbol{U}^h}^h \in \mathbb{R}^N$ is the vector of FE coefficients of the output filtered variable $\overline{\boldsymbol{u}^h}^h$, and $\boldsymbol{U}^h \in \mathbb{R}^N$ is the vector of FE coefficients of the input variable $\boldsymbol{u}^h$.

ROM Differential Filter (ROM-DF)

The *ROM differential filter (ROM-DF)* is defined as follows: Let δ be the radius of the ROM-DF. For all $\boldsymbol{u}^r \in \mathbf{X}^r$, find $\overline{\boldsymbol{u}^r}^r \in \boldsymbol{X}^r$ such that

$$-\delta^2 \Delta \overline{\boldsymbol{u}^r}^r + \overline{\boldsymbol{u}^r}^r = \boldsymbol{u}^r \qquad \text{in } \Omega \tag{2.29}$$

$$\overline{\boldsymbol{u}^r}^r = \boldsymbol{0} \qquad \text{on } \partial\Omega\,, \tag{2.30}$$

which yields the following linear system:

$$\left(\boldsymbol{M}_r + \delta^2\, \boldsymbol{S}_r\right) \overline{\boldsymbol{a}_r}^r = \boldsymbol{a}^r\,, \tag{2.31}$$

where $\boldsymbol{M}_r \in \mathbb{R}^{r\times r}$ is the POD mass matrix with entries $(\boldsymbol{M}_r)_{ij} = (\boldsymbol{\varphi}_j, \boldsymbol{\varphi}_i)$, $\boldsymbol{S}_r \in \mathbb{R}^{r\times r}$ is the POD stiffness matrix with entries $(\boldsymbol{S}_r)_{ij} = (\nabla\boldsymbol{\varphi}_j, \nabla\boldsymbol{\varphi}_i)$, $\overline{\boldsymbol{a}_r}^r \in \mathbb{R}^r$ is the vector of POD coefficients of the output filtered variable $\overline{\boldsymbol{u}^r}^r$, and $\boldsymbol{a} \in \mathbb{R}^r$ is the vector of POD coefficients of the input variable $\boldsymbol{u}^r$.

We emphasize that the computational cost of the ROM-DF is much lower than the computational cost of the FE-DF, since the former yields a very small $r \times r$ linear system, whereas the latter yields a much larger, $N \times N$ linear system.

2.2.2 Large Eddy Simulation ROM (LES-ROM)

To develop the LES-ROM, we use the standard LES approach [27, 75, 93, 134, 139]:

1. Filter the NSE to obtain the *spatially filtered NSE (SF-NSE)*.

2. Use the SF-NSE and the ROM approximation to obtain the LES-ROM.

The motivation for using the above strategy in the LES field is straightforward: The SF-NSE are a system of equations for the filtered flow variables, which can be approximated on the given coarse mesh more accurately than the original, unfiltered flow variables. In this section, we use a similar argument to develop the LES-ROM framework: Since the spatial structures in the SF-NSE are larger than the spatial structures in the NSE, we expect that, for a fixed target numerical accuracy of the ROM, the SF-NSE require fewer POD modes than the NSE. This, of course, is advantageous from a computational point of view in challenging, realistic applications.

In the remainder of this section, we first present the derivation of the SF-NSE and the associated closure problem. Then, we outline the general LES-ROM framework. In chapter 3, we use this LES-ROM framework to develop the new AD-ROM.

We filter the NSE (1.3) with the continuous ROM differential filter, C-DF, and assume that differentiation and filtering commute (see, e.g., Chapter 9 in [27]):

$$\frac{\partial \overline{\boldsymbol{u}}}{\partial t} - \frac{2}{Re}\,\nabla \cdot \mathbb{D}(\overline{\boldsymbol{u}}) + \overline{\boldsymbol{u} \cdot \nabla \boldsymbol{u}} + \nabla \overline{p} = \boldsymbol{0}\,. \tag{2.32}$$

Next, we project (2.32) onto a space of weakly divergence-free functions $\boldsymbol{\phi}$ to obtain the weak formulation of the *spatially filtered NSE (SF-NSE)*:

$$\left(\frac{\partial \overline{\boldsymbol{u}}}{\partial t}, \boldsymbol{\phi}\right) + \frac{2}{Re}\left(\mathbb{D}(\overline{\boldsymbol{u}}), \nabla \boldsymbol{\phi}\right) + \left(\overline{\boldsymbol{u} \cdot \nabla \boldsymbol{u}}, \boldsymbol{\phi}\right) = \boldsymbol{0}\,. \tag{2.33}$$

Remark 2.3. *The standard SF-NSE generally include a pressure term. Since we used weakly divergence-free test functions $\boldsymbol{\phi}$, the pressure term from* (2.32) *dropped out of* (2.33)*. We used this approach since the POD basis functions employed in our ROMs are weakly divergence-free. Thus, the SF-NSE* (2.33)*, although nonstandard, are better suited to our ROM framework.*

Of course, in LES one cannot simply discretize the SF-NSE (2.33), since these equations are not closed. Thus, one first needs to address the *closure problem*, i.e., to model the filtered nonlinearity $\overline{\boldsymbol{u} \cdot \nabla \boldsymbol{u}}$ in terms of the filtered velocity $\overline{\boldsymbol{u}}$:

$$\overline{\boldsymbol{u} \cdot \nabla \boldsymbol{u}} = \text{function}(\overline{\boldsymbol{u}})\,. \tag{2.34}$$

The *closure model* is the explicit dependence on $\overline{\boldsymbol{u}}$ of the filtered nonlinearity $\overline{\boldsymbol{u} \cdot \nabla \boldsymbol{u}}$ in equation (2.34). Once the closure problem is addressed, the SF-NSE (2.33) yield the *LES model*:

$$\left(\frac{\partial \overline{\boldsymbol{u}}}{\partial t}, \boldsymbol{\phi}\right) + \frac{2}{Re}\left(\mathbb{D}(\overline{\boldsymbol{u}}), \nabla \boldsymbol{\phi}\right) + \left(\text{function}(\overline{\boldsymbol{u}}), \boldsymbol{\phi}\right) = \boldsymbol{0}\,. \tag{2.35}$$

Of course, the LES model (2.35) needs to be discretized. We use a ROM approximation and obtain the *LES-ROM*: $\forall\, i = 1, \ldots, r$,

$$\left(\frac{\partial \boldsymbol{W}_r}{\partial t}, \boldsymbol{\varphi}_i\right) + \frac{2}{Re}\left(\mathbb{D}(\boldsymbol{W}_r), \nabla \boldsymbol{\varphi}_i\right) + \left(\text{function}(\boldsymbol{W}_r), \boldsymbol{\varphi}_i\right) = \boldsymbol{0}\,, \tag{2.36}$$

where $\boldsymbol{W}_r$ is the ROM approximation of the filtered velocity $\overline{\boldsymbol{u}}$ in (2.35).

Remark 2.4. *In the LES literature, the ROM closure problem generally takes a form that is different from* (2.34)*. In this chapter, however, we use the equivalent form in* (2.34)*, since just as in LES [27, 139, 107], it makes the derivation of the AD-ROM in chapter 3 straightforward.*

For completeness, however, we outline the derivation of the standard formulation of the LES closure problem. First, in the NSE, we use the fact that $\nabla \cdot \boldsymbol{u} = 0$ *to rewrite the nonlinear term as follows:* $\boldsymbol{u} \cdot \nabla \boldsymbol{u} = \nabla \cdot (\boldsymbol{u}\,\boldsymbol{u})$*. Thus, the SF-NSE* (2.33) *can be written as*

$$\left(\frac{\partial \overline{\boldsymbol{u}}}{\partial t}, \boldsymbol{\phi}\right) + \frac{2}{Re}\left(\mathbb{D}(\overline{\boldsymbol{u}}), \nabla \boldsymbol{\phi}\right) + \left(\overline{\nabla \cdot \left(\boldsymbol{u}\,\boldsymbol{u}\right)}, \boldsymbol{\phi}\right) = \boldsymbol{0}\,. \tag{2.37}$$

Next, we decompose the nonlinearity in (2.37) *as follows:*

$$\overline{\boldsymbol{u}\,\boldsymbol{u}} = \overline{\boldsymbol{u}}\;\overline{\boldsymbol{u}} + \left(\overline{\boldsymbol{u}\;\boldsymbol{u}} - \overline{\boldsymbol{u}}\;\overline{\boldsymbol{u}}\right) = \overline{\boldsymbol{u}}\;\overline{\boldsymbol{u}} + \boldsymbol{\tau}\,, \tag{2.38}$$

where, following the LES terminology [27, 139], the stress tensor

$$\boldsymbol{\tau} = \overline{\boldsymbol{u}\,\boldsymbol{u}} - \overline{\boldsymbol{u}}\;\overline{\boldsymbol{u}} \tag{2.39}$$

is the subfilter-scale stress tensor. Thus, the SF-NSE (2.37) *take the form*

$$\left(\frac{\partial \overline{\boldsymbol{u}}}{\partial t}, \boldsymbol{\phi}\right) + \frac{2}{Re}\left(\mathbb{D}(\overline{\boldsymbol{u}}), \nabla \phi_i\right) + \left(\nabla \cdot \left(\overline{\boldsymbol{u}}\;\overline{\boldsymbol{u}}\right), \boldsymbol{\phi}\right) + \left(\nabla \cdot \boldsymbol{\tau}, \boldsymbol{\phi}\right) = \boldsymbol{0}\,. \tag{2.40}$$

We emphasize that equation (2.40) *can be regarded as the standard weak formulation for the NSE for* $\overline{\boldsymbol{u}}$ *(not* $\boldsymbol{u}$*), plus the extra term* $\left(\nabla \cdot \boldsymbol{\tau}, \boldsymbol{\phi}\right)$*. The closure problem for the SF-NSE* (2.40) *is*

$$\boldsymbol{\tau} = \mathit{function}(\overline{\boldsymbol{u}})\,. \tag{2.41}$$

Equation (2.41) *is the form generally used in the LES literature for the closure problem.*

2.3 Functional ROM Closure

Using a ROM approximation in (2.40), we get another form of the LES-ROM (2.36):

$$\left(\frac{\partial \overline{\boldsymbol{u}}}{\partial t}, \boldsymbol{\varphi}\right) + \frac{2}{Re}\left(\mathbb{D}(\overline{\boldsymbol{u}}), \nabla \boldsymbol{\varphi}_i\right) + \left(\nabla \cdot \left(\overline{\boldsymbol{u}}\;\overline{\boldsymbol{u}}\right), \boldsymbol{\varphi}\right) + \left(\nabla \cdot \boldsymbol{\tau}, \boldsymbol{\varphi}\right) = \boldsymbol{0}\,. \tag{2.42}$$

where $\boldsymbol{\tau}$ is defined in (2.41).

Definition 2.5 (ROM closure problem)**.** *The ROM closure problem is modeling the ROM stress tensor* $\boldsymbol{\tau}$*, i.e., finding a function* $\boldsymbol{f}$ *that satisfies*

$$\nabla \cdot \boldsymbol{\tau} \approx \boldsymbol{f}(\boldsymbol{a}, \boldsymbol{X}^r)\,. \tag{2.43}$$

The ROM closure model is the explicit dependence of the ROM stress tensor $\boldsymbol{\tau}$ *on the ROM coefficients* $\boldsymbol{a}$ *and ROM space* $\boldsymbol{X}^r$ *in equation* (2.41)*.*

In this section, we present standard ROM closure modeling approaches, which are based on phenomenological arguments. Since the analogy between ROM and LES closure modeling is relevant to LES-ROM framework, we first include next a "time capsule" of LES closure modeling. Of course, we only include the LES developments that are relevant to our work; for more details, we refer the reader to specialized texts, e.g., [27, 75, 134, 139].

The idea in LES is straightforward. Instead of approximating the flow variables, the LES models aim at approximating the spatially filtered flow variables, which can be discretized on significantly coarser spatial meshes. However, since the underlying NSE are nonlinear, filtering them yields a system of equations that is not closed. Closing the spatially filtered NSE, which is commonly known as the *closure problem*, represents the main challenge faced by LES.

There are many closure models used in LES. Following Sagaut's terminology [139], these LES closure models could be divided into two categories: (i) functional closure models and (ii) structural closure models. Chronologically, the functional closure models were developed first and the structural closure models were put forth later.

The *functional* LES closure models (chapters 5 and 6 in [139]) follow from phenomenological arguments and aim at modeling the physical effect of the sub-filter scales (i.e., the scales below the spatial filter radius). The main tool used in developing these functional closure models is Kolmogorov's statistical theory of turbulence and the resulting energy cascade, which postulates that the main role of the sub-filter scales is to drain energy out of the LES model. In the functional closure models, this is generally achieved by adding an eddy viscosity to the molecular viscosity of the system. Probably the most popular example in this class is the Smagorinsky model [146].

The second type of closure models is *structural LES* closure models (chapter 7 in [139]). These closure models are generally derived through mathematical rather than phenomenological arguments, e.g., formal series expansions. One of the most popular models in this class is the approximate deconvolution (AD) model [148, 150, 151], which uses the deconvolution approaches developed in the image processing and inverse problems communities to recover the original signal from a blurred filtered signal.

The ROM closure model development was similar to the LES closure modeling development. Just as in LES, the first ROM closure models have been of the functional type. These functional ROM closure models have generally used some sort of stabilization procedure to model the effect of the discarded POD modes. Probably the first functional ROM closure model has been the mixing length model pioneered by Lumley and his group [12]. This model was followed by more accurate ROM closure models, such as the Smagorinsky model [119, 157, 161], the variational multiscale model [26, 162] and the dynamic subgrid-scale model [162]. As already mentioned, a physical motivation for the approach used to develop these functional ROM closure models is the concept of energy cascade [57]. We emphasize that, just as in the LES case, these functional ROM closure models do not usually employ an explicit spatial filter; instead, they use phenomenological arguments to model the effect of the discarded POD modes. In what follows, we list some of the functional ROM closure models that have been studied over the years. For more references, see [12, 13, 26, 54, 55, 67, 68, 121, 131, 141, 144].

The first ROM closure model was the *mixing length ROM (ML-ROM)* introduced in [12]. This is an eddy viscosity ROM closure model which increases the viscosity coefficient ν by

$\nu_{ML} = \alpha\nu_T = \alpha U_{ML} L_{ML}$, where U_{ML} and L_{ML} are characteristic velocity and length scales, α is a non-dimensional constant parameter which characterizes the amount of energy being dissipated.

The Smagorinksy ROM is an improvement of the ML-ROM. It is obtained by replacing the constant ν_{ML} with a variable turbulent viscosity defined as, $\nu_S = 2(C_S\delta)^2\|\mathbb{D}(\boldsymbol{u}_r)\|$, where C_S is the Smagorinsky constant, δ is the filter length scale, and $\|\mathbb{D}(\boldsymbol{u}_r)\| = \|(\nabla\boldsymbol{u}_r + \nabla\boldsymbol{u}_r^T)/2\|$ is the Frobenius norm of the deformation tensor. This model is more accurate than the ML-ROM because it recomputes the coefficient at every time step, whereas the ML-ROM uses a constant eddy viscosity.

The variational multiscale ROM (VMS-ROM) is based on the principle of locality of energy transfer [26, 89, 162]. It utilizes the ansatz that energy is transferred mainly between neighboring scales. We divide the ROM space $\boldsymbol{X}^r$ into the large resolved POD modes $\boldsymbol{X}_L^r$ (large structures) and small resolved modes $\boldsymbol{X}_S^r$ (small structures): $\boldsymbol{X}_L^r = span\{\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, ..., \boldsymbol{\varphi}_R\}$, $\boldsymbol{X}_S^r = span\{\boldsymbol{\varphi}_{R+1}, \boldsymbol{\varphi}_{R+2}, ..., \boldsymbol{\varphi}_r\}$. We also decompose the flow variable $\boldsymbol{u}_r$ as, $\boldsymbol{u}_r = \boldsymbol{u}_r^L + \boldsymbol{u}_r^S$, $\boldsymbol{u}_r^L = \boldsymbol{U} + \sum_{j=1}^{R} a_j\boldsymbol{\varphi}_j$, $\boldsymbol{u}_r^S = \sum_{j=R+1}^{r} a_j\boldsymbol{\varphi}_j$. Finally, we get the VMS-ROM by substituting the above decomposition into the Galerkin projection and using a Smagorinsky ROM closure model only for $\boldsymbol{u}_r^S$. The resulting VMS-ROM is different from the Smagorinsky ROM because it acts only on the small resolved scales.

The dynamic subgrid-scale ROM (DS-ROM) was proposed in [162]. It is based on the state-of-the-art LES dynamic subgrid-scale model [73]. The DS-ROM employs a Smagorinsky term as closure model. However, the Smagorinsky "constant" is recomputed at every point in space and at every time instance. This process utilizes the Germano identity [73], which requires two explicit spatial filters. In [162] it was shown that the DS-ROM outperforms the ML-ROM, Smagorinsky-ROM, and VMS-ROM in the numerical simulation of a 3D flow past a circular cylinder at $Re = 1000$.

Chapter 3

Approximate Deconvolution Reduced Order Model (AD-ROM)

Within the LES-ROM framework introduced in Chapter 2, an approximate deconvolution (AD) approach is used to solve the ROM closure problem and develop a new AD-ROM. To our knowledge, the AD-ROM is the first structural ROM closure problem. To build the AD-ROM, we start with a given approximation of the filtered ROM variables, and then we use AD to obtain an approximation of the original unfiltered ROM variables and solve the ROM closure problem. Since the AD problem is notoriously ill-posed, we use a regularization method from image processing and inverse problems to obtain stable AD approximations. The AD-ROM is tested in the numerical simulation of the three-dimensional flow past a circular cylinder at a Reynolds number $Re = 1000$.

3.1 Introduction

Numerous functional ROM closure modeling approaches have been proposed to address the inaccuracy (and numerical instability) of the standard G-ROM (see, e.g., [118, 133, 162] and references therein). Functional ROM closure models do not employ an explicit spatial filter. In this chapter, we use the explicit ROM spatial filter introduced in Section 2.2 to develop a structural ROM closure model. To our knowledge, structural ROM closure models have not been proposed yet. Again, this is not surprising, since in LES the structural closure models were developed much later than the functional closure models. We believe that the main reason why structural ROM closure models have not been proposed is that the concept of *explicit spatial filtering* (which is central in the development of structural LES closure models) has not been exploited yet in the ROM field.

Most ROM closure models have generally used some sort of stabilization procedure, e.g., [7, 12, 13, 16, 26, 110, 121, 125, 131, 133, 162]. A physical motivation for this popular approach is given in [57], where it is shown that the concept of energy cascade is also valid in a POD setting. Thus, the role of the ROM closure model is to drain energy out of the system.

In this chapter, we use a fundamentally different, novel ROM closure modeling strategy that centers around the concept of *approximate deconvolution (AD)*. In LES, the AD models

have been pioneered in a series of papers by Adams and Stolz [148, 149, 150, 151]. These AD models are based on the *deconvolution* idea, which is central in *image processing* and *inverse problems* [28, 81, 158]: Given an approximation of the filtered input signal, find an approximation of the input itself. In LES, the deconvolution is used as follows: Given the available approximation of the filtered velocity, the deconvolution is used to obtain an approximation of the unfiltered velocity, which is then used to solve the LES closure problem. We use the LES-ROM framework developed in Section 2.2.2 to construct a novel AD-ROM. In Section 3.2, we present the deconvolution procedure. In Section 3.3, we use this deconvolution to develop the new AD-ROM.

3.2 Approximate Deconvolution

For clarity of presentation, let us denote with G the explicit ROM spatial filter used to develop the LES-ROM framework in Section 2.2.2. We assume that an approximation of the filtered velocity

$$\overline{\boldsymbol{u}} := G\,\boldsymbol{u} \tag{3.1}$$

is available. The goal in the ROM deconvolution problem is to find an approximation of the original flow variable, $\boldsymbol{u}$, so that the ROM closure problem (2.34) can be solved.

Since G is invertible, at first glance, one just has to use an *exact deconvolution (ED)*, i.e., employ the inverse of the filter G in (3.1) to solve the deconvolution problem:

$$\boldsymbol{u}^{ED} = G^{-1}\,\overline{\boldsymbol{u}} = \boldsymbol{u}\,. \tag{3.2}$$

It is well-known in the inverse problem community that the ED is generally a bad idea because the inverse problem (3.1) is *ill-posed*: small changes in the data can lead to large changes in the solution [28, 107, 158]. Indeed, inverting the operator G in (3.2) results in division by small coefficients of the high-frequency components of the operator G. Thus, any changes in the input (3.1) translate into large unphysical oscillations in the output (3.2) [28, 158]. The ED (3.2) was carefully investigated in Section 5.2 of [167], where it was shown that, as expected, for input signals ($\boldsymbol{u}$) of different types, adding noise to the filtered signal ($\overline{\boldsymbol{u}}$) and then using the ED (3.2) results in inaccurate approximations ($\boldsymbol{u}^{ED}$), usually in the form of numerical oscillations.

To avoid the numerical oscillations in the ED (3.2), we use one of the *regularized deconvolution* methods developed in the image processing and inverse problems communities [28, 81, 158]. Following the LES terminology, we call this method *approximate deconvolution (AD)*. Since this work is the first investigation of the new AD-ROM, we employ one of the simplest regularized deconvolutions used in the image processing and inverse problems communities, the *Lavrentiev regularization*:

$$\boldsymbol{u}^{AD} = \Big(G + \mu\, I\Big)^{-1}\,\overline{\boldsymbol{u}}\,, \tag{3.3}$$

where I is the identity operator and μ is a regularization parameter.

The AD (3.3) was carefully investigated in Section 5.4 in [167], where it was shown that it performs much better than the ED (3.2): For input signals of different types, noise was added

to the filtered signal ($\overline{\boldsymbol{u}}$), and the ED and AD were compared. The AD (3.3) produced much better results than the ED (3.2): The latter increased the amplitude of the noise added to the filtered input signal, which, in turn, yielded spurious numerical oscillations. The former, on the other hand, significantly decreased the amplitude of the noise and yielded relatively accurate results.

Remark 3.1 (AD vs ED)**.** *We emphasize that these results are not surprising at all. Indeed, they reflect the reality in the inverse problems and image processing communities, where it is well known that the ED produces inaccurate results, and the AD should be used instead. However, both the AD and ED were carefully investigated in [167], since, to our knowledge, this was the first investigation of deconvolution in the ROM field.*

3.3 Approximate Deconvolution ROM (AD-ROM)

The AD approximation (3.3) is used to close the LES model (2.35): $\forall\, i = 1, \ldots, r$,

$$\left(\frac{\partial \overline{\boldsymbol{u}}}{\partial t}, \boldsymbol{\varphi}_i\right) + \frac{2}{Re}\left(\mathbb{D}(\overline{\boldsymbol{u}}), \nabla \boldsymbol{\varphi}_i\right) + \left(\overline{\boldsymbol{u}^{AD} \cdot \nabla \boldsymbol{u}^{AD}}, \boldsymbol{\varphi}_i\right) = \boldsymbol{0}\,. \tag{3.4}$$

At first glance, to obtain an LES-ROM, one would simply replace the filtered velocity $\overline{\boldsymbol{u}}$ in (3.4) with a ROM approximation

$$\boldsymbol{W}_r = \boldsymbol{U} + \sum_{j=1}^{r} a_j\, \boldsymbol{\varphi}_j = \boldsymbol{U} + \boldsymbol{w}_r\,. \tag{3.5}$$

Remark 3.2. *Note that the coefficients a_j in the $\boldsymbol{w}_r$ expansion* (3.5) *are different from the coefficients a_j used in the $\boldsymbol{u}_r$ expansion* (2.4)*. We use the same notation, however, to ensure consistency with the traditional ROM notation.*

We emphasize that approximation (3.5) is not appropriate. Indeed, since the POD basis functions were obtained from the numerical discretization of the *unfiltered* NSE, they can only approximate the *unfiltered* flow velocity, $\boldsymbol{u}$:

$$\boldsymbol{u} \approx \boldsymbol{U}_r = \boldsymbol{U} + \boldsymbol{u}_r\,. \tag{3.6}$$

Thus, the *filtered* flow velocity in (3.4) should be approximated as

$$\overline{\boldsymbol{u}} \approx \overline{\boldsymbol{U}_r} \overset{(3.6)}{=} \overline{\boldsymbol{U}} + \overline{\boldsymbol{u}_r}\,. \tag{3.7}$$

Furthermore, to be able to approximate $\overline{\boldsymbol{u}}$ in the ROM space $\mathbf{X}^r$, we make the following ansatz:

$$\overline{\boldsymbol{u}_r} \approx \boldsymbol{w}_r = \sum_{j=1}^{r} a_j\, \boldsymbol{\varphi}_j\,. \tag{3.8}$$

Thus, the filtered velocity $\overline{\boldsymbol{u}}$ in (3.4) should be replaced with

$$\boldsymbol{W}_r = \overline{\boldsymbol{U}} + \overline{\boldsymbol{u}_r} \overset{(3.8)}{\approx} \overline{\boldsymbol{U}} + \boldsymbol{w}_r\,. \tag{3.9}$$

We emphasize that, although approximations (3.5) and (3.9) are similar in that they both use ansatz (3.8), they are different in one important aspect: Approximation (3.5) *assumes* that the *filtered* centering trajectory $\overline{\boldsymbol{U}}$ can be approximated by the unfiltered centering trajectory $\boldsymbol{U}$ and the POD basis, whereas approximation (3.9) does not make this assumption.

Next, we present the ROM approximation of $\boldsymbol{u}^{AD}$ in (3.4):

$$\begin{aligned}\boldsymbol{u}^{AD} &\approx G^{-1}\,\overline{\boldsymbol{u}} \overset{(3.6)}{\approx} G^{-1}\,\overline{\boldsymbol{U}_r} \overset{(3.8)}{\approx} G^{-1}\left(\overline{\boldsymbol{U}}+\boldsymbol{w}_r\right) = \boldsymbol{U}+G^{-1}\,\boldsymbol{w}_r \\ &\overset{(3.3)}{\approx} \boldsymbol{U}+\left(G+\mu\,I\right)^{-1}\boldsymbol{w}_r\,. \\ &= \boldsymbol{U}+\boldsymbol{w}_r^{AD}\,, \end{aligned} \tag{3.10}$$

where $\boldsymbol{w}_r^{AD}$ is a ROM approximation of the unfiltered flow velocity.

Using the ROM approximations (3.9) and (3.10) in the LES model (3.4), we obtain the *approximate deconvolution ROM (AD-ROM)*:

$$\begin{aligned}\left(\frac{\partial(\overline{\boldsymbol{U}}+\boldsymbol{w}_r)}{\partial t},\boldsymbol{\varphi}_i\right) &+ \frac{2}{Re}\left(\mathbb{D}(\overline{\boldsymbol{U}}+\boldsymbol{w}_r),\nabla\boldsymbol{\varphi}_i\right) \\ &+ \left(\overline{(\boldsymbol{U}+\boldsymbol{w}_r^{AD})\cdot\nabla(\boldsymbol{U}+\boldsymbol{w}_r^{AD})},\boldsymbol{\varphi}_i\right)=\boldsymbol{0}\,. \end{aligned} \tag{3.11}$$

Furthermore, since the C-DF is self-adjoint (see Lemma 2.2), the AD-ROM (3.11) can be written is the following equivalent form:

$$\begin{aligned}&\left(\cancelto{0}{\frac{\partial\overline{\boldsymbol{U}}}{\partial t}}+\frac{\partial\boldsymbol{w}_r}{\partial t},\boldsymbol{\varphi}_i\right)+\frac{2}{Re}\left(\mathbb{D}(\boldsymbol{U}),\nabla\overline{\boldsymbol{\varphi}_i}\right)+\frac{2}{Re}\left(\mathbb{D}(\boldsymbol{w}_r),\nabla\boldsymbol{\varphi}_i\right) \\ &+\left(\boldsymbol{U}\cdot\nabla\boldsymbol{U},\overline{\boldsymbol{\varphi}_i}\right)+\left(\boldsymbol{w}_r^{AD}\cdot\nabla\boldsymbol{U},\overline{\boldsymbol{\varphi}_i}\right)+\left(\boldsymbol{U}\cdot\nabla\boldsymbol{w}_r^{AD},\overline{\boldsymbol{\varphi}_i}\right) \\ &+\left(\boldsymbol{w}_r^{AD}\cdot\nabla\boldsymbol{w}_r^{AD},\overline{\boldsymbol{\varphi}_i}\right)=\boldsymbol{0}\,. \end{aligned} \tag{3.12}$$

We note that the alternative AD-ROM formulation (3.12) is better suited for discretization than the standard AD-ROM formulation (3.11), since it avoids the spatial filtering of the vectors and tensors involving the centering trajectory ($\boldsymbol{U}$), which can be challenging in a ROM setting. Thus, in the numerical investigation in Section 3.5, we will use the alternative AD-ROM formulation (3.12).

Remark 3.3. *At first glance, the approach used in the derivation of the alternative formulation of the AD-ROM in* (3.12) *(i.e., using the self-adjointness of the C-DF) is reminiscent of other approaches used in LES. For example, using test functions that are multiplied by the inverse of the C-DF and then using the self-adjointness of the C-DF was essential in the proof of the well-posedness of the Rational LES model (see discussion on page 161 in [27]). The same approach was later used to develop a new type of AD-LES models in [70, 95, 113].*

We emphasize, however, that there is a fundamental difference between the AD-ROM (3.12) *and the AD-LES in [70, 95, 113]: To develop the alternative AD-ROM formulation* (3.12)*, we first multiplied by the standard test function and then we used the self-adjointness of the C-DF to move the filtering operator from the terms involving the centering trajectory to the test function. We note that the filtered test functions that appear in* (3.12) *do not constitute a challenge in our ROM setting, since they can be precomputed and stored. However, for the discretization of the AD-LES models in [70, 95, 113] with standard numerical methods (e.g., FE), storing all the filtered basis functions is not possible in realistic settings with a large number of degrees of freedom. Thus, for the AD-LES models in [70, 95, 113], one has first to use test functions that are multiplied by the inverse of the C-DF and only then use the self-adjointness of the C-DF. The price paid, however, is that fourth-order differential operators need to be discretized in these AD-LES models [70, 95].*

3.4 Spatial Filtering for the AD-ROM

To use the AD-ROM (3.12) in actual numerical simulations, the continuous C-DF (2.20)–(2.21) needs to be replaced by one of its two discrete variants: the FE-DF (2.26)–(2.27) or the ROM-DF (2.29)–(2.30).

3.4.1 ROM-DF Spatial Filtering

With this choice, the AD-ROM (3.12) can be written as the following dynamical system:

$$\dot{\boldsymbol{a}} = \boldsymbol{b} + \boldsymbol{A}\,\boldsymbol{a} + \boldsymbol{A}^{AD}\,\boldsymbol{a}^{AD} + (\boldsymbol{a}^{AD})^{\top}\,\boldsymbol{B}^{AD}\,\boldsymbol{a}^{AD}\,, \tag{3.13}$$

which can be written componentwise as follows: For all $i = 1, \ldots, r$,

$$\begin{aligned} \dot{a}_i(t) &= b_i + \sum_{m=1}^{r} A_{im}\, a_m(t) + \sum_{m=1}^{r} A_{im}^{AD}\, a_m^{AD}(t) \\ &+ \sum_{m=1}^{r}\sum_{n=1}^{r} B_{imn}^{AD}\, a_m^{AD}(t)\, a_n^{AD}(t)\,, \end{aligned} \tag{3.14}$$

where

$$b_i = -\Big(\mathbf{U}\cdot\nabla\mathbf{U}, \overline{\boldsymbol{\varphi}_i}^{r}\Big) - \frac{2}{Re}\left(\frac{\nabla\mathbf{U} + \nabla\mathbf{U}^{\top}}{2}, \nabla\overline{\boldsymbol{\varphi}_i}^{r}\right), \tag{3.15}$$

$$A_{im} = -\frac{2}{Re}\left(\frac{\nabla\boldsymbol{\varphi}_m + \nabla\boldsymbol{\varphi}_m{}^{\top}}{2}, \nabla\boldsymbol{\varphi}_i\right), \tag{3.16}$$

$$A_{im}^{AD} = -\Big(\mathbf{U}\cdot\nabla\boldsymbol{\varphi}_m, \overline{\boldsymbol{\varphi}_i}^{r}\Big) - \Big(\boldsymbol{\varphi}_m\cdot\nabla\mathbf{U}, \overline{\boldsymbol{\varphi}_i}^{r}\Big), \tag{3.17}$$

$$B_{imn}^{AD} = -\Big(\boldsymbol{\varphi}_m\cdot\nabla\boldsymbol{\varphi}_n, \overline{\boldsymbol{\varphi}_i}^{r}\Big), \tag{3.18}$$

and the vector $\boldsymbol{a}^{AD} = \{a_m^{AD}\}_{m=1}^{r}$ in (3.14) is the solution of the following linear system:

$$\Big(\big(1+\mu\big)\,\boldsymbol{M}_r + \delta^2\,\boldsymbol{S}_r\Big)\,\boldsymbol{a}^{AD} = \Big(\boldsymbol{M}_r + \delta^2\,\boldsymbol{S}_r\Big)\,\boldsymbol{a}\,. \tag{3.19}$$

3.4.2 FE-DF Spatial Filtering

To our knowledge, the only investigation of the FE-DF in a ROM setting has been carried out in [164]. In this section, we outline only the FE-DF features that are relevant to the AD-ROM (3.12). For more details, we refer the reader to [164].

ROM filtering with the FE-DF relies on the fact that the ROM space ($\boldsymbol{X}^r$) is a subspace of the FE space ($\boldsymbol{X}^h$). Indeed, each POD basis function $\boldsymbol{\varphi}_i$, $i = 1, \ldots, r$ has a unique FE representation:

$$\boldsymbol{\varphi}_i = \sum_{j=1}^{N} c_{ji}\, \boldsymbol{h}_j\,, \tag{3.20}$$

where $\{\boldsymbol{h}_j\}_{j=1}^N$ are the FE basis functions. (We note in passing that it is precisely representation (3.20) that allows the offline stage assembly of the vectors, matrices and tensors in standard ROMs.)

The FE-DF is used in the AD-ROM (3.12) in two instances: To compute $\overline{\boldsymbol{\varphi}_i}^h$ and $\boldsymbol{w}_r^{AD}$. Next, we briefly outline the approach used to compute these terms.

To compute $\overline{\boldsymbol{\varphi}_i}^h$, we use the representation (3.20) for both $\boldsymbol{\varphi}_i$ and $\overline{\boldsymbol{\varphi}_i}^h$:

$$\boldsymbol{\varphi}_i = \sum_{j=1}^{N} c_{ji}\, \boldsymbol{h}_j \qquad \text{and} \qquad \overline{\boldsymbol{\varphi}_i}^h = \sum_{j=1}^{N} \overline{c}_{ji}\, \boldsymbol{h}_j\,. \tag{3.21}$$

Thus, the equation $\overline{\boldsymbol{\varphi}_i}^h \overset{(2.26)}{=} (I - \delta^2\, \Delta)^{-1} \boldsymbol{\varphi}_i$ yields the following linear systems: $\forall\, i = 1, \ldots, r$,

$$\left(\boldsymbol{M}_h + \delta^2\, \boldsymbol{S}_h\right) \overline{\boldsymbol{c}}_i = \boldsymbol{M}_h\, \boldsymbol{c}_i\,, \tag{3.22}$$

where $\boldsymbol{M}_h$ and $\boldsymbol{S}_h$ are the FE mass and stiffness matrices defined in Section 2.2.1, and $\overline{\boldsymbol{c}}_i = \{\overline{c}_{ji}\}_{j=1}^N$ and $\boldsymbol{c}_i = \{c_{ji}\}_{j=1}^N$ are the vectors of coefficients in (3.21), which need to be computed and stored in the offline stage to be used in the AD-ROM in the online stage.

To compute $\boldsymbol{w}_r^{AD}$, we need to solve the following equation:

$$\boldsymbol{w}_r^{AD} \overset{(2.26),(3.3)}{=} \left((I - \delta^2\, \Delta)^{-1} + \mu\, I \right)^{-1} \boldsymbol{w}_r\,, \tag{3.23}$$

which is equivalent to

$$\begin{aligned} \boldsymbol{w}_r^{AD} + \mu\, (I - \delta^2\, \Delta)\, \boldsymbol{w}_r^{AD} &= (I - \delta^2\, \Delta)\, \boldsymbol{w}_r \\ &\overset{(3.8),(3.20)}{=} \sum_{j=1}^{r} a_j \sum_{i=1}^{N} c_{ij}\, (I - \delta^2\, \Delta)\, \boldsymbol{h}_i\,. \end{aligned} \tag{3.24}$$

We emphasize that (3.24) needs to be solved in $\boldsymbol{X}^h$, since the FE-DF is defined in $\boldsymbol{X}^h$. There are two natural approaches to solve (3.24):

(i) In the first approach, we simply let

$$\boldsymbol{w}_r^{AD} = \sum_{i=1}^{N} \alpha_i\, \boldsymbol{h}_i \tag{3.25}$$

in (3.24), which yields the following $N \times N$ linear system:

$$\left((1+\mu)\,\boldsymbol{M}_h + \mu\,\delta^2\,\boldsymbol{S}_h\right)\boldsymbol{\alpha} = \sum_{j=1}^{r} a_j \left((\boldsymbol{M}_h + \delta^2\,\boldsymbol{S}_h)\,\boldsymbol{c}_j\right), \tag{3.26}$$

where $\boldsymbol{\alpha} = \{\alpha_j\}_{j=1}^N$ is the vector of coefficients in (3.25).

(ii) In the second approach for solving (3.24), we let

$$\boldsymbol{w}_r^{AD} = \sum_{j=1}^{r} a_j \sum_{i=1}^{N} c_{ij}^{AD} \boldsymbol{h}_i = \sum_{j=1}^{r} a_j\,\boldsymbol{\varphi}_j^{AD}, \tag{3.27}$$

where $\boldsymbol{\varphi}_j^{AD}$ is defined as

$$\boldsymbol{\varphi}_j^{AD} = \sum_{i=1}^{N} c_{ij}^{AD} \boldsymbol{h}_i\,. \tag{3.28}$$

We emphasize that $\boldsymbol{\varphi}_j^{AD}$ belongs to the FE space $\mathbf{X}^h$, but does not necessarily belong to the ROM space $\mathbf{X}^r$. To solve (3.24), we simply solve the following r linear systems of dimension $N \times N$ in the offline stage:

$$\left((1+\mu)\,\boldsymbol{M}_h + \mu\,\delta^2\,\boldsymbol{S}_h\right)\boldsymbol{c}_j^{AD} = \left(\boldsymbol{M}_h + \delta^2\,\boldsymbol{S}_h\right)\boldsymbol{c}_j\,, \qquad j = 1, \ldots, r\,, \tag{3.29}$$

where $\boldsymbol{c}_j^{AD} = \{c_{ij}^{AD}\}_{i=1}^N$ is the vector of coefficients in (3.27).

Comparing approaches (i) and (ii) in terms of computational efficiency, we draw the following conclusions: In approach (i), there is no extra storage required, but the $N \times N$ linear system (3.26) needs to be solved at each time step in the online stage. In approach (ii), the r linear systems (3.29) are solved only once, in the offline stage, after which the r vectors $\boldsymbol{c}_j^{AD}$, $j = 1, \ldots, r$ (corresponding to $\boldsymbol{\varphi}_j^{AD}$ in (3.28)) are used to assemble and store the $r \times r$ matrices and $r \times r \times r$ tensor corresponding to the nonlinearity in the AD-ROM (3.12). Since the extra storage requirements in approach (ii) are relatively low for the small r that we use in this numerical test, in Section 3.5 we utilize this approach instead of approach (i), which has a significantly higher CPU cost.

Using approach (ii), the AD-ROM (3.12) can be written as the following dynamical system:

$$\dot{\boldsymbol{a}} = \boldsymbol{b} + \left(\boldsymbol{A} + \boldsymbol{A}^{AD}\right)\boldsymbol{a} + \boldsymbol{a}^\top \boldsymbol{B}^{AD}\,\boldsymbol{a}\,, \tag{3.30}$$

which can be written componentwise as follows: For all $i = 1, \ldots, r$,

$$\dot{a}_i(t) = b_i + \sum_{m=1}^{r} \left(A_{im} + A_{im}^{AD}\right) a_m(t) + \sum_{m=1}^{r}\sum_{n=1}^{r} B_{imn}^{AD}\,a_m(t)\,a_n(t), \tag{3.31}$$

where

$$b_i = -\Big(\mathbf{U}\cdot\nabla\mathbf{U}, \overline{\boldsymbol{\varphi}_i}^h\Big) - \frac{2}{Re}\left(\frac{\nabla\mathbf{U}+\nabla\mathbf{U}^\top}{2}, \nabla\overline{\boldsymbol{\varphi}_i}^h\right), \tag{3.32}$$

$$A_{im} = -\frac{2}{Re}\left(\frac{\nabla\boldsymbol{\varphi}_m+\nabla\boldsymbol{\varphi}_m{}^\top}{2}, \nabla\boldsymbol{\varphi}_i\right), \tag{3.33}$$

$$A^{AD}_{im} = -\Big(\mathbf{U}\cdot\nabla\boldsymbol{\varphi}^{AD}_m, \overline{\boldsymbol{\varphi}_i}^h\Big) - \Big(\boldsymbol{\varphi}^{AD}_m\cdot\nabla\mathbf{U}, \overline{\boldsymbol{\varphi}_i}^h\Big), \tag{3.34}$$

$$B^{AD}_{imn} = -\Big(\boldsymbol{\varphi}^{AD}_m\cdot\nabla\boldsymbol{\varphi}^{AD}_n, \overline{\boldsymbol{\varphi}_i}^h\Big). \tag{3.35}$$

3.5 Numerical Tests

Goals The main goal of this section is to assess the performance of the AD-ROM (3.12) in the numerical simulation of a 3D flow past a circular cylinder at Reynolds number $Re = 1000$. Since the AD-ROM is an LES-ROM, we follow the standard approach used in LES model assessment, which we briefly describe next (see, e.g., [27, 139] for details). To investigate a new LES model, the following models are generally tested on the same coarse mesh: (i) the new LES model; (ii) the under-resolved solver (i.e., the NSE code without an LES model); (iii) the filtered DNS data (which serves as a benchmark); and (iv) the current state-of-the-art LES model. The new LES model is expected to perform significantly better than the under-resolved solver and to show some improvement over the current state-of-the-art LES model.

To assess the performance of the new AD-ROM, we follow the standard LES approach described above. Thus, we also use the spatially filtered DNS results as a benchmark. Furthermore, we also include G-ROM (2.5) results as a ROM counterpart of the under-resolved solver used in the LES assessment. Finally, we also make a brief qualitative comparison of the new AD-ROM with the eddy viscosity ROMs (EV-ROMs) [12, 162] (which, to our knowledge, are the current state-of-the-art LES-ROMs) and the regularized ROMs (Reg-ROMs) [165]. Just as in the standard LES model assessment, we expect the new AD-ROM to perform significantly better than the G-ROM and show some improvement over the EV-ROMs and Reg-ROMs.

Computational Domain In this section, the x-axis represents the streamwise direction, the y-axis represents the normal direction, and the z-axis represents the spanwise direction. Furthermore, u denotes the streamwise velocity component (associated with the x-axis), v denotes the normal velocity component (associated with the y-axis), and w denotes the spanwise velocity component (associated with the z-axis).

The computational domain is cylindrical and is centered around the z-axis, i.e., along the spanwise direction. The cylinder of diameter D is also centered around the z-axis. The diameter of the cylindrical computational domain is $15D$ and its spanwise length is $2D$. The free-stream flow is in the positive x-direction. The same computational domain was used in [161, 162] (see Fig. 9 and Appendix A in [161]).

Criteria The qualitative behavior of the ROMs is judged according to the following six criteria [162]: (i) the kinetic energy spectrum; (ii) the mean velocity; (iii) the Reynolds stresses; (iv) the root mean square (rms) values of the velocity fluctuations; (v) the time evolution of the POD coefficients; and (vi) the Strouhal number. The first four criteria are statistics that measure the temporal and spatial average behavior of the ROMs, whereas the fifth criterion measures the instantaneous behavior of the ROMs. We also include a computational efficiency assessment for all the ROMs. Next, we briefly present these criteria.

For all statistics, we use the averaging operator $\langle\cdot\rangle = \langle\cdot\rangle_{tyz}$, which consists of averaging in time (over the time interval $[0, 300]$) and in the y- and z- directions. Specifically, to compute $\langle q\rangle_{tyz}$ for a given quantity q, for each fixed point x, we have

$$\langle q\rangle_{tyz}(x) = \frac{1}{T\,L_y\,L_z}\sum_{t,y,z} q(x,y,z,t)\,, \tag{3.36}$$

where T is the total time length (i.e., $T = 300$), L_y is the dimension of the computational domain in the y-direction, and L_z is the dimension of the computational domain in the z-direction. Since the numerical results with spatial averaging in the xz-direction are qualitatively similar to those with spatial averaging in the yz-direction, we do not include them here.

Energy Spectrum: All energy spectra are calculated from the average kinetic energy of the nodes in the cube with side 0.1 centered at the probe $(0.9992, 0.3575, 1.0625)$.

Mean Velocity Components: The following mean velocity components are plotted: $\langle u\rangle$ (the mean streamwise velocity), $\langle v\rangle$ (the mean normal velocity), and $\langle w\rangle$ (the mean spanwise velocity).

Reynolds Stresses: The following Reynolds stresses are plotted: $\langle u - \langle u\rangle, v - \langle v\rangle\rangle$ (the xy-component of the Reynolds stress), $\langle u - \langle u\rangle, w - \langle w\rangle\rangle$ (the xz-component of the Reynolds stress), and $\langle v - \langle v\rangle, w - \langle w\rangle\rangle$ (the yz-component of the Reynolds stress).

RMS Values of Velocity Fluctuations: The following rms values of velocity fluctuations are plotted: $\langle u\rangle_{rms} = \langle u - \langle u\rangle, u - \langle u\rangle\rangle$ (the rms of the streamwise velocity fluctuations), $\langle v\rangle_{rms} = \langle v - \langle v\rangle, v - \langle v\rangle\rangle$ (the rms of the normal velocity fluctuations), and $\langle w\rangle_{rms} = \langle w - \langle w\rangle, w - \langle w\rangle\rangle$ (the rms of the spanwise velocity fluctuations).

Strouhal Number: The Strouhal number (St) is computed as [2, 4]

$$St = \frac{f\,D}{U_\infty}\,, \tag{3.37}$$

where f is the shedding frequency, U_∞ is the freestream velocity (i.e., the velocity at the inlet), and D is the diameter of the cylinder. Since we have limited the scope of this Chapter to just velocity ROMs, we do not consider a ROM representation of the pressure field. Hence we do not calculate the lift and drag coefficient spectra. Thus, we compute the shedding frequency f as the frequency corresponding to the first spike in the energy spectrum (see discussion on page 410 in [98]).

Numerical Methods We investigate all ROMs in the numerical simulation of 3D flow past a circular cylinder at Re = 1000. The Reynolds number is computed using the diameter (D)

of the cylinder as the length scale and the freestream velocity (U_∞) as the velocity scale. A parallel CFD solver is employed on the time interval $[0, 300]$ to generate the DNS data. Details on the numerical discretization are presented in [2, 3, 4].

We obtain the POD basis by collecting 1000 snapshots of the DNS velocity field (u, v, w) over the time interval $[0, 75]$ and applying the method of snapshots developed in [145]. These POD modes are then interpolated onto a structured quadratic FE triangulation with nodes coinciding with the nodes used in the original DNS finite difference discretization. All the ROMs use the first $r = 6$ POD modes, which capture 84% of the energy of the velocity fluctuations. As temporal discretization, all the ROMs employ the explicit Euler method with a time step $\Delta t = 7.5 \times 10^{-4}$. (We note that decreasing the time step by a factor of 10 does not significantly change the numerical results.) As spatial discretization, all the ROMs use piecewise quadratic Lagrange FEs. Finally, all the ROMs are investigated on the time interval $[0, 300]$.

Model Parameters In Section 3.4, we discussed two possible discretizations of the C-DF: the ROM-DF and the FE-DF. Since in our preliminary numerical tests the FE-DF yielded more accurate results than the ROM-DF, in this section we use the FE-DF and the resulting AD-ROM formulation in equations (3.30)–(3.35).

One of the most important parameters of the AD-ROM is the radius of the ROM spatial filter, δ. In our numerical investigation, we tried only three δ values: $\delta = 0.03, \delta = 0.3$ and $\delta = 0.4$. The low value (i.e., $\delta = 0.03$) yielded inaccurate results. The high values (i.e., $\delta = 0.3$ and $\delta = 0.4$) yielded similar results, significantly more accurate than the results for $\delta = 0.03$. Thus, in what follows, we present results for $\delta = 0.3$. We emphasize, however, that the value $\delta = 0.3$ does not represent an optimal value; it is just the best value of the very few we tried. To find the optimal value, numerous δ values (and different test problems) should be thoroughly investigated. We plan to carry out this investigation in a future study (see the discussion in Chapter 7).

Another important parameter of the AD-ROM is the regularization parameter μ used in the Lavrentiev regularization of the AD procedure. In the image processing and inverse problems communities, numerous approaches are proposed to determine the regularization parameters [28, 81, 158]. In our numerical investigation, for the fixed $\delta = 0.3$ value, we choose the μ value that ensures that the time average of the L^2-norm of the AD-ROM solution is as close as possible to the time average of the L^2-norm of the filtered DNS data. This approach yields the following optimal value: $\mu = 0.0285$.

3.5.1 Numerical Results

In Fig. 3.1, we plot the energy spectra of the AD-ROM and, for comparison purposes, of the G-ROM. The two energy spectra are compared with the spatially filtered DNS energy spectrum. The energy spectrum of the AD-ROM is significantly more accurate (i.e., closer to the filtered DNS energy spectrum) than the energy spectrum of the G-ROM. The latter overestimates the energy spectrum of the filtered DNS.

In Fig. 3.2, we plot the mean velocity components for different x values. Figure 3.2 yields the following conclusions: Both the AD-ROM and the G-ROM yield an accurate approximation of the streamwise velocity. The AD-ROM yields an accurate approximation

of the mean normal velocity, but the G-ROM does not. Finally, both the AD-ROM and the G-ROM yield inaccurate approximations of the mean spanwise velocity.

In Fig. 3.3, we plot the Reynolds stresses for different x values. For all the Reynolds stresses, the AD-ROM yields accurate approximations, whereas the G-ROM yields inaccurate approximations.

In Fig. 3.4, we plot the rms values for different x values. For all the rms values, the AD-ROM yields accurate approximations, whereas the G-ROM yields inaccurate approximations.

The time evolutions of the ROM's POD basis coefficients a_1, a_2 and a_6 on the entire time interval $[0, 300]$ are shown in Fig. 3.5. We note that the other POD coefficients have a similar behavior. Thus, for clarity of exposition, we include only a_1, a_2 and a_6. The G-ROM results are clearly inaccurate, since the coefficients' magnitudes are several times larger than that of the filtered DNS data. The AD-ROM yields significantly more accurate results for all the coefficients.

Table 3.1: Strouhal numbers predicted by ROMs.

	DNS	G-ROM	AD-ROM
St	0.2083	–	0.1888

The Strouhal numbers predicted by the AD-ROM, G-ROM and filtered DNS are listed in Table 3.1. The Strouhal number predicted by the AD-ROM is slightly lower than the Strouhal number computed from the filtered DNS data because the AD-ROM's shedding frequency is lower than the filtered DNS' shedding frequency (see Fig. 3.1). We note that the G-ROM fails to predict a clear shedding frequency and, thus, cannot yield a Strouhal number.

Table 3.2: Offline and online CPU times (in seconds) of DNS and ROMs.

DNS	G-ROM offline	G-ROM online	AD-ROM offline	AD-ROM online
2.3e+06	5.9e+04	9.2e+01	6.1e+04	1.4e+02

To measure the computational efficiency of the AD-ROM (which is one of the main requirements for any successful ROM), we list in Table 3.2 its offline and online CPU times together with the G-ROM offline and online CPU times and the DNS CPU time. We draw the following conclusions from Table 3.2:

First, the AD-ROM online CPU time is on the same order as the G-ROM online CPU time. Both online CPU times are about four orders of magnitude lower than the DNS CPU time.

Second, the offline CPU times of both the AD-ROM and the G-ROM are about two orders of magnitude lower than the DNS CPU time. We note, however, that the AD-ROM and G-ROM offline CPU times are significantly higher than their corresponding online CPU times.

Indeed, it is well known in the ROM community that in the offline stage, in order to assemble the vectors, matrices and tensors used in the dynamical systems formulation of

the G-ROM, the underlying numerical discretization (e.g., FE discretization) needs to be used. Thus, the G-ROM offline CPU time is significant. We also note that, since we use the FE-DF in the numerical discretization of the AD-ROM (see Section 3.4), additional vectors and matrices need to be assembled. This makes the AD-ROM offline CPU time higher than the G-ROM offline CPU time.

Finally, we note that in the applications generally targeted by ROMs, the models are used numerous times and for long time intervals [82, 118, 132]. Thus, in this type of applications, the bulk of the computational cost of the ROMs is paid in the online stage. Since, as mentioned above, the AD-ROM (and G-ROM) online CPU time is significantly lower than the DNS CPU time, we deem the AD-ROM as computationally efficient.

3.5.2 Comparison With Other ROMs

In Section 3.5.1, to assess the performance of the new AD-ROM, we compared it with the G-ROM and the filtered DNS data. Overall, the AD-ROM results were significantly more accurate than the G-ROM results. Thus, the AD-ROM satisfied the basic accuracy criterion of any successful ROM. Furthermore, the AD-ROM results were relatively close to the filtered DNS results, which prompted us to conclude that the AD-ROM was relatively accurate. A natural question, however, is how the AD-ROM's accuracy compares with the accuracies of other ROMs for convection-dominated flows.

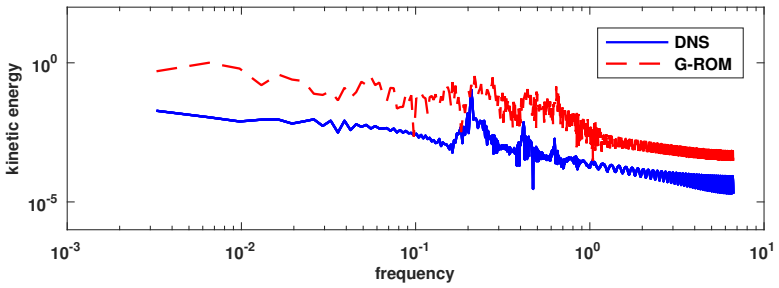
We emphasize that a thorough comparison of the AD-ROM with the plethora of ROMs for convection-dominated flows would require the numerical investigation of different types of ROM strategies for various computational settings and numerous parameter choices. Thus, while needed, this comparison is beyond the scope of this chapter. We will, however, present next a brief qualitative comparison of the AD-ROM with two other representative classes of ROMs: the EV-ROMs [12, 85, 162] and the Reg-ROMs [138, 165]. Since both the test problem and the statistics used in [162, 165] are identical to those used in this work, our qualitative comparison is meaningful.

Overall, the AD-ROM results are similar to those for the EV-ROMs in [162] and Reg-ROMs in [138, 165]: For example, the accuracy of the AD-ROM energy spectrum (see Fig. 3.1) is similar to the accuracy of the energy spectra of the *best* EV-ROMs (see Fig. 3 in [162]) and *best* Reg-ROMs (see Fig. 2 in [165]). The AD-ROM's accuracy is also similar to that of the best EV-ROMs and Reg-ROMs with respect to other statistics and time evolution of the ROM coefficients (compare, e.g., Fig. 3.5 in this chapter, Fig. 7 and Fig. 8 in [162], and Fig. 6 in [165]).

We believe that the AD-ROM's performance is impressive given that we did not use any explicit numerical dissipation mechanisms, which are generally expected for ROMs of convection-dominated flows. Indeed, we emphasize that the EV-ROMs include terms that allow explicit dissipation [12, 85, 162] and the Reg-ROMs increase the numerical stability by using explicit spatial filtering of various terms in the models [138, 165]. The AD-ROM, however, does not use any explicit dissipation mechanism. We followed this approach in order to decouple the AD and dissipation effects, which, we believe, is necessary in the first step of a careful assessment the AD procedure in a ROM setting.

Figure 3.1: Kinetic energy spectrum of the spatially filtered DNS (blue) and ROMs (red): (a) the G-ROM, and (b) the AD-ROM.

(a)



(b)

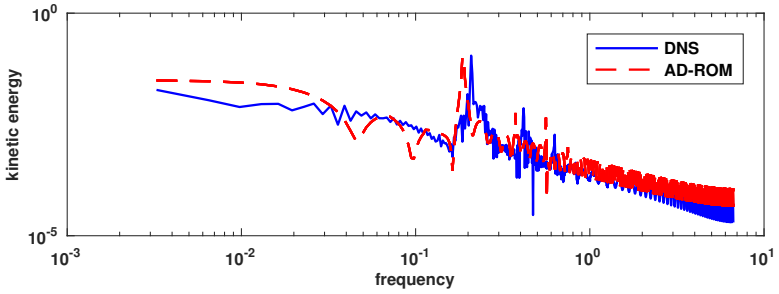
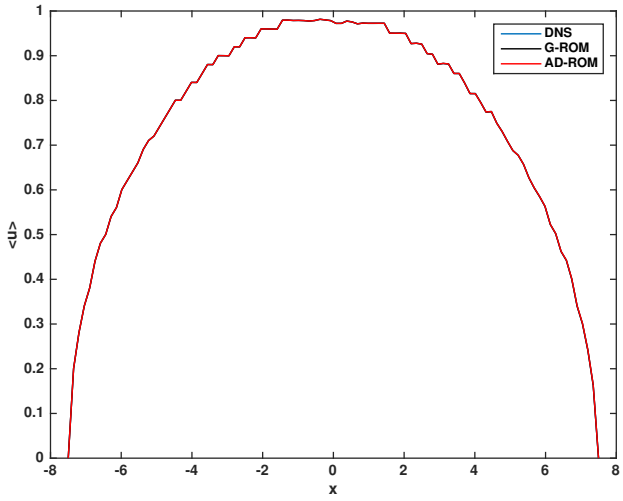
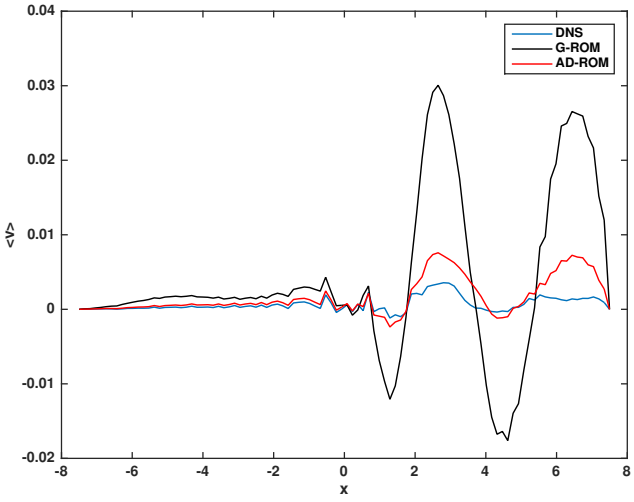


Figure 3.2: Mean velocity components of the spatially filtered DNS and ROMs: (a) $\langle u \rangle$ (the mean streamwise velocity), (b) $\langle v \rangle$ (the mean normal velocity), and (c) $\langle w \rangle$ (the mean spanwise velocity).

(a)



(b)



(c)

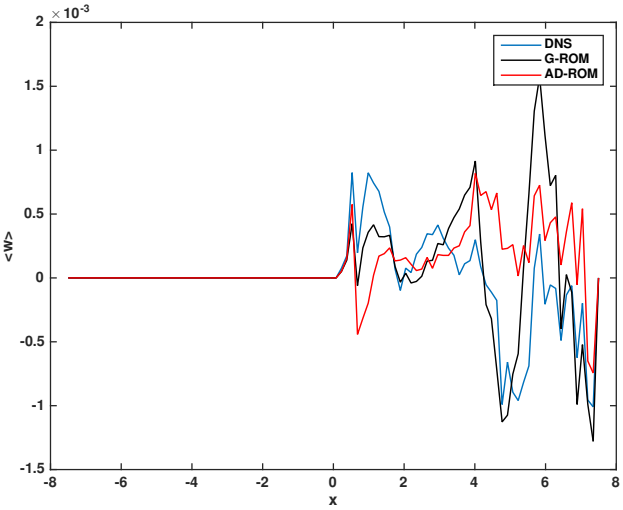
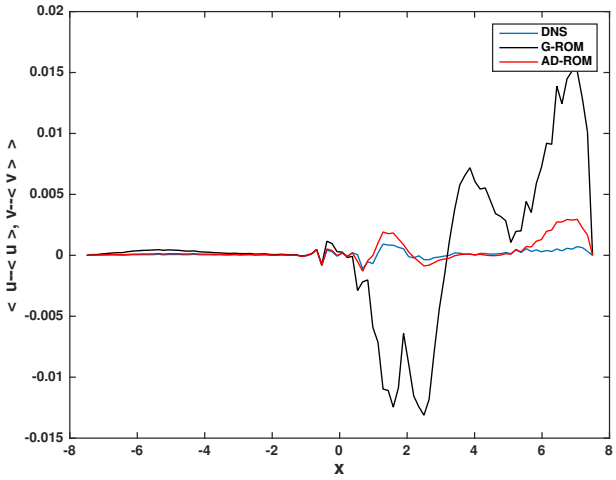
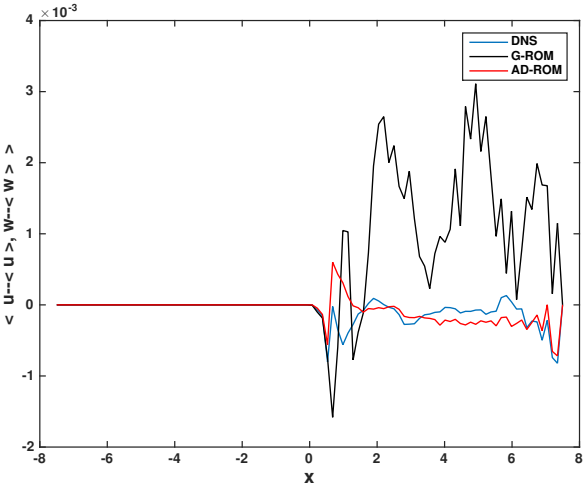


Figure 3.3: Reynolds stresses of the spatially filtered DNS and ROMs: (a) $\langle u - \langle u \rangle, v - \langle v \rangle \rangle$ (the xy-component of the Reynolds stress), (b) $\langle u - \langle u \rangle, w - \langle w \rangle \rangle$ (the xz-component of the Reynolds stress), and (c) $\langle v - \langle v \rangle, w - \langle w \rangle \rangle$ (the yz-component of the Reynolds stress).

(a)



(b)



(c)

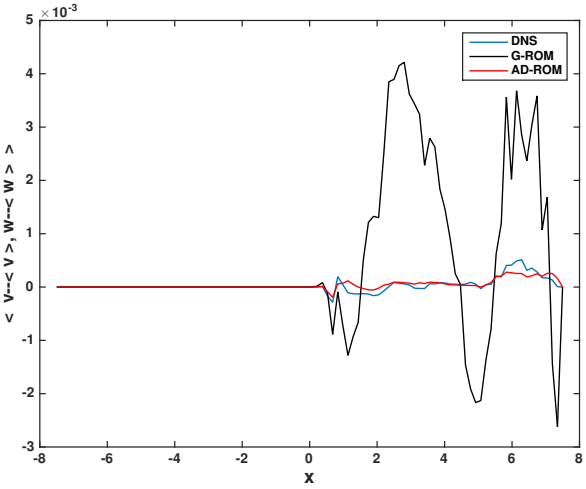
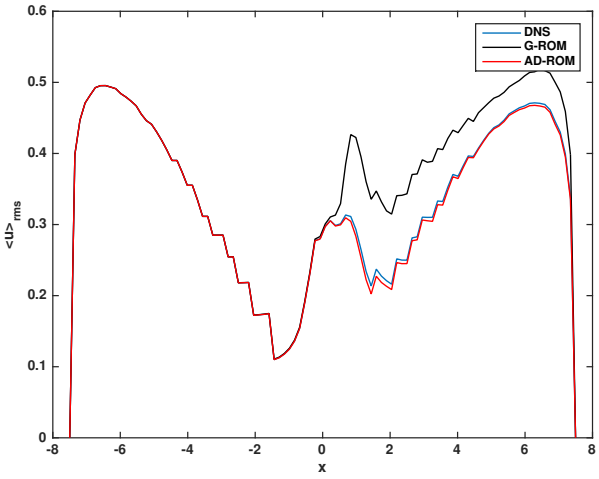
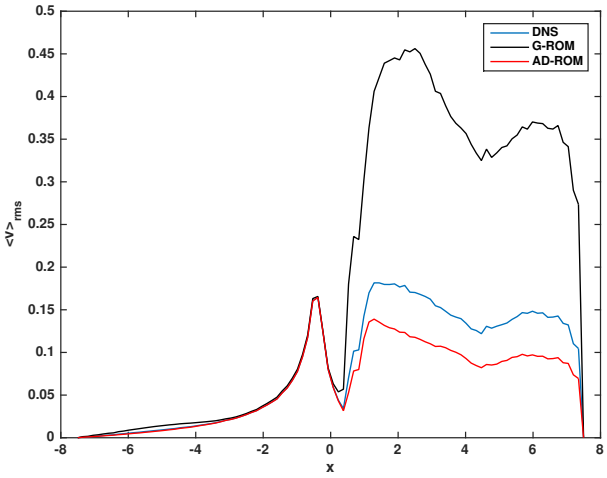


Figure 3.4: Rms values of the velocity fluctuations of the spatially filtered DNS and ROMs: (a) $\langle u \rangle_{rms} = \langle u - \langle u \rangle, u - \langle u \rangle \rangle$ (the rms value of the streamwise velocity fluctuations), (b) $\langle v \rangle_{rms} = \langle v - \langle v \rangle, v - \langle v \rangle \rangle$ (the rms value of the normal velocity fluctuations), and (c) $\langle w \rangle_{rms} = \langle w - \langle w \rangle, w - \langle w \rangle \rangle$ (the rms value of the spanwise velocity fluctuations).

(a)



(b)



(c)

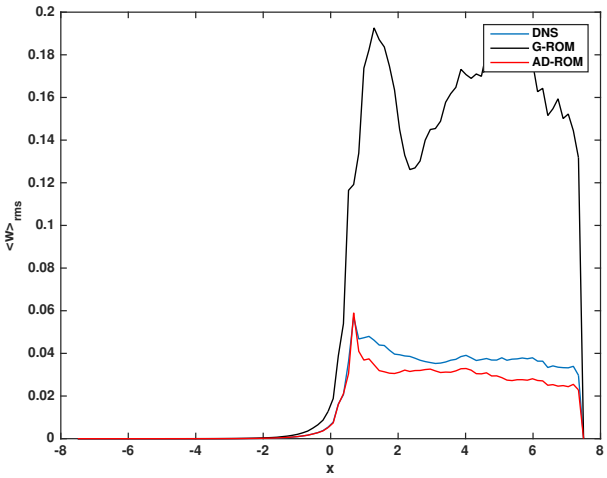
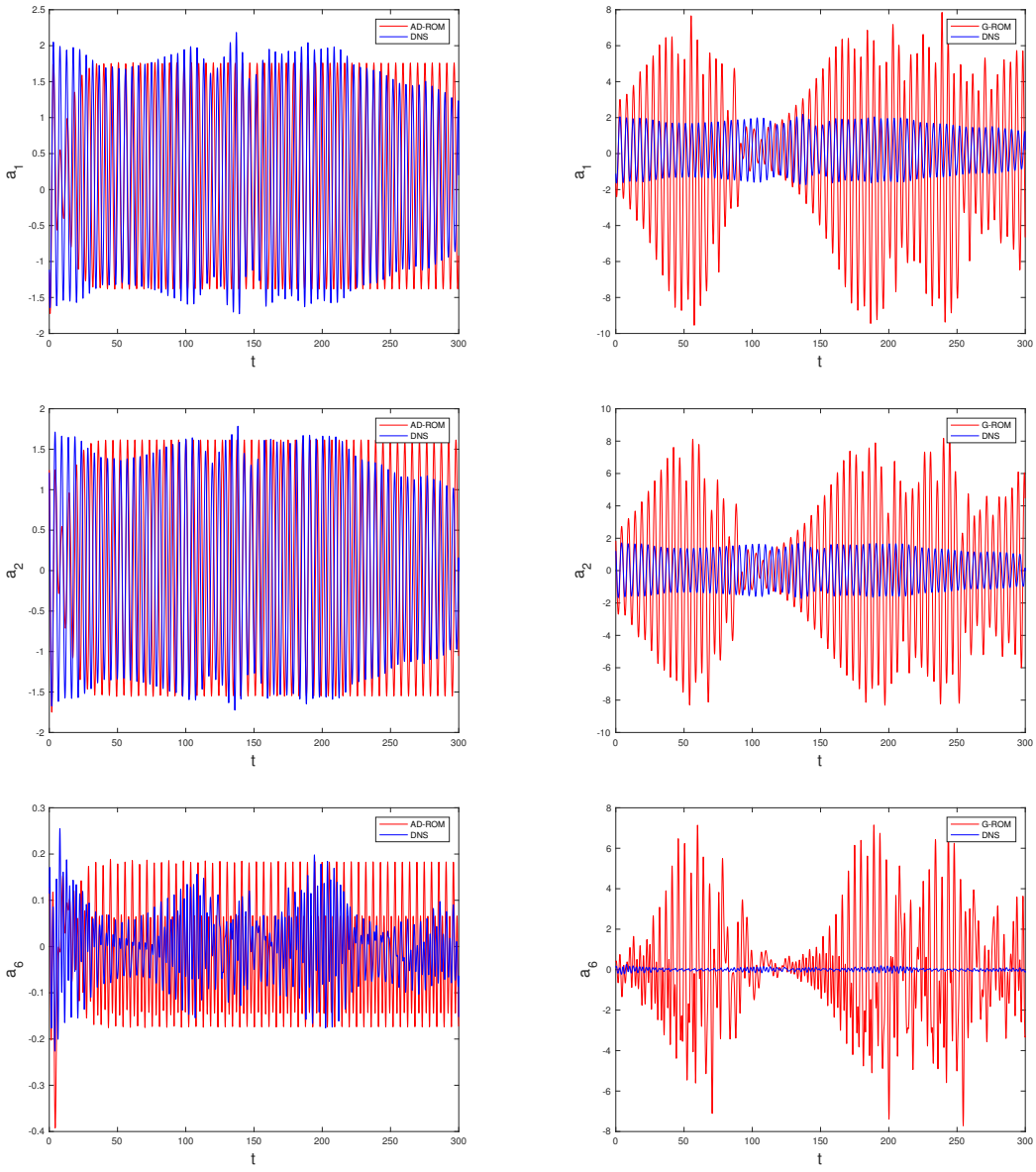


Figure 3.5: Time evolution of the POD basis coefficients a_1 (first row), a_2 (second row) and a_6 (third row) of the spatially filtered DNS (blue) and ROMs (red): the AD-ROM (left column), and G-ROM (right column).



3.6 Summary

We used explicit ROM spatial filtering to develop a large eddy simulation ROM (LES-ROM) framework. Within this LES-ROM framework, we proposed an approximate deconvolution ROM (AD-ROM). We assessed the AD-ROM in the numerical simulation of the 3D flow past a circular cylinder at $Re = 1000$. The numerical investigation showed that the accuracy of the AD-ROM was significantly higher than the accuracy of the standard G-ROM and on the same order as the accuracy of the eddy viscosity ROMs (EV-ROMs) [162] and Reg-ROMs [165]. Finally, we emphasize that the CPU time of the AD-ROM was orders of magnitude lower than the CPU time of the DNS.

These first steps in the investigation of the LES-ROM framework and AD-ROM yielded encouraging results. We emphasize, however, that there are numerous open questions, both at a fundamental (theoretical and modeling) and practical (discretization) level. This is, of course, natural, since in the LES field the same issues have been tackled over decades. Some of the most stringent issues that need to be addressed in the LES-ROM framework (in general) and AD-ROM (in particular) are listed in chapter 7.

Chapter 4

Calibrated Filtered Reduced Order Model (CF-ROM)

In this chapter, we use the ROM spatial filtering idea to propose a novel ROM closure model, named *calibrated filtered reduced order model (CF-ROM)*, for the numerical simulation of general nonlinear PDEs that are amenable to reduced order modeling. The novel CF-ROM framework consists of two steps: (i) In the first step, we use explicit ROM spatial filtering of the nonlinear PDE to construct a filtered ROM. This filtered ROM is low-dimensional, but is not closed (because of the nonlinearity in the given PDE). (ii) In the second step, we use a calibration procedure to close the filtered ROM, i.e., to model the interaction between resolved and unresolved modes. To this end, we use a linear or quadratic ansatz to model this interaction. To find the new coefficients in the closed filtered ROM, we solve an optimization problem that minimizes the difference between the full order model data and our ansatz. Although we use a fluid dynamics setting to illustrate how to construct and use the CF-ROM framework, we emphasize that it is built on general ideas of spatial filtering and optimization and is independent of (restrictive) phenomenological arguments. Thus, the CF-ROM framework can be applied to a wide variety of PDEs.

4.1 Introduction

The quest for better ROMs has yielded two types of approaches: In the first approach, the ROM is fixed and the ROM basis $\mathbf{X}^r := \text{span}\{\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_r\}$ is improved (see, e.g., the recent survey in [153]). In the second approach, the basis is fixed and the ROM is improved. In this chapter, we exclusively use the latter. Specifically, we try to answer the following question:

$$\boxed{\text{For a fixed ROM space } \mathbf{X}^r, \text{ what is the best ROM in } \mathbf{X}^r?} \tag{4.1}$$

The standard ROM is developed as follows: (i) the given nonlinear PDE($\boldsymbol{u}$) is projected onto the ROM space $\mathbf{X}^r$; (ii) the Galerkin truncation $\boldsymbol{u}_r = \sum_{j=1}^{r} u_j \boldsymbol{\varphi}_j$ is used; (iii) $\boldsymbol{u}$ is replaced with $\boldsymbol{u}_r$ in step (i). The standard ROM is often efficient and relatively accurate [16, 85, 118], but can fail in realistic applications when large numbers of modes are needed to accurately represent the system. To ensure a low computational cost, it is desirable that ROMs use only a few modes (i.e., low r values) and discard the remaining modes

$\{\boldsymbol{\varphi}_{r+1}, \ldots, \boldsymbol{\varphi}_d\}$. The resulting low-dimensional ROM, however, can yield inaccurate results and numerical oscillations (see, e.g., [7, 13, 18, 25, 26, 43, 76, 125, 131]). The general explanation for these inaccurate results is that the models fail to account for the interaction between resolved and unresolved modes [12, 57, 77, 121, 162, 165, 168].

To address the standard ROM's inaccuracy, we propose a novel *calibrated-filtered ROM (CF-ROM)* framework for the numerical simulation of *general nonlinear PDEs*. The new framework explicitly models the interaction between the resolved modes $\{\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_r\}$ and unresolved modes $\{\boldsymbol{\varphi}_{r+1}, \ldots, \boldsymbol{\varphi}_d\}$. This is in stark contrast with the standard ROM, which simply ignores this interaction. We believe that the new CF-ROM framework represents a potential answer to the question (4.1), as we explain below, after first describing the general ideas. The CF-ROM framework is schematically illustrated in (4.2), below.

$$\boxed{\text{FOM} \xrightarrow{\text{filtering}} \text{F-ROM} \xrightarrow{\text{calibration}} \text{CF-ROM}} \tag{4.2}$$

In the first step of (4.2), the full order model (FOM), which contains all the information in the underlying system, is filtered with an *explicit ROM spatial filter*. The resulting *filtered ROM (F-ROM)* approximates only the large spatial structures of the system and, therefore, requires fewer modes than the FOM. To create a usable model, one still needs to solve the *ROM closure problem* to model the interaction between the resolved and unresolved modes in the F-ROM, which is done in the second step of (4.2). We use a calibration procedure to *construct* the actual ROM (i.e., not to data fit its solution [1, 54, 56, 160]). We employ a linear or quadratic ansatz to model the interaction between the resolved and unresolved modes in the F-ROM, and then we find the ROM coefficients by solving a (low-dimensional) optimization problem that minimizes the difference between the full order model data and our ansatz.

Our proposed CF-ROM framework has some connections to some other popular models, in particular the *nonlinear Galerkin* [64] and *variational multiscale* [87] methods, since they all use the small-large scale separation. The CF-ROM is also reminiscent of LES in a sense, since it uses spatial filtering. However, CF-ROM is different from the nonlinear Galerkin method since it can use general types of spatial filtering, whereas the latter is based on projection. CF-ROM is also different from the variational multiscale method and LES since it uses calibration to model the interaction with the unresolved modes, whereas the other two methods use phenomenological arguments based on, e.g., the energy cascade and eddy viscosity. Indeed, the fact that CF-ROM does not use phenomenological arguments is precisely whey it is not restricted in its application, and can be utilized for a wide variety of nonlinear PDEs.

The data-driven inference method proposed in [129] is also similar (see also [128]) in spirit to the CF-ROM. There is, however, a significant difference between the two: In [129] all the PDE operators are inferred from the data; in the CF-ROM, we use the available data only to infer the closure model. Finally, the EV closure model proposed in [131] uses the available data to calibrate the EV coefficient. We emphasize that this approach is fundamentally different from ours: We use calibration to build a CF-ROM closure model that can be applied to general PDEs; the EV closure model in[131], however, can be applied only to turbulent flow computations, where the energy cascade concept is valid. Other related, but different closure strategies can be found in [77, 127].

4.2 Filtered ROMs (F-ROM)

In this section, we present details regarding the first step of the CF-ROM framework (4.2). That is, we discuss the creation of the *filtered ROM (F-ROM)*, from which the ROM closure problem will reveal itself. To this end, in Section 4.2.1 we present the ROM projection, which is the explicit ROM spatial filter that we use to construct the CF-ROM. In Section 4.2.2, we develop the F-ROM framework, which has also been used to develop LES-ROMs [168]. We emphasize that the ROM closure problem is treated completely differently in CF-ROM and LES-ROM: CF-ROM uses calibration, while LES-ROMs generally use phenomenological arguments (e.g., energy cascade and eddy viscosity); see however the AD-ROM in Chapter 3 for an exception.

4.2.1 Explicit ROM Spatial Filtering

In this chapter, we exclusively use the ROM projection [162, 165] as spatial filter, but we note that we could also other spatial filters (e.g., the ROM differential filter [138, 165, 168]).

For a fixed $r \leq d$ and a given $\boldsymbol{u} \in \boldsymbol{X}^h$, the ROM projection seeks $\overline{\boldsymbol{u}}^r \in \boldsymbol{X}^r$ such that

$$\left(\overline{\boldsymbol{u}}^r, \boldsymbol{\varphi}_j\right) = (\boldsymbol{u}, \boldsymbol{\varphi}_j) \quad \forall\, j = 1, \ldots r\,. \tag{4.3}$$

The ROM projection (4.3) has been used for theoretical purposes, e.g., in the error analysis of the G-ROM [100], and as a smoothing ROM spatial filter in [162, 165].

4.2.2 F-ROM Framework

We now outline the F-ROM framework, which has also been used to develop LES-ROMs in Chapter 3. We start with the spatially filtered NSE (2.39) – (2.40):

$$\left(\frac{\partial \overline{\boldsymbol{u}}}{\partial t}, \boldsymbol{\phi}\right) + Re^{-1}\left(\nabla \overline{\boldsymbol{u}}, \nabla \boldsymbol{\phi}\right) + \left(\nabla \cdot \left(\overline{\boldsymbol{u}}\,\overline{\boldsymbol{u}}\right), \boldsymbol{\phi}\right) + \left(\nabla \cdot \boldsymbol{\tau}, \boldsymbol{\phi}\right) = \boldsymbol{0}\,, \tag{4.4}$$

where

$$\boldsymbol{\tau} = \overline{\boldsymbol{u}\,\boldsymbol{u}} - \overline{\boldsymbol{u}}\,\overline{\boldsymbol{u}} \tag{4.5}$$

is the *subfilter-scale stress tensor*. The spatial structures in the spatially filtered NSE (4.4) are larger than the spatial structures in the NSE (1.3). Thus, we expect that for a fixed target numerical accuracy of the ROM, the spatially filtered NSE will require fewer POD modes than the NSE, which is advantageous from a computational point of view.

Of course, to develop a useful ROM from the spatially filtered NSE (4.4), we need to do the following: (i) use a ROM approximation for the continuous velocity field $\overline{\boldsymbol{u}}$; and (ii) use a ROM approximation of the spatial filter. We make the following choices: Use $\boldsymbol{u}_d \sim \boldsymbol{u}$ in (i) (where d is the rank of the snapshot matrix) and $\overline{\boldsymbol{u}}^r \sim \overline{\boldsymbol{u}}$ in (ii) (where $\overline{\boldsymbol{u}}^r$ is the ROM projection (4.3)).

Using $\boldsymbol{u}_d \sim \boldsymbol{u}$ in (i) means that we employ the best possible approximation of the continuous velocity field $\boldsymbol{u}$ in the set of snapshots (i.e., in $\boldsymbol{X}^d$). Using the ROM projection

onto $\boldsymbol{X}^r$ as the spatial filter in (ii) means that we are projecting the equations from $\boldsymbol{X}^d$ onto $\boldsymbol{X}^r$. With these choices in the spatially filtered NSE (4.4), we obtain the spatially *filtered ROM (F-ROM)*: $\forall\, i = 1, \ldots, r$,

$$\left(\frac{\partial \overline{\boldsymbol{u}_d}^r}{\partial t}, \boldsymbol{\varphi}_i\right) + Re^{-1}\left(\nabla \overline{\boldsymbol{u}_d}^r, \nabla \boldsymbol{\varphi}_i\right) + \left(\nabla \cdot \left(\overline{\boldsymbol{u}_d}^r\, \overline{\boldsymbol{u}_d}^r\right), \boldsymbol{\varphi}_i\right) + \left(\nabla \cdot \boldsymbol{\tau}_r, \boldsymbol{\varphi}_i\right) = \boldsymbol{0}\,, \tag{4.6}$$

where

$$\boldsymbol{\tau}_r = \overline{\boldsymbol{u}_d\, \boldsymbol{u}_d}^r - \overline{\boldsymbol{u}_d}^r\, \overline{\boldsymbol{u}_d}^r\,. \tag{4.7}$$

Since we are using the ROM projection onto $\boldsymbol{X}^r$ as the spatial filter, we have

$$\overline{\boldsymbol{u}_d}^r = \boldsymbol{u}_r. \tag{4.8}$$

Thus, the F-ROM (4.6) becomes

$$\left(\frac{\partial \boldsymbol{u}_r}{\partial t}, \boldsymbol{\varphi}_i\right) + Re^{-1}\left(\nabla \boldsymbol{u}_r, \nabla \boldsymbol{\varphi}_i\right) + \left(\nabla \cdot \left(\boldsymbol{u}_r\, \boldsymbol{u}_r\right), \boldsymbol{\varphi}_i\right) + \left(\nabla \cdot \boldsymbol{\tau}_r, \boldsymbol{\varphi}_i\right) = \boldsymbol{0}\,, \tag{4.9}$$

where the ROM stress tensor is

$$\boldsymbol{\tau}_r = \overline{\boldsymbol{u}_d\, \boldsymbol{u}_d}^r - \boldsymbol{u}_r\, \boldsymbol{u}_r\,. \tag{4.10}$$

The F-ROM (4.9) is an r-dimensional ODE system for $\boldsymbol{u}_r$. Since $r \ll N$, the F-ROM (4.9) is a computationally efficient surrogate model for the FOM (i.e., the FE approximation of the NSE, which is an N-dimensional ODE system). We emphasize, however, that the F-ROM (4.9) is not a closed system of equations, since the ROM stress tensor $\boldsymbol{\tau}_r$ depends on $\boldsymbol{u}_d$ (see equation (2.40)). Thus, to close the F-ROM (4.9), we need to solve the ROM closure problem [67, 121, 144, 162].

Note that by neglecting the last term on the LHS of (4.9), the F-ROM (4.9) is identical to the standard G-ROM (5.11). Thus, *formally*, one could write the following decomposition:

$$\boxed{\text{F-ROM} = \text{G-ROM} + \text{ROM Closure Model}} \tag{4.11}$$

In the ROM community, it is well known that the standard G-ROM with few ROM modes (i.e., with small r) generally yields inaccurate results in complex applications. Probably the most popular "solution" to this problem is to supplement the G-ROM with extra terms:

$$\boxed{\text{Good ROM} = \text{G-ROM} + \text{Extra Terms}} \tag{4.12}$$

Numerical and physical arguments are used to build the Extra Terms in (4.12). Furthermore, the decomposition (4.12) is usually called "solving the ROM closure problem" [67, 121, 144, 162], and the Extra Terms are thought to model the effect of the discarded ROM modes $\{\boldsymbol{\varphi}_{r+1}, \ldots, \boldsymbol{\varphi}_d\}$. We emphasize, however, that to our knowledge no precise definition of the ROM closure problem has been given so far. The F-ROM (4.9) allows for the first time a clear definition of the ROM closure problem (in Definition 2.5).

Remark 4.1 (F-ROM Consistency)**.** *Probably the most popular ROMs of the form* (4.12) *are the eddy viscosity closure models pioneered by Lumley and his group [12, 85] (see Section 2.3) and extended to state-of-the-art LES closure models in [162]. As emphasized in, e.g., Section 2.4 in [14], although these eddy viscosity ROM closure models have been successfully applied to a large number of complex, high Reynolds number flows, they have a significant drawback, namely the loss of consistency between the NSE and the ROM. Indeed, since the G-ROM in* (4.12) *is modified empirically, the resulting eddy viscosity ROM no longer corresponds to a Galerkin projection of the NSE.*

The F-ROM, however, is consistent with the NSE. Indeed, the F-ROM (4.9) *is just the projection of the full order model (i.e., the best representation of the NSE in the snapshot space,* $\boldsymbol{X}^d = span\{\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_d\}$*) onto the ROM space,* $\boldsymbol{X}^r$*.*

4.3 Calibrated ROMs (C-ROMs)

In this section, we outline the general ROM calibration procedure (see, e.g., [54]), and in Section 4.4 we use it to construct the new CF-ROM from the F-ROM.

4.3.1 Modified G-ROM

We start by rewriting the standard G-ROM (6.6) as

$$\dot{\boldsymbol{a}} = \boldsymbol{F}(\boldsymbol{a})\,, \tag{4.13}$$

where, for $i = 1, \ldots, r$,

$$F_i(\boldsymbol{a}) = b_i + \sum_{m=1}^{r} A_{im}\, a_m + \sum_{m=1}^{r}\sum_{n=1}^{r} B_{imn}\, a_n\, a_m\,. \tag{4.14}$$

In the ROM literature, ROM calibration is used when, for different reasons, the G-ROM (6.6) is deemed inaccurate and is replaced by the following *modified G-ROM*:

$$\dot{\boldsymbol{a}} = \boldsymbol{F}(\boldsymbol{a}) + \boldsymbol{G}\,. \tag{4.15}$$

Thus, the modified G-ROM (4.15) depends on a new quantity, $\boldsymbol{G}$, whose aim is to make the modified G-ROM (4.15) more accurate than the standard G-ROM (6.6). Examples of quantities $\boldsymbol{G}$ used in the current C-ROMs include the pressure representation proposed in [120].

4.3.2 Ansatz: Linear or Quadratic?

To make the modified G-ROM (4.15) fit the general ROM framework, the following ansatz is generally used in the current C-ROMs:

$$\boldsymbol{G} = \boldsymbol{G}(\boldsymbol{a})\,. \tag{4.16}$$

In fact, to make the modified G-ROM (4.15) even further resemble the standard G-ROM (6.6), the following choices are generally made in ansatz (4.16): For $i = 1, \ldots, r$,

$$G_i(\boldsymbol{a}) = \sum_{m=1}^{r} \tilde{A}_{im}\, a_m \qquad \text{linear ansatz} \tag{4.17}$$

or

$$G_i(\boldsymbol{a}) = \sum_{m=1}^{r} \tilde{A}_{im}\, a_m + \sum_{m=1}^{r}\sum_{n=1}^{r} \widetilde{B}_{imn}\, a_m\, a_n \quad \text{quadratic ansatz}\,. \tag{4.18}$$

Note that the linear ansatz (4.17) is a particular case of the quadratic ansatz (4.18), in which $\widetilde{B}_{imn} = 0,\ \forall\, m, n$. Thus, for clarity purposes, in this section we will exclusively use the quadratic ansatz (4.18) as our base ansatz.

The linear ansatz (4.17) was used, e.g., in [39] (for the ROM closure and ROM pressure, see page 145), [67] (for the ROM closure and ROM pressure, see pages 356–357), [69] (for the ROM pressure, see page 164) and [120] (for the ROM pressure, see page 361). The quadratic ansatz (4.18) was used, e.g., in [120] (for the ROM pressure, see equations (2.18)–(2.19)). The linear ansatz and the quadratic ansatz were shown to yield similar qualitative results for the pressure representation (see page 361 in [120]).

Using the quadratic ansatz (4.18) in the modified G-ROM (4.15) yields, for $i = 1, \ldots, r$,

$$\dot{a}_i = b_i + \sum_{m=1}^{r} \left(A_{im} + \tilde{A}_{im}\right)\, a_m + \sum_{m=1}^{r}\sum_{n=1}^{r} \left(B_{imn} + \widetilde{B}_{imn}\right)\, a_n\, a_m\,. \tag{4.19}$$

4.3.3 Optimization: Constrained or Unconstrained?

The ROM calibration aims at finding the coefficients $\tilde{A}_{im}$ and $\widetilde{B}_{imn}$ that ensure the highest accuracy of the modified G-ROM (4.19). In the current C-ROMs, the accuracy is based on the difference between the ROM coefficients a_i in the modified G-ROM (4.19) and the ROM coefficients obtained from the snapshots, a_i^{snap}. The values $a_i^{snap}(t_j)$, with snapshot time instances $t_j,\ j = 1, \ldots, M$, are obtained by projecting the corresponding snapshots $\boldsymbol{u}(t_j) = \sum_{i=1}^{d} a_i^{snap}(t_j)\, \boldsymbol{\varphi}_i$ onto the POD basis functions $\boldsymbol{\varphi}_i$ and using the orthogonality of the POD basis functions: For $i = 1, \ldots, d$ and $j = 1, \ldots, M$,

$$a_i^{snap}(t_j) = \left(\boldsymbol{u}(t_j), \boldsymbol{\varphi}_i\right). \tag{4.20}$$

To find the coefficients $\tilde{A}_{im}$ and $\widetilde{B}_{imn}$ that ensure the highest accuracy of the modified G-ROM (4.19), the current C-ROMs solve an optimization problem, which can be of two types: an ODE *constrained optimization* problem

$$\min_{\tilde{A}_{im},\, \widetilde{B}_{imn}} \sum_{j=1}^{M} \|\boldsymbol{a}(t_j) - \boldsymbol{a}^{snap}(t_j)\|^2 \tag{4.21}$$

$$\text{s.t. } \boldsymbol{a} \text{ solves } (4.19)\,, \tag{4.22}$$

where $\|\cdot\|$ is the Euclidian norm in $\mathbb{R}^r$, or an *unconstrained optimization* problem

$$\min_{\tilde{A}_{im},\widetilde{B}_{imn}} \sum_{j=1}^{M} \|\boldsymbol{G}(\boldsymbol{a}^{snap}(t_j)) - \boldsymbol{G}^{snap}\|^2 , \tag{4.23}$$

where $\boldsymbol{G}^{snap}$ is the *true* quantity $\boldsymbol{G}$ computed with snapshot data.

The constrained minimization problem (4.21)–(4.22) was used, e.g., in [39] (see page 145), [54] (see page 277), [56] (see page 203), [67] (see page 357), and [69] (see page 164). As explained on page 357 in [67], the constrained minimization problem (4.21)–(4.22) is usually solved in two steps: First, one passes from the discrete to the continuous time variable by using a spline interpolation. Then, in the continuous setting, one solves an optimality problem that consists of the direct problem, the adjoint problem and optimality conditions.

The unconstrained minimization problem (4.23) was used, e.g., in [120] (see page 361). This unconstrained minimization problem was solved in [120] at a discrete level (not at a continuous level, as done for the constrained minimization problem (4.21)–(4.22)). Linear regression was used in [120] to solve the resulting overdetermined linear system.

Comparison Both the constrained minimization problem (4.21)–(4.22) and the unconstrained minimization problem (4.23) have been used to determine the C-ROM coefficients. For example, Noack et al. [120] solved the unconstrained minimization problem (4.23), whereas Galletti et al. [69] solved the constrained minimization problem (4.21)–(4.22).

We are not aware of any comparison of the two approaches. Although a comparison of the two approaches is needed, it is beyond the scope of the present work. We only mention that, for a fair comparison, we believe that the *predictive capabilities* of the C-ROM should be assessed (see, e.g., [162] for other examples of predictive ROMs). Indeed, we believe that by solving the constrained minimization problem (4.21)–(4.22) (as in Galletti et al. [69]), one effectively ensures that the C-ROM results are as close as possible to the NSE results. We emphasize, however, that this is true only on the time interval on which the snapshots were collected, say, $[0, T]$. On a longer time interval, say, $[0, 10\,T]$, the C-ROM obtained with the approach used in [69] might not perform optimally. In other words, the predictive capabilities of the C-ROM are not clear. On the other hand, by solving the unconstrained minimization problem (4.23) (as in Noack et al. [120] and in this chapter), one aims at finding the best model for the unknown quantity. The hope is that this model has some universal features that can be determined from the available snapshots. If this is the case, then the C-ROM obtained with the approach used in Noack et al. [120] and in our work might have predictive capabilities, i.e., it might yield accurate results on a longer time interval, say, $[0, 10\,T]$. The following quote from page 361 in Noack et al. [120] seems to support our conjecture: "In a recent study of wake flows, Galletti et al. [69] use the same ansatz for the pressure term. In that study the coefficients are determined from an optimization problem which minimizes the error between the Galerkin solution and the Navier–Stokes simulation. The present empirical approach minimizes the error between the pressure term in primitive variables and in the Galerkin model representation. By construction, the term of Galletti et al. [69] is more accurate in the short term whereas the present term is more forgiving of long-term phase errors."

For simplicity purposes, in this work we are using the unconstrained minimization problem (4.23). We plan to compare the two approaches and investigate the potential predictive capabilities of the CF-ROM in a future study.

4.3.4 C-ROM Algorithm

The optimal coefficients $\tilde{A}_{im}^{opt}$ and $\widetilde{B}_{imn}^{opt}$ that solve the minimization problems (4.21)–(4.22) or (4.23) are used in the modified G-ROM (4.19), yielding the *calibrated ROM (C-ROM)*

$$\dot{a}_i = \sum_{m=1}^{r} \left(A_{im} + \tilde{A}_{im}^{opt} \right) a_m + \sum_{m=1}^{r} \sum_{n=1}^{r} \left(B_{imn} + \widetilde{B}_{imn}^{opt} \right) a_n \, a_m \, . \tag{4.24}$$

In Algorithm 1, we summarize the C-ROM for a generic quantity of interest, $\boldsymbol{G}$.

Algorithm 1 C-ROM

Step 1 Consider the modified G-ROM $\dot{\boldsymbol{a}} = \boldsymbol{F}(\boldsymbol{a}) + \boldsymbol{G}$

Step 2 Use ansatz $\boldsymbol{G} = \boldsymbol{G}(\boldsymbol{a})$ for the unknown quantity $\boldsymbol{G}$:

$$G_i(\boldsymbol{a}) = \sum_{m=1}^{r} \tilde{A}_{im}\, a_m \qquad \text{linear ansatz} \tag{4.25}$$

or

$$G_i(\boldsymbol{a}) = \sum_{m=1}^{r} \tilde{A}_{im}\, a_m + \sum_{m=1}^{r}\sum_{n=1}^{r} \widetilde{B}_{imn}\, a_m\, a_n \qquad \text{quadratic ansatz}\,. \tag{4.26}$$

Step 3 Solve the constrained minimization problem (4.27)–(4.28) or the unconstrained minimization problem (4.29) to determine the unknown coefficients $\tilde{A}_{im}$ and $\tilde{B}_{imn}$ in (4.25)–(4.26):

$$\min_{\tilde{A}_{im},\, \widetilde{B}_{imn}} \sum_{j=1}^{M} \|\boldsymbol{a}(t_j) - \boldsymbol{a}^{snap}(t_j)\|^2 \tag{4.27}$$

$$\text{s.t. } \boldsymbol{a} \text{ solves } (4.19)\,, \tag{4.28}$$

or

$$\min_{\tilde{A}_{im},\, \widetilde{B}_{imn}} \sum_{j=1}^{M} \|\boldsymbol{G}(\boldsymbol{a}^{snap}(t_j)) - \boldsymbol{G}^{snap}\|^2\,. \tag{4.29}$$

Step 4 The C-ROM has the following form:

$$\dot{a}_i = \sum_{m=1}^{r} \left(A_{im} + \tilde{A}_{im}^{opt}\right) a_m + \sum_{m=1}^{r}\sum_{n=1}^{r} \left(B_{imn} + \widetilde{B}_{imn}^{opt}\right) a_n\, a_m\,, \tag{4.30}$$

where the optimal coefficients $\tilde{A}_{im}^{opt}$ and $\widetilde{B}_{imn}^{opt}$ solve the minimization problems (4.27)–(4.28) or (4.29).

4.4 Calibrated Filtered ROM (CF-ROM)

The F-ROM (4.9) naturally fits into the general ROM calibration framework developed in Section 4.3. Indeed, the F-ROM (4.9) can be written as the weak form of the modified G-ROM (4.15), i.e., as the sum of the standard G-ROM (4.13) and the unknown ROM stress tensor contribution $\left(\nabla \cdot \boldsymbol{\tau}_r, \boldsymbol{\varphi}_i\right)$, where $\boldsymbol{\tau}_r$ is defined in (2.39). Thus, the general ROM calibration procedure presented in Section 4.3 can be used as ROM closure modeling, i.e., to model the term $\left(\nabla \cdot \boldsymbol{\tau}_r, \boldsymbol{\varphi}_i\right)$ in (4.9) as a function of the ROM coefficients ($\boldsymbol{a}$). In the remainder of this section, we use the general ROM calibration framework developed in

Section 4.3 to solve the closure problem in the F-ROM (4.9).

We start by denoting by $\boldsymbol{G}$ the unknown ROM stress tensor contribution: For $i = 1, \ldots, r$,

$$G_i(\boldsymbol{a}) := \Big(\nabla \cdot \boldsymbol{\tau}_r, \boldsymbol{\varphi}_i\Big)\,. \tag{4.31}$$

Next, we make a linear or quadratic ansatz for $\boldsymbol{G}$ in (4.31) (see equations (4.17) and (4.18), respectively):

$$G_i = \sum_{m=1}^{r} \tilde{A}_{im}\, a_m + \sum_{m=1}^{r}\sum_{n=1}^{r} \widetilde{B}_{imn}\, a_m\, a_n\,. \tag{4.32}$$

The next step in the ROM calibration framework is solving the constrained minimization problem (4.21)–(4.22) or the unconstrained minimization problem (4.23). For simplicity purposes, we will solve the latter. To this end, we first compute the ROM coefficients from the snapshot data ($a_i^{snap}(t_j)$) by using equation (4.20). Next, we compute the *true* quantity $\boldsymbol{G}$ from the snapshot data: $\forall\, i = 1, \ldots, r$,

$$\begin{aligned} G_i^{snap}(t_j) &= \Big(\nabla \cdot \Big[\overline{\boldsymbol{u}_d^{snap}(t_j)\, \boldsymbol{u}_d^{snap}(t_j)}^{r} - \boldsymbol{u}_r^{snap}(t_j)\, \boldsymbol{u}_r^{snap}(t_j)\Big], \boldsymbol{\varphi}_i\Big) \\ &= \Bigg(\nabla \cdot \Bigg[\overline{\left(\sum_{k_1=1}^{d} a_{k_1}^{snap}(t_j)\boldsymbol{\varphi}_{k_1}\right)\left(\sum_{k_2=1}^{d} a_{k_2}^{snap}(t_j)\boldsymbol{\varphi}_{k_2}\right)}^{r} \\ &\quad - \left(\sum_{k_3=1}^{r} a_{k_3}^{snap}(t_j)\boldsymbol{\varphi}_{k_3}\right)\left(\sum_{k_4=1}^{r} a_{k_4}^{snap}(t_j)\boldsymbol{\varphi}_{k_4}\right)\Bigg], \boldsymbol{\varphi}_i\Bigg)\,, \end{aligned} \tag{4.33}$$

where

$$\boldsymbol{u}_d^{snap}(t_j) = \sum_{k=1}^{d} a_k^{snap}(t_j)\, \boldsymbol{\varphi}_k\,, \qquad \boldsymbol{u}_r^{snap}(t_j) = \sum_{k=1}^{r} a_k^{snap}(t_j)\, \boldsymbol{\varphi}_k\,. \tag{4.34}$$

Remark 4.2 (Computational Efficiency)**.** *In practical settings, where the rank of the snapshot matrix can be extremely large (e.g., $d = \mathcal{O}(1000)$), using $\boldsymbol{u}_d^{snap} \in \boldsymbol{X}^d$ in (4.33) would be extremely costly. One possible solution would be to replace $\boldsymbol{u}_d^{snap}$ with, say, $\boldsymbol{u}_{2r}^{snap}$ and the ROM projection on $\boldsymbol{X}^d$ with the ROM projection on $\boldsymbol{X}^{2r}$. We numerically investigate the accuracy of this approximation in Section 4.5.*

Finally, we compute the quantity $\boldsymbol{G}$ from the linear or quadratic ansatz (4.32) and the snapshot data:

$$G_i\Big(\boldsymbol{a}^{snap}(t_j)\Big) = \sum_{m=1}^{r} \tilde{A}_{im}\, a_m^{snap}(t) + \sum_{m=1}^{r}\sum_{n=1}^{r} \widetilde{B}_{imn}\, a_m^{snap}(t)\, a_n^{snap}(t)\,. \tag{4.35}$$

Plugging (4.20), (4.33), and (4.35) into the unconstrained minimization problem (4.23), we obtain a *linear least squares* problem.

Using the quadratic ansatz in (4.35) seems more appropriate from a physical point of view. Indeed, the ROM stress tensor $\boldsymbol{\tau}_r$ is closely connected to the nonlinearity in the NSE. Since this nonlinearity yields a quadratic term in the G-ROM (4.15), one would expect that the ROM stress tensor be best approximated by a quadratic term, too, just as in the quadratic ansatz in (4.35). From a numerical point of view, however, the resulting nonlinear least

squares problem seems more challenging. In this case, one could use, e.g., Gauss-Newton methods or similar approaches (see Chapter 10 in [122]).

Using the linear ansatz in (4.35) does not have as much physical support as using the quadratic ansatz. For simplicity purposes, however, in what follows (including the numerical investigation in Section 4.5), we will use the linear ansatz. We plan to investigate the more physical quadratic ansatz in a future study.

Next, we show how the unconstrained minimization problem can be numerically solved with the linear ansatz. Plugging (4.33) and (4.35) with the linear ansatz into the unconstrained minimization problem (4.23), we obtain the following minimization problem:

$$\min_{\tilde{A}_{im}} \sum_{j=1}^{M} \sum_{i=1}^{r} \left| \sum_{m=1}^{r} \tilde{A}_{im}\, a_m^{snap}(t_j) - G_i^{snap}(t_j) \right|^2 = \min_{\tilde{A}_{im}} f(\tilde{A}_{im})\,. \tag{4.36}$$

The optimality conditions for the minimization problem (4.36) yield

$$\frac{\partial f}{\partial \tilde{A}_{kl}} = 0 \qquad \forall\, k, l = 1, \ldots, r\,. \tag{4.37}$$

The LHS of (4.37) can be calculated as follows:

$$\begin{aligned} \frac{\partial f}{\partial \tilde{A}_{kl}} &= \sum_{j=1}^{M} 2 \left(\sum_{m=1}^{r} \tilde{A}_{km}\, a_m^{snap}(t_j) - \boldsymbol{G}_k^{snap}(t_j) \right) a_l^{snap}(t_j) \\ &= 2 \sum_{j=1}^{M} \left(\sum_{m=1}^{r} \tilde{A}_{km}\, a_m^{snap}(t_j)\, a_l^{snap}(t_j) - \boldsymbol{G}_k^{snap}(t_j)\, a_l^{snap}(t_j) \right) \\ &= \sum_{m=1}^{r} \tilde{A}_{km} \sum_{j=1}^{M} \left(a_m^{snap}(t_j)\, a_l^{snap}(t_j) \right) - \sum_{j=1}^{M} \left(\boldsymbol{G}_k^{snap}(t_j)\, a_l^{snap}(t_j) \right). \end{aligned} \tag{4.38}$$

Thus, the optimality conditions (4.37) reduce to the following matrix equation:

$$\tilde{A}\, D = E\,, \tag{4.39}$$

where

$$D_{ml} := \sum_{j=1}^{M} a_m^{snap}(t_j)\, a_l^{snap}(t_j)\,, \qquad E_{kl} := \sum_{j=1}^{M} \boldsymbol{G}_k^{snap}(t_j)\, a_l^{snap}(t_j)\,. \tag{4.40}$$

Therefore, assuming that the matrix D is invertible, the matrix $\tilde{A}$ is

$$\tilde{A} = E\, D^{-1}\,. \tag{4.41}$$

The optimal coefficients $\tilde{A}_{im}^{CF-ROM}$ that solve (4.41) yield the *calibrated filtered ROM (CF-ROM)*:

$$\dot{a}_i = \sum_{m=1}^{r} \left(A_{im} + \tilde{A}_{im}^{CF-ROM} \right) a_m + \sum_{m=1}^{r} \sum_{n=1}^{r} B_{imn}\, a_n\, a_m\,. \tag{4.42}$$

The CF-ROM (4.42) construction can be summarized in the following algorithm:

Algorithm 2 CF-ROM

1. Use (4.20) to compute $a_i^{snap}(t_j),\ i = 1, \ldots, d$.
2. Use (4.40) to assemble matrix D.
3. Use (4.33) to compute $G_i^{snap}(t_j),\ i = 1, \ldots, r$. Use the computationally efficient approach in Remark 4.2.
4. Use (4.40) to assemble matrix E.
5. Solve the matrix equation (4.41).
6. The CF-ROM is given by (4.42) with the optimal coefficients $\tilde{A}_{im}^{CF-ROM}$ that solve the matrix equation (4.41) in step 5.

4.5 Numerical Results

In this section, we numerically investigate the CF-ROM (4.42) applied to the Burgers equation, which has become a common benchmark problem in ROM testing [32, 100, 140] due to its being a one-dimensional (1D) problem but yet displaying complex solution structure. Indeed, Burgers equation is used to model phenomena in statistical mechanics, vehicle traffic, adhesion in cosmology, as well as simplistic modeling of fluid dynamics [20]. In Section 4.5.1, we describe the mathematical and computational setting of the test problem. In Section 4.5.2, we construct the CF-ROM for the Burgers equation, following a procedure that is analogous to that used above for the CF-ROM construction for the 3D NSE. In Section 4.5.3, we compare the CF-ROM with the standard G-ROM, both in terms of numerical accuracy and computational efficiency. Finally, in Section 4.5.4, we compare the CF-ROM with some of the most recent ROMs, both in terms of numerical accuracy and computational efficiency.

4.5.1 Burgers Equation

For the test problem in the CF-ROM numerical investigation, we use the 1D Burgers equation as mathematical model, which was also used in [32, 100, 140]:

$$\begin{cases} u_t - \nu\, u_{xx} + u\, u_x = 0 & x \in [0,1], \\ u(x,0) = u_0(x) & x \in [0,1], \\ u(0,t) = u(1,t) = 0\,, \end{cases} \tag{4.43}$$

where $\nu = 10^{-3}$ is the diffusion parameter and $t \in [0,1]$. We use the following initial condition:

$$u_0(x) = \begin{cases} 1 & x \in \left(0, \frac{1}{2}\right], \\ 0 & x \in \left(\frac{1}{2}, 1\right). \end{cases} \tag{4.44}$$

For the direct numerical simulation (DNS) (see Fig. 4.1), we use a uniform mesh with $\Delta x = 1/1024$ and a linear FE for the spatial discretization and the backward Euler method with a time step $\Delta t = 10^{-3}$ for the time discretization. For all the ROMs, we use the forward Euler method with a time step $\Delta t = 10^{-4}$ for the time discretization. To generate the POD basis, we collect 101 snapshots in the time interval $[0,1]$.

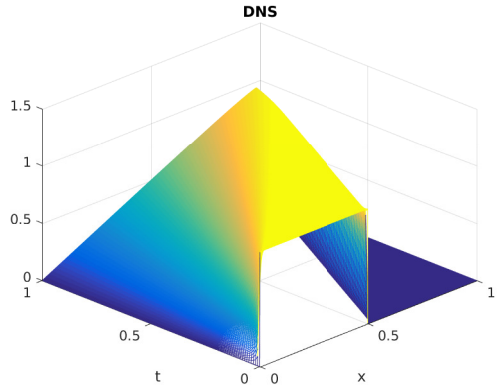


Figure 4.1: Shown above is a plot of the solution of the Burgers equation DNS.

4.5.2 CF-ROM for Burgers Equation

In Section 4.4, the entire derivation of the CF-ROM is carried out for the general 3D NSE. We now adapt the CF-ROM framework to the 1D Burgers equation (4.43), for which the F-ROM (4.9) becomes the following: $\forall\, i = 1, \ldots, r$,

$$\left(\frac{\partial u_r}{\partial t}, \varphi_i\right) + \nu \left(\frac{\partial u_r}{\partial x}, \frac{\partial \varphi_i}{\partial x} d\right) + \left(u_r \frac{\partial u_r}{\partial x}, \varphi_i\right) + \left(f(\boldsymbol{a}, X^r), \varphi_i\right) = 0\,. \tag{4.45}$$

Thus, $G_i(\boldsymbol{a})$ in (4.31) becomes the following: $\forall\, i = 1, \ldots, r$,

$$G_i(\boldsymbol{a}) := \left(\overline{u_d \frac{\partial u_d}{\partial x}}^{r} - u_r \frac{\partial u_r}{\partial x}\,,\, \varphi_i\right). \tag{4.46}$$

This, in turn, implies that (4.33) becomes: $\forall\, i = 1, \ldots, r$,

$$\begin{aligned} G_i^{snap}(t_j) \;=\; & \left(\overline{\left[\left(\sum_{k_1=1}^{d} a_{k_1}^{snap}(t_j)\varphi_{k_1}\right)\left(\sum_{k_2=1}^{d} a_{k_2}^{snap}(t_j)\frac{\partial \varphi_{k_2}}{\partial x}\right)\right.}^{r} \right. \\ & \left. - \left(\sum_{k_3=1}^{r} a_{k_3}^{snap}(t_j)\varphi_{k_3}\right)\left(\sum_{k_4=1}^{r} a_{k_4}^{snap}(t_j)\frac{\partial \varphi_{k_4}}{\partial x}\right)\right], \varphi_i\Bigg)\,. \end{aligned} \tag{4.47}$$

To compute the first term in the inner product (4.47), we use the definition of the ROM projection (4.3): $\forall\, j = 1, \ldots, r$,

$$\left(\sum_{k=1}^{r} b_k \, \varphi_k, \varphi_j \right) = \left(\left(\sum_{k_1=1}^{d} a_{k_1}^{snap}(t_j) \varphi_{k_1} \right) \left(\sum_{k_2=1}^{d} a_{k_2}^{snap}(t_j) \frac{\partial \varphi_{k_2}}{\partial x} \right), \varphi_j \right) . \tag{4.48}$$

Solving for b_k, $k = 1, \ldots, r$ in (4.50), we get

$$\overline{\left(\sum_{k_1=1}^{d} a_{k_1}^{snap}(t_j) \varphi_{k_1} \right) \left(\sum_{k_2=1}^{d} a_{k_2}^{snap}(t_j) \frac{\partial \varphi_{k_2}}{\partial x} \right)}^{r} = \sum_{k=1}^{r} b_k \, \varphi_k \, , \tag{4.49}$$

which allows us to compute $G_i^{snap}(t_j)$ in (4.47). At this stage, we have all the information needed to apply Algorithm 2 and construct the CF-ROM (4.42) for the Burgers equation.

4.5.3 CF-ROM vs G-ROM

For our first test, we compare the new CF-ROM (4.42) with the standard G-ROM. We consider the following r values: $r = 6$, $r = 10$, and $r = 15$. Table 4.1 shows the maximum errors in these ROMs (comparing to the DNS solution in the L^2 norm at each time step, and then taking the average over the time steps). We observe that the CF-ROM error is consistently lower than the G-ROM error, generally by a factor of two. To better illustrate the improvement of CF-ROM over G-ROM, we show in Figure 4.2 the respective solutions with $r = 15$.

We also investigate the efficiency of methods and give timings for the various simulations in Table 4.1. Although the CF-ROM is more accurate than the G-ROM, it is less efficient, and so a comparison is in order. In its original form, the CF-ROM uses (4.33) to compute $G_i^{snap}(t_j),\ i = 1, \ldots, r$. As mentioned in Remark 4.2, this computation utilizes $\boldsymbol{u}_d^{snap} \in \boldsymbol{X}^d$. Since $d = \mathcal{O}(1000)$ in some practical settings, this could make the CF-ROM computationally costly. Hence, following Remark 4.2, we replace $\boldsymbol{u}_d^{snap}$ in (4.33) with $\boldsymbol{u}_m^{snap}$, where $r \le m \le d$:

$$\boldsymbol{u}_d^{snap} \approx \boldsymbol{u}_m^{snap} \qquad r \le m \le d \, . \tag{4.50}$$

When m in (4.50) is low (i.e., close to r), the CF-ROM computational cost will be low, but the accuracy will also be low. On the other hand, when m in (4.50) is large (i.e., close to d), the CF-ROM accuracy will be high, but the computational cost will also be high. Thus, we need to find an m value in (4.50) that ensures a compromise between accuracy and efficiency in the CF-ROM. From Table 4.1, we observe that $m = 2r$ seems to achieve compromise between accuracy and efficiency in the CF-ROM.

Method	Proj. Space	$\|u_{DNS} - u_{ROM}\|$			CPU time (seconds)		
		r=6	r=10	r=15	r=6	r=10	r=15
G-ROM		0.2208	0.1589	0.0848	2.06	3.28	4.79
CF-ROM	X^r	0.2208	0.1589	0.0848	2.06	3.28	4.79
CF-ROM	X^{r+1}	0.1214	0.0854	0.0654	2.16	3.43	4.95
CF-ROM	X^{2r}	0.0928	0.0627	0.0446	2.75	4.42	6.67
CF-ROM	X^{3r}	0.0923	0.0632	0.0452	3.44	5.61	8.57
CF-ROM	X^d	0.0931	0.0638	0.0454	14.21	15.06	16.06

Table 4.1: G-ROM and CF-ROM errors and timings for $r = 6$, $r = 10$ and $r = 15$ for Burgers equation simulation, using varying r and ROM projection spaces.

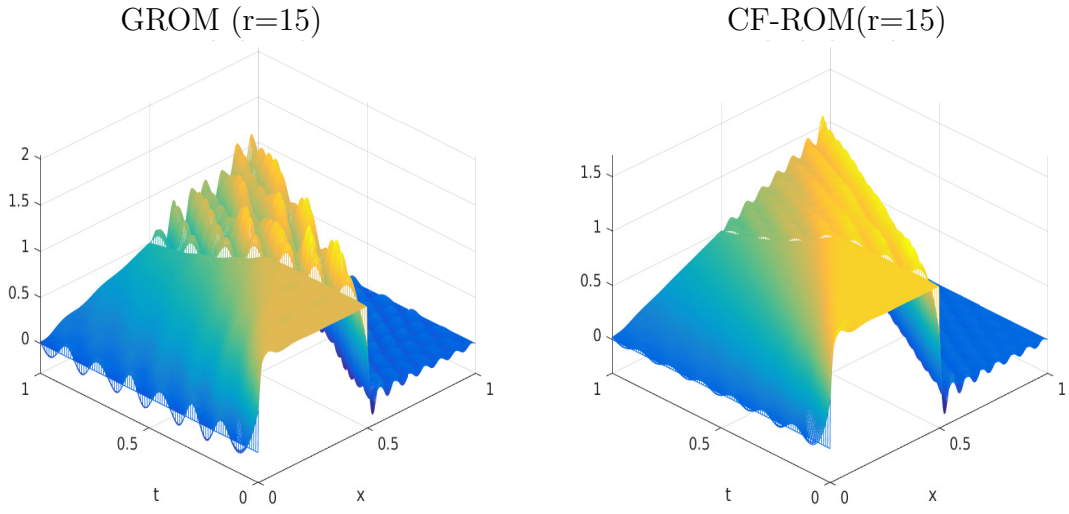


Figure 4.2: Shown above are plots for r=15 G-ROM (left) and CF-ROM (right) for the Burgers equation simulations.

4.5.4 CF-ROM vs. State-Of-The-Art ROMs

Above we have shown that the new CF-ROM is a clear improvement over the standard G-ROM, and a natural question is whether the CF-ROM is also an improvement over other, more accurate ROMs. To address this question, we consider two recently proposed ROMs for fluid flows. The first type of ROM that we consider is regularized ROMs (Reg-ROMs) [165]. We consider two Reg-ROMs: the Leray ROM (L-ROM) and the evolve-then-filter ROM (EF-ROM). We consider a differential filter in both Reg-ROMs, since this ensured the most accurate results in [165]. The second type of ROM that we consider is an LES-ROM: the approximate deconvolution ROM (AD-ROM) proposed in [168] (see chapter 3).

We compare the new CF-ROM with the AD-ROM, the L-ROM and the EF-ROM. We test all ROMs on the Burgers equation test problem considered in this section. We list the errors in Table 4.2. The CF-ROM errors are slightly lower than all the other ROM errors. We list the CPU times in Table 4.3. The CF-ROM CPU time is significantly lower than the

CPU times of the other ROMs. The results in Table 4.2 and Table 4.3 consistently show that, for this test problem, the CF-ROM is at least competitive with the other ROMs.

	L-DF	EF-ROM	AD-ROM	CF-ROM(X^{2r})
$r = 6$	0.1385	0.1005	0.1096	0.0928
$r = 10$	0.1135	0.0699	0.0633	0.0627
$r = 15$	0.1037	0.0549	0.0532	0.0446

Table 4.2: Burgers equation, ROM comparison: Errors for L-ROM-DF, EF-ROM-DF, AD-ROM and the new CF-ROM with $m = 2r$ in (4.50).

	L-DF	EF-ROM	AD-ROM	CF-ROM(X^{2r})
$r = 6$	4.12	4.25	4.44	2.27
$r = 10$	6.72	6.91	7.26	4.42
$r = 15$	9.97	10.14	10.32	6.67

Table 4.3: Burgers equation, ROM comparison: CPU times for L-ROM-DF, EF-ROM-DF, AD-ROM and the new CF-ROM with $m = 2r$ in (4.50).

4.6 Summary

In this chapter, we proposed a novel ROM framework for the numerical simulation of general nonlinear PDEs. This framework was based on explicit ROM spatial filtering and calibration. The explicit ROM spatial filtering ensured computational efficiency of the filtered ROM. The calibration procedure solved the ROM closure problem in the filtered ROM. We numerically investigated the resulting CF-ROM in the simulation of the 1D Burgers equation. First, we compared the new CF-ROM with the standard G-ROM. The CF-ROM was significantly more accurate than the G-ROM. Furthermore, the computational costs of the CF-ROM and G-ROM were similar, both costs being orders of magnitude lower than the computational cost of the full order model. We also compared the new CF-ROM with state-of-the-art LES-ROMs. The CF-ROM was as accurate as state-of-the-art LES-ROMs. However, the CF-ROM was significantly more efficient than these LES-ROMs.

Chapter 5

Leray ROM (L-ROM): Numerical Analysis

Standard ROMs generally yield spurious numerical oscillations in the simulation of convection-dominated flows. Regularized ROMs use explicit ROM spatial filtering to decrease these spurious numerical oscillations. The Leray ROM is a recently introduced regularized ROM that utilizes explicit ROM spatial filtering of the convective term in the NSE. This chapter presents the numerical analysis of the finite element discretization of the Leray ROM. Error estimates for the ROM differential filter, which is the explicit ROM spatial filter used in the Leray ROM, are proved. These ROM filtering error estimates are then used to prove error estimates for the Leray ROM. Finally, both the ROM filtering error estimates and the Leray ROM error estimates are numerically investigated in the simulation of the 2D NSE with an analytic solution.

5.1 Introduction

For convection-dominated flows, standard ROMs generally yield inaccurate results, usually in the form of spurious numerical oscillations (see, e.g., [76, 162]). To mitigate these ROM inaccuracies, several numerical stabilization techniques have been proposed over the years (see, e.g., [13, 14, 25, 94, 125, 162, 160]). *Regularized ROMs (Reg-ROMs)* are recently proposed stabilized ROMs for the numerical simulation of convection-dominated flows [88, 138, 165]. These Reg-ROMs use *explicit ROM spatial filtering* to smooth various ROM terms and thus increase the numerical stability of the resulting ROM. This idea goes back to the great Jean Leray [108], who used it in the mathematical study of the NSE. In standard CFD, this idea was used to develop regularized models for the numerical simulation of turbulent flows [74, 107]). In a ROM setting, a Reg-ROM was first used in [138] in the numerical simulation of the 1D Kuramoto-Sivashinsky equations. A different Reg-ROM was proposed in [165] for the numerical simulation of the 3D NSE. Reg-ROMs were also employed for the stabilization of ROMs in the numerical simulation of a stochastic Burgers equation [88].

Reg-ROMs were successful in the numerical simulation of convection-dominated flows. Two Reg-ROMs (the Leray ROM and the evolve-then-filter ROM) were used in the numerical simulation of a 3D flow past a circular cylinder at a Reynolds number $Re = 1000$ [165].

These two Reg-ROMs produced accurate results in which the spurious numerical oscillations of standard ROMs were significantly decreased. Despite the Reg-ROMs' success, to our knowledge there is no numerical analysis of the Reg-ROMs and the explicit ROM spatial filter used in their development. In this chapter, we take a first step in this direction and prove error estimates for the finite element discretization of (i) the *Leray ROM* [88, 138, 165], which is a Reg-ROM; and (ii) the *ROM differential filter*, which is an explicit ROM spatial filter.

The rest of the chapter is organized as follows: In Section 5.2, we present some notation and preliminaries. In Section 5.3, we present the Leray ROM. In Section 5.4, we prove error estimates for the ROM differential filter and the Leray ROM. In Section 5.5, we verify numerically the error estimates proved in Section 5.4. Finally, in Section 5.6, we draw conclusions and outline possible future research directions.

5.2 Notation and Preliminaries

We consider the numerical solution of the incompressible NSE:

$$\begin{cases} \dfrac{\partial \boldsymbol{u}}{\partial t} - \nu\Delta\boldsymbol{u} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} + \nabla p = \boldsymbol{f}, & \text{in } \Omega\times(0,T], \\ \nabla\cdot\boldsymbol{u} = 0, & \text{in } \Omega\times(0,T], \\ \boldsymbol{u} = 0, & \text{on } \partial\Omega\times(0,T], \\ \boldsymbol{u}(\boldsymbol{x},0) = \boldsymbol{u}^0(\boldsymbol{x}), & \text{in } \Omega, \end{cases} \tag{5.1}$$

where ν is the reciprocal of the Reynolds number. The following functional spaces and notations will be used in the chapter:

$$\boldsymbol{X} = \boldsymbol{H}_0^1(\Omega) = \left\{\boldsymbol{v}\in[L^2(\Omega)]^n : \nabla\boldsymbol{v}\in[L^2(\Omega)]^{n\times n} \text{ and } \boldsymbol{v}=\boldsymbol{0} \text{ on } \partial\Omega\right\},$$

$$Q = L_0^2(\Omega) = \left\{q\in L^2(\Omega) : \int_\Omega q\, d\boldsymbol{x} = 0\right\},$$

$$\boldsymbol{V} = \{\boldsymbol{v}\in\boldsymbol{X} : (\nabla\cdot\boldsymbol{v}, q) = 0, \forall\, q\in Q\}, \text{ and}$$

$$\boldsymbol{V}^h = \left\{\boldsymbol{v}_h\in\boldsymbol{X}^h : (\nabla\cdot\boldsymbol{v}_h, q_h) = 0, \forall\, q_h\in Q^h\right\},$$

where $\boldsymbol{X}^h\subset\boldsymbol{X}$ and $Q^h\subset Q$ are the FE spaces of the velocity and pressure, respectively, and h is the quasi-uniform mesh size. We consider the div-stable pair of FE spaces $(\boldsymbol{X}^h/Q^h) = (\mathbb{P}^m/\mathbb{P}^{m-1})$, $m\geq 2$ [106]. We emphasize, however, that our analysis extends to more general FE spaces.

Let $\mathcal{H}$ be a real Hilbert space endowed with inner product $(\cdot,\cdot)_{\mathcal{H}}$ and norm $\|\cdot\|_{\mathcal{H}}$. Let the trilinear form $b^*(\cdot,\cdot,\cdot)$ be defined as

$$b^*(\boldsymbol{u},\boldsymbol{v},\boldsymbol{w}) = \frac{1}{2}\left[((\boldsymbol{u}\cdot\nabla)\boldsymbol{v},\boldsymbol{w}) - ((\boldsymbol{u}\cdot\nabla)\boldsymbol{w},\boldsymbol{v})\right].$$

Definition 5.1 (ROM Differential Filter)**.** *$\forall\boldsymbol{v}\in\boldsymbol{X}$, let $\overline{\boldsymbol{v}}^r$ be the unique element of $\mathbf{X}^r$ such that*

$$\delta^2\,(\nabla\overline{\boldsymbol{v}}^r, \nabla\boldsymbol{v}_r) + (\overline{\boldsymbol{v}}^r, \boldsymbol{v}_r) = (\boldsymbol{v},\boldsymbol{v}_r) \quad \forall\boldsymbol{v}_r\in\mathbf{X}^r. \tag{5.2}$$

Lemma 5.2 (see Lemma 13, Lemma 14 and Lemma 18 in [106]). *For any functions $\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w} \in \boldsymbol{X}$, the skew-symmetric trilinear form $b^*(\cdot,\cdot,\cdot)$ satisfies*

$$b^*(\boldsymbol{u},\boldsymbol{v},\boldsymbol{v}) = 0, \tag{5.3}$$

$$b^*(\boldsymbol{u},\boldsymbol{v},\boldsymbol{w}) \le C\|\nabla\boldsymbol{u}\|\|\nabla\boldsymbol{v}\|\|\nabla\boldsymbol{w}\|, \tag{5.4}$$

and a sharper bound

$$b^*(\boldsymbol{u},\boldsymbol{v},\boldsymbol{w}) \le C\sqrt{\|\boldsymbol{u}\|\|\nabla\boldsymbol{u}\|}\|\nabla\boldsymbol{v}\|\|\nabla\boldsymbol{w}\|. \tag{5.5}$$

The weak formulation of the NSE (5.1) reads: Find $\boldsymbol{u} \in \boldsymbol{X}$ and $p \in Q$ such that

$$\begin{cases} \left(\dfrac{\partial \boldsymbol{u}}{\partial t}, \boldsymbol{v}\right) + \nu(\nabla\boldsymbol{u}, \nabla\boldsymbol{v}) + b^*(\boldsymbol{u},\boldsymbol{u},\boldsymbol{v}) - (p, \nabla\cdot\boldsymbol{v}) = (\boldsymbol{f},\boldsymbol{v}), & \forall\, \boldsymbol{v} \in \boldsymbol{X}, \\ (\nabla\cdot\boldsymbol{u}, q) = 0, & \forall\, q \in Q. \end{cases} \tag{5.6}$$

To ensure the uniqueness of the solution to (5.6), we make the following regularity assumptions (see Definition 29, Proposition 15, and Remark 10 in [106]):

Assumption 5.2.1. *In (5.1), we assume that $\boldsymbol{f} \in L^2(0,T;\boldsymbol{L}^2(\Omega))$, $\boldsymbol{u}^0 \in \boldsymbol{V}$, $\boldsymbol{u} \in L^2(0,T;\boldsymbol{X}) \bigcap L^\infty(0,T;\boldsymbol{L}^2(\Omega))$, $\nabla\boldsymbol{u} \in (L^4(0,T;L^2(\Omega)))^{n\times n}$, $\boldsymbol{u}_t \in L^2(0,T;\boldsymbol{X}^*)$, and $p \in L^2(0,T;Q)$.*

For any positive integer M, consider the time instances $t_k = k\,\Delta t$, $k = 0,\ldots,M$, where $\Delta t = T/M$. Denote the solution of (5.6) at time t_k to be $\boldsymbol{u}^k = \boldsymbol{u}(t_k)$ and the force at t_k to be $\boldsymbol{f}^k = \boldsymbol{f}(t_k)$, respectively. We make the following assumption, which will be used in Theorem 5.13:

Assumption 5.2.2. *We assume that the solution of (5.6) satisfies the following stability estimate for all integers $\widetilde{M}$ such that $1 \le \widetilde{M} \le M$:*

$$\|\boldsymbol{u}^{\widetilde{M}}\|^2 + \Delta t \sum_{k=0}^{\widetilde{M}-1} \|\nabla\boldsymbol{u}^{k+1}\|^2 \le C\,, \tag{5.7}$$

where C is a constant that does not depend on Δt, but can depend on the initial data.

For $k = 0,\ldots,M$, denote the FE approximate solution of (5.8) at $t_k = k\,\Delta t$ to be $\boldsymbol{u}_h^k = \boldsymbol{u}_h(t_k)$. The FE semidiscretization of (5.6) can be written as follows: Find $\boldsymbol{u}_h \in \boldsymbol{V}^h$ such that

$$\left(\frac{\partial \boldsymbol{u}_h}{\partial t}, \boldsymbol{v}_h\right) + \nu(\nabla\boldsymbol{u}_h, \nabla\boldsymbol{v}_h) + b^*(\boldsymbol{u}_h,\boldsymbol{u}_h,\boldsymbol{v}_h) = (\boldsymbol{f},\boldsymbol{v}_h), \quad \forall\, \boldsymbol{v}_h \in \boldsymbol{V}^h \tag{5.8}$$

and $\boldsymbol{u}_h(\cdot,0) = \boldsymbol{u}_h^0 \in \boldsymbol{V}^h$.

Assumption 5.2.3 (Finite Element Error). *We assume that the FE approximation $\boldsymbol{u}_h$ of the full discretization of (5.8) satisfies the following error estimate:*

$$\|\boldsymbol{u} - \boldsymbol{u}_h\| + h\|\nabla(\boldsymbol{u} - \boldsymbol{u}_h)\| \le C(h^{m+1} + \Delta t). \tag{5.9}$$

We also assume the following standard approximation property:

$$\inf_{q_h \in Q^h} \|p - q_h\| \le Ch^m. \tag{5.10}$$

The constant C in (5.9)–(5.10) is a constant that does not depend on $h, m, \Delta t$, but can depend on the initial data.

5.3 The Leray ROM (L-ROM)

In this section, we present the Leray ROM, which we will analyze in Section 5.4. To this end, we start with the *Galerkin ROM (G-ROM)* for the NSE: Find $\boldsymbol{u}_r \in \boldsymbol{X}^r$ such that

$$\left(\frac{\partial \boldsymbol{u}_r}{\partial t}, \boldsymbol{\varphi}\right) + \nu(\nabla \boldsymbol{u}_r, \nabla \boldsymbol{\varphi}) + b^*(\boldsymbol{u}_r, \boldsymbol{u}_r, \boldsymbol{\varphi}) = (\mathbf{f}, \boldsymbol{\varphi}), \quad \forall\, \boldsymbol{\varphi} \in \boldsymbol{X}^r \tag{5.11}$$

and $\boldsymbol{u}_r(\cdot, 0) \in \boldsymbol{X}^r$. The error analysis of the spatial and temporal discretizations of the G-ROM (5.11) was considered in [46, 90, 97, 100, 101, 111, 143]. Despite its appealing computational efficiency, the G-ROM (5.11) has generally been limited to laminar flows. To overcome this restriction, we consider the *Leray reduced order model (L-ROM)* [138, 165], which is a regularized ROM: Find $\boldsymbol{u}_r \in \boldsymbol{X}^r$ such that

$$\left(\frac{\partial \boldsymbol{u}_r}{\partial t}, \boldsymbol{\varphi}\right) + \nu(\nabla \boldsymbol{u}_r, \nabla \boldsymbol{\varphi}) + b^*(\overline{\boldsymbol{u}_r}, \boldsymbol{u}_r, \boldsymbol{\varphi}) = (\boldsymbol{f}, \boldsymbol{\varphi}), \quad \forall\, \boldsymbol{\varphi} \in \boldsymbol{X}^r, \tag{5.12}$$

where $\overline{\boldsymbol{u}}_r$ is the ROM velocity filtered with the ROM differential filter (DF) and the initial condition is given by the L^2 projection of $\boldsymbol{u}^0$ on $\boldsymbol{X}^r$: $\boldsymbol{u}_r(\cdot, 0) := \sum_{j=1}^r (\boldsymbol{u}^0, \boldsymbol{\varphi}_j)\boldsymbol{\varphi}_j$.

We consider the full discretization of (5.12): We use the backward Euler method with a time step Δt for the time integration and the FE space $\mathbb{P}^m$ with $m \geq 2$ and a mesh size h for the spatial discretization. For $k = 0, \ldots, M$, we denote the approximation solution of (5.12) at $t_k = k\Delta t$ to be $\boldsymbol{u}_r^k = \boldsymbol{u}_{h,r}(t_k)$ and the force at t_k to be $\boldsymbol{f}^k = \boldsymbol{f}(t_k)$, respectively. Note that we have dropped the subscript "h" in $\boldsymbol{u}_r^k$ for clarity of notation. The discretized L-ROM reads: Find $\boldsymbol{u}_r^k \in \boldsymbol{X}^r$ such that, $\forall\, \boldsymbol{\varphi} \in \boldsymbol{X}^r$, $\forall\, k = 0, \ldots, M-1$,

$$\left(\frac{\boldsymbol{u}_r^{k+1} - \boldsymbol{u}_r^k}{\Delta t}, \boldsymbol{\varphi}\right) + \nu(\nabla \boldsymbol{u}_r^{k+1}, \nabla \boldsymbol{\varphi}) + b^*(\overline{\boldsymbol{u}_r^{k+1}}^r, \boldsymbol{u}_r^{k+1}, \boldsymbol{\varphi}) = (\boldsymbol{f}^{k+1}, \boldsymbol{\varphi}), \tag{5.13}$$

where the initial condition is $\mathbf{u}_r^0 = \sum_{j=1}^r (\mathbf{u}^0, \boldsymbol{\varphi}_j)\, \boldsymbol{\varphi}_j$.

5.4 Error Analysis

In this section, we present the error analysis for the L-ROM discretization (5.13). We take the FE solutions $\boldsymbol{u}_h(\cdot, t_i)$, $i = 0, \ldots, M$ as snapshots and choose $\mathcal{H} = \boldsymbol{L}^2$ in the POD generation. The error source includes three main components: the spatial FE discretization error, the temporal discretization error, and the POD truncation error. We derive the error estimate in three steps: First, we gather some necessary assumptions and preliminary results in Section 5.4.1. Then, we prove a ROM filtering error estimate in Section 5.4.2. Finally, we prove an L-ROM error estimate in Section 5.4.3.

5.4.1 Preliminaries

Definition 5.3 (Generic Constant C)**.** *In what follows, C and C_j, where j is a positive integer, will denote generic constants that do not depend on $\delta, r, h, \Delta t, m, \boldsymbol{\varphi}_j, \lambda_j$, but can depend on $\nu, \boldsymbol{u}_0, \boldsymbol{f}, \boldsymbol{u}_r^0, n, T$.*

Definition 5.4 (ROM Laplacian)**.** *Let*

$$\Delta_r : \boldsymbol{X} \to \mathbf{X}^r \tag{5.14}$$

such that, $\forall \boldsymbol{v} \in \boldsymbol{X}, \Delta_r \boldsymbol{v}$ *is the unique element of* $\mathbf{X}^r$ *such that*

$$(\Delta_r \boldsymbol{v}, \boldsymbol{v}_r) = -(\nabla \boldsymbol{v}, \nabla \boldsymbol{v}_r) \quad \forall \boldsymbol{v}_r \in \mathbf{X}^r . \tag{5.15}$$

We list a POD inverse estimate, which will be used in what follows. Let $S_r \in \mathbb{R}^{r\times r}$ with $(S_r)_{ij} = (\nabla\varphi_j, \nabla\varphi_i)_{L^2}$ be the POD stiffness matrix. Let $\|\cdot\|_2$ denote the matrix 2-norm.

Lemma 5.5 (POD Inverse Estimates)**.** *For all* $\boldsymbol{v}_r \in \boldsymbol{X}^r$*, the following POD inverse estimate holds:*

$$\|\nabla \boldsymbol{v}_r\|_{L^2} \;\leq\; C_{inv}^{\nabla}(r)\, \|\boldsymbol{v}_r\|_{L^2} , \tag{5.16}$$

where $C_{inv}^{\nabla}(r) := \sqrt{\|S_r\|_2}$.

The inverse estimate (5.16) was proved in Lemma 2 and Remark 2 in [100] and was numerically investigated in Remark 3.3 in [90] and in Remark 3.2 in [76].

Definition 5.6 (ROM L^2 Projection)**.** *Let*

$$P_r : \boldsymbol{L}^2 \to \mathbf{X}^r \tag{5.17}$$

such that, $\forall\, \boldsymbol{v} \in \boldsymbol{L}^2, P_r(\boldsymbol{v})$ *is the unique element of* $\mathbf{X}^r$ *such that*

$$(P_r(\boldsymbol{v}), \boldsymbol{v}_r) = (\boldsymbol{v}, \boldsymbol{v}_r) \quad \forall \boldsymbol{v}_r \in \mathbf{X}^r . \tag{5.18}$$

Proposition 5.7 (L^2 Stability of ROM L^2 Projection)**.**

$$\|P_r(\boldsymbol{v})\| \;\leq\; \|\boldsymbol{v}\| \qquad \forall\, \boldsymbol{v} \in \boldsymbol{L}^2 . \tag{5.19}$$

Proof. By choosing $\boldsymbol{v}_r := P_r(\boldsymbol{v})$ in (5.18) and using the Cauchy-Schwarz inequality, we can prove (5.19). □

The following error estimate was proved in Lemma 3.3 in [91]:

Lemma 5.8. *For any* $\boldsymbol{u}^k \in \boldsymbol{X}$*, its* L^2 *projection,* $\boldsymbol{w}_r^k = P_r(\boldsymbol{u}^k)$*, satisfies the following error estimates:*

$$\frac{1}{M+1}\sum_{k=0}^{M} \left\|\boldsymbol{u}^k - \boldsymbol{w}_r^k\right\|^2 \leq C\left(h^{2m+2} + \Delta t^2 + \sum_{j=r+1}^{d} \lambda_j\right), \tag{5.20}$$

$$\frac{1}{M+1}\sum_{k=0}^{M} \left\|\nabla(\boldsymbol{u}^k - \boldsymbol{w}_r^k)\right\|^2 \leq C\Bigg(h^{2m} + \|S_r\|_2 h^{2m+2} + (1+\|S_r\|_2)\Delta t^2 + \sum_{j=r+1}^{d} \|\boldsymbol{\varphi}_j\|_1^2\, \lambda_j\Bigg). \tag{5.21}$$

We assume the following estimates, which were also assumed in [90]:

Assumption 5.4.1. *For any $\boldsymbol{u}^k \in \boldsymbol{X}$, its L^2 projection, $\boldsymbol{w}_r^k = P_r(\boldsymbol{u}^k)$, satisfies the following error estimates:*

$$\left\|\boldsymbol{u}^k - \boldsymbol{w}_r^k\right\| \leq C \left(h^{m+1} + \Delta t + \sqrt{\sum_{j=r+1}^{d} \lambda_j}\right), \tag{5.22}$$

$$\left\|\nabla\left(\boldsymbol{u}^k - \boldsymbol{w}_r^k\right)\right\| \leq C \left(h^m + \sqrt{\|S_r\|_2} h^{m+1} + \sqrt{1 + \|S_r\|_2}\Delta t + \sqrt{\sum_{j=r+1}^{d} \|\boldsymbol{\varphi}_j\|_1^2 \lambda_j}\right). \tag{5.23}$$

5.4.2 ROM Filtering Error Estimates

In this section, we present theoretical results for the DF earlier definition, which was the essential tool that we used in developing the Leray ROM (5.12). The main result in this section is the estimate for the ROM filtering error in Lemma 5.9. To our knowledge, this is the first estimate for the ROM filtering error. This estimate is an extension of the FE filtering error estimates proved in [61, 62, 105]. This ROM filtering error estimate is important for the L-ROM error analysis in Section 5.4.3, since we use it to treat the nonlinear term in (5.61).

Lemma 5.9 (ROM Filtering Error Estimates)**.** *For $\boldsymbol{u}^k \in \boldsymbol{X}$ and $\Delta\boldsymbol{u}^k \in \boldsymbol{L}^2$,*

$$\begin{aligned}
&\delta^2 \|\nabla(\boldsymbol{u}^k - \overline{\boldsymbol{u}^k}^r)\|^2 + \|\boldsymbol{u}^k - \overline{\boldsymbol{u}^k}^r\|^2 \\
&\leq C \left(h^{2m+2} + \Delta t^2 + \sum_{j=r+1}^{d} \lambda_j\right) + C\,\delta^4 \|\Delta\boldsymbol{u}^k\|^2 \\
&+ C\,\delta^2 \left(h^{2m} + \|S_r\|_2\, h^{2m+2} + (1 + \|S_r\|_2)\,\Delta t^2 + \sum_{j=r+1}^{d} \|\boldsymbol{\varphi}_j\|_1^2 \lambda_j\right).
\end{aligned} \tag{5.24}$$

Proof. Using the definition of the DF, we have

$$\delta^2 \left(\nabla\overline{\boldsymbol{u}^k}^r, \nabla\boldsymbol{v}_r\right) + \left(\overline{\boldsymbol{u}^k}^r, \boldsymbol{v}_r\right) = \left(\boldsymbol{u}^k, \boldsymbol{v}_r\right) \quad \forall \boldsymbol{v}_r \in \mathbf{X}^r. \tag{5.25}$$

Since $\Delta\boldsymbol{u}^k \in \boldsymbol{L}^2$ (by hypothesis), we also have

$$\delta^2 \left(\nabla\boldsymbol{u}^k, \nabla\boldsymbol{v}_r\right) + \left(\boldsymbol{u}^k, \boldsymbol{v}_r\right) = -\delta^2 \left(\Delta\boldsymbol{u}^k, \boldsymbol{v}_r\right) + \left(\boldsymbol{u}^k, \boldsymbol{v}_r\right) \quad \forall \boldsymbol{v}_r \in \mathbf{X}^r. \tag{5.26}$$

Subtracting (5.25) from (5.26), we get

$$\delta^2 \left(\nabla(\boldsymbol{u}^k - \overline{\boldsymbol{u}^k}^r), \nabla\boldsymbol{v}_r\right) + \left(\boldsymbol{u}^k - \overline{\boldsymbol{u}^k}^r, \boldsymbol{v}_r\right) = -\delta^2 \left(\Delta\boldsymbol{u}^k, \boldsymbol{v}_r\right) \quad \forall \boldsymbol{v}_r \in \mathbf{X}^r. \tag{5.27}$$

We decompose the error $\boldsymbol{e} := \boldsymbol{u}^k - \overline{\boldsymbol{u}^k}^r$ as follows:

$$\boldsymbol{e} = \left(\boldsymbol{u}^k - \boldsymbol{w}_r\right) - \left(\overline{\boldsymbol{u}^k}^r - \boldsymbol{w}_r\right) := \boldsymbol{\eta} - \boldsymbol{\Phi}_r \,, \tag{5.28}$$

where $\boldsymbol{w}_r \in \mathbf{X}^r$ is arbitrary. Using (5.28) in (5.27), we get: $\forall \boldsymbol{v}_r \in \mathbf{X}^r$

$$\delta^2 \, (\nabla \boldsymbol{\Phi}_r, \nabla \boldsymbol{v}_r) + (\boldsymbol{\Phi}_r, \boldsymbol{v}_r) = \delta^2 \, (\nabla \boldsymbol{\eta}, \nabla \boldsymbol{v}_r) + (\boldsymbol{\eta}, \boldsymbol{v}_r) + \delta^2 \left(\Delta \boldsymbol{u}^k, \boldsymbol{v}_r\right) . \tag{5.29}$$

To prove (5.24), we let $\boldsymbol{v}_r := \boldsymbol{\Phi}_r \in \mathbf{X}^r$ in (5.29) and then use the Cauchy-Schwarz and Young inequalities:

$$\begin{aligned}
&\delta^2 \, \|\nabla \boldsymbol{\Phi}_r\|^2 + \|\boldsymbol{\Phi}_r\|^2 = \delta^2 \, (\nabla \boldsymbol{\eta}, \nabla \boldsymbol{\Phi}_r) + (\boldsymbol{\eta}, \boldsymbol{\Phi}_r) + \delta^2 \left(\Delta \boldsymbol{u}^k, \boldsymbol{\Phi}_r\right) \\
&\leq \delta^2 \, \|\nabla \boldsymbol{\eta}\| \, \|\nabla \boldsymbol{\Phi}_r\| + \|\boldsymbol{\eta}\| \, \|\boldsymbol{\Phi}_r\| + \delta^2 \, \|\Delta \boldsymbol{u}^k\| \, \|\boldsymbol{\Phi}_r\| \\
&\leq \delta^2 \left(\frac{\|\nabla \boldsymbol{\eta}\|^2}{2} + \frac{\|\nabla \boldsymbol{\Phi}_r\|^2}{2}\right) + \left(\|\boldsymbol{\eta}\|^2 + \frac{\|\boldsymbol{\Phi}_r\|^2}{4}\right) + \left(\delta^4 \, \|\Delta \boldsymbol{u}^k\|^2 + \frac{\|\boldsymbol{\Phi}_r\|^2}{4}\right) .
\end{aligned} \tag{5.30}$$

Rearranging (5.30) yields

$$\frac{\delta^2}{2} \, \|\nabla \boldsymbol{\Phi}_r\|^2 + \frac{1}{2} \, \|\boldsymbol{\Phi}_r\|^2 \leq \frac{\delta^2}{2} \, \|\nabla \boldsymbol{\eta}\|^2 + \|\boldsymbol{\eta}\|^2 + \delta^4 \, \|\Delta \boldsymbol{u}^k\|^2 \,. \tag{5.31}$$

Using (5.31) and the triangle inequality yields

$$\begin{aligned}
&\delta^2 \, \|\nabla (\boldsymbol{u}^k - \overline{\boldsymbol{u}^k}^r)\|^2 + \|\boldsymbol{u}^k - \overline{\boldsymbol{u}^k}^r\|^2 \\
\leq \;& C \inf_{\boldsymbol{w}_r \in \mathbf{X}^r} \left(\delta^2 \, \|\nabla (\boldsymbol{u}^k - \boldsymbol{w}_r)\|^2 + \|\boldsymbol{u}^k - \boldsymbol{w}_r\|^2\right) + C \, \delta^4 \, \|\Delta \boldsymbol{u}^k\|^2 \,.
\end{aligned} \tag{5.32}$$

Using Assumption 5.4.1, we get

$$\inf_{\boldsymbol{w}_r \in \mathbf{X}^r} \|\boldsymbol{u}^k - \boldsymbol{w}_r\|^2 \leq \|\boldsymbol{u}^k - P_r(\boldsymbol{u}^k)\|^2 \leq C \left(h^{2m+2} + \Delta t^2 + \sum_{j=r+1}^{d} \lambda_j\right) . \tag{5.33}$$

$$\begin{aligned}
&\inf_{\boldsymbol{w}_r \in \mathbf{X}^r} \|\nabla (\boldsymbol{u}^k - \boldsymbol{w}_r)\|^2 \leq \|\nabla (\boldsymbol{u}^k - P_r(\boldsymbol{u}^k))\|^2 \\
&\qquad \leq \; C \left(h^{2m} + \|S_r\|_2 \, h^{2m+2} + (1 + \|S_r\|_2) \, \Delta t^2 + \sum_{j=r+1}^{d} \|\boldsymbol{\varphi}_j\|_1^2 \, \lambda_j\right) .
\end{aligned} \tag{5.34}$$

Plugging (5.33)-(5.34) into (5.32) proves (5.24).

□

Remark 5.10. *Lemma 5.9 extends Lemma 2.12 in [105] from the FE setting to the ROM setting. We could have extended Lemma 2.4 in [62] instead of Lemma 2.12 in [105] since the former yields better δ scalings of the H^1 seminorm of the filtering error. We emphasize, however, that the proof of Lemma 2.4 in [62] uses the H^1 stability of the L^2 projection [17, 63]. To our knowledge, the H^1 stability of the L^2 projection has not been yet proven in a ROM setting. Thus, we decided to extend to the ROM setting Lemma 2.12 in [105], which does not rely on the H^1 stability of the ROM L^2 projection.*

In the following lemma, we prove the stability of the ROM filtered variables, which will be used to prove Theorem 5.13. This lemma extends Lemma 2.11 in [105] (see also Lemma 2.3 in [62]) from the FE case to the ROM case.

Lemma 5.11 (Stability of ROM Filtered Variables)**.** *For $\boldsymbol{v} \in \boldsymbol{X}$, we have*

$$\begin{aligned} \|\overline{\boldsymbol{v}}^r\| &\leq \|\boldsymbol{v}\| && (5.35)\\ \|\nabla\overline{\boldsymbol{v}}^r\| &\leq \sqrt{\|S_r\|_2}\,\|\boldsymbol{v}\|\,. && (5.36) \end{aligned}$$

For $\boldsymbol{v} \in \mathbf{X}^r$, we have

$$\|\nabla\overline{\boldsymbol{v}}^r\| \leq \|\nabla\boldsymbol{v}\|\,. \tag{5.37}$$

Proof. To prove (5.35), we let $\boldsymbol{v}_r = \overline{\boldsymbol{v}}^r$ in (5.2):

$$\delta^2\,\|\nabla\overline{\boldsymbol{v}}^r\|^2 + \|\overline{\boldsymbol{v}}^r\|^2 = (\boldsymbol{v}, \overline{\boldsymbol{v}}^r)\ . \tag{5.38}$$

Applying the Cauchy-Schwarz inequality to the RHS of (5.38), we get (5.35).

To prove (5.36), we use the POD inverse estimate (5.16) and (5.35):

$$\|\nabla\overline{\boldsymbol{v}}^r\| \overset{(5.16)}{\leq} \sqrt{\|S_r\|_2}\,\|\overline{\boldsymbol{v}}^r\| \overset{(5.35)}{\leq} \sqrt{\|S_r\|_2}\,\|\boldsymbol{v}\|\,. \tag{5.39}$$

Finally, to prove (5.37), we let $\boldsymbol{v}_r = \Delta_r\overline{\boldsymbol{v}}^r$ in (5.2):

$$\delta^2\left(\nabla\overline{\boldsymbol{v}}^r, \nabla(\Delta_r\overline{\boldsymbol{v}}^r)\right) + \left(\overline{\boldsymbol{v}}^r, \Delta_r\overline{\boldsymbol{v}}^r\right) = \left(\boldsymbol{v}, \Delta_r\overline{\boldsymbol{v}}^r\right). \tag{5.40}$$

Using the definition of the ROM Laplacian (5.15) in (5.40), we get

$$-\delta^2\left(\Delta_r\overline{\boldsymbol{v}}^r, \Delta_r\overline{\boldsymbol{v}}^r\right) - \left(\nabla\overline{\boldsymbol{v}}^r, \nabla\overline{\boldsymbol{v}}^r\right) = -\left(\nabla\boldsymbol{v}, \nabla\overline{\boldsymbol{v}}^r\right). \tag{5.41}$$

Applying the Cauchy-Schwarz inequality to the RHS of (5.41), we get (5.37). □

5.4.3 Leray ROM Error Estimates

In this section, we prove a stability estimate (Lemma 5.12) and an error estimate (Theorem 5.13) for the L-ROM (5.13).

Lemma 5.12. *The solution of* (5.13) *satisfies the following bound:*

$$\left\|\boldsymbol{u}_r^{\widetilde{M}}\right\|^2 + \Delta t\sum_{k=0}^{\widetilde{M}-1}\left\|\nabla\boldsymbol{u}_r^{k+1}\right\|^2 \leq C \qquad \forall\, 1 \leq \widetilde{M} \leq M-1\,. \tag{5.42}$$

Proof. Choosing $\boldsymbol{\varphi} := \boldsymbol{u}_r^{k+1}$ in (5.13) and noting that $b^*(\overline{\boldsymbol{u}_r^{k+1}}^r, \boldsymbol{u}_r^{k+1}, \boldsymbol{u}_r^{k+1}) = 0$ by (5.3), we obtain

$$\left(\boldsymbol{u}_r^{k+1} - \boldsymbol{u}_r^k, \boldsymbol{u}_r^{k+1}\right) + \nu\Delta t \left(\nabla \boldsymbol{u}_r^{k+1}, \nabla \boldsymbol{u}_r^{k+1}\right) = \Delta t \left(\boldsymbol{f}^{k+1}, \boldsymbol{u}_r^{k+1}\right). \tag{5.43}$$

Using the Cauchy-Schwarz and Young inequalities yields

$$\frac{1}{2}\left\|\boldsymbol{u}_r^{k+1}\right\|^2 - \frac{1}{2}\left\|\boldsymbol{u}_r^{k}\right\|^2 + \nu\Delta t \left\|\nabla \boldsymbol{u}_r^{k+1}\right\|^2 \leq \Delta t \left(\boldsymbol{f}^{k+1}, \boldsymbol{u}_r^{k+1}\right). \tag{5.44}$$

Applying the Cauchy-Schwarz and Young inequalities in (5.44), we get

$$\frac{1}{2}\left\|\boldsymbol{u}_r^{k+1}\right\|^2 - \frac{1}{2}\left\|\boldsymbol{u}_r^{k}\right\|^2 + \nu\Delta t \left\|\nabla \boldsymbol{u}_r^{k+1}\right\|^2 \leq \frac{\Delta t}{2\nu}\left\|\boldsymbol{f}^{k+1}\right\|_{-1}^2 + \frac{\nu\Delta t}{2}\left\|\nabla \boldsymbol{u}_r^{k+1}\right\|^2. \tag{5.45}$$

The stability estimate (5.42) follows by summing (5.45) from 0 to $\widetilde{M} - 1$. □

Theorem 5.13. *Under the regularity assumption of the exact solution (Assumption 5.2.1 and Assumption 5.2.2), the assumption on the FE approximation (Assumption 5.2.3), and the assumption on the ROM projection error (Assumption 5.4.1), the solution of the L-ROM* (5.13) *satisfies the following error estimate: There exists $\Delta t^* > 0$ such that the inequality*

$$\begin{aligned} &\left\|\mathbf{u}^M - \mathbf{u}_r^M\right\|^2 + \Delta t \sum_{k=0}^{M-1} \left\|\nabla\left(\mathbf{u}^{k+1} - \mathbf{u}_r^{k+1}\right)\right\|^2 \\ \leq\ & C\,\mathcal{F}\left(\delta, h, \Delta t, \|S_r\|_2, \{\lambda_j\}_{j=r+1}^d, \{\|\boldsymbol{\varphi}_j\|_1\}_{j=r+1}^d\right) \end{aligned} \tag{5.46}$$

holds for all $\Delta t < \Delta t^*$, where

$$\begin{aligned} &\mathcal{F}\left(\delta, h, \Delta t, \|S_r\|_2, \{\lambda_j\}_{j=r+1}^d, \{\|\boldsymbol{\varphi}_j\|_1\}_{j=r+1}^d\right) \\ &= \left(h^{2m} + \|S_r\|_2 h^{2m+2} + (1 + \|S_r\|_2)\,\Delta t^2 + \sum_{j=r+1}^d \|\boldsymbol{\varphi}_j\|_1^2\,\lambda_j\right) \\ &+ \|S_r\|_2^{1/2}\left(h^{2m+2} + \Delta t^2 + \sum_{j=r+1}^d \lambda_j\right) + \frac{1}{\delta}\left(h^{2m+2} + \Delta t^2 + \sum_{j=r+1}^d \lambda_j\right) \\ &+ \delta\left(h^{2m} + \|S_r\|_2\, h^{2m+2} + (1 + \|S_r\|_2)\,\Delta t^2 + \sum_{j=r+1}^d \|\boldsymbol{\varphi}_j\|_1^2\,\lambda_j\right) + \delta^3. \end{aligned} \tag{5.47}$$

Proof. We start by splitting the error into two terms:

$$\boldsymbol{u}^{k+1} - \boldsymbol{u}_r^{k+1} = \left(\boldsymbol{u}^{k+1} - \boldsymbol{w}_r^{k+1}\right) - \left(\boldsymbol{u}_r^{k+1} - \boldsymbol{w}_r^{k+1}\right) = \boldsymbol{\eta}^{k+1} - \boldsymbol{\Phi}_r^{k+1}, \tag{5.48}$$

where

$$\boldsymbol{w}_r^{k+1} := P_r(\boldsymbol{u}^{k+1}). \tag{5.49}$$

The first term in (5.48), $\boldsymbol{\eta}^{k+1} = \boldsymbol{u}^{k+1} - \boldsymbol{w}_r^{k+1}$, represents the difference between $\boldsymbol{u}^{k+1}$ and its L^2 projection on $\boldsymbol{X}^r$, which has been bounded in Lemma 5.8. The second term, $\boldsymbol{\Phi}_r^{k+1}$, is the

remainder. Next, we construct the error equation. We first evaluate the weak formulation of the NSE (5.6) at $t = t^{k+1}$ and let $\boldsymbol{v} = \boldsymbol{\Phi}_r^{k+1}$, then subtract the L-ROM (5.13) from it. We obtain

$$\begin{aligned}&\left(\boldsymbol{u}_t^{k+1}, \boldsymbol{\Phi}_r^{k+1}\right) - \left(\frac{\boldsymbol{u}_r^{k+1} - \boldsymbol{u}_r^k}{\Delta t}, \boldsymbol{\Phi}_r^{k+1}\right) + \nu\left(\nabla \boldsymbol{u}^{k+1} - \nabla \boldsymbol{u}_r^{k+1}, \nabla \boldsymbol{\Phi}_r^{k+1}\right)\\ &+b^*\left(\boldsymbol{u}^{k+1}, \boldsymbol{u}^{k+1}, \boldsymbol{\Phi}_r^{k+1}\right) - b^*\left(\overline{\boldsymbol{u}_r^{k+1}}^r, \boldsymbol{u}_r^{k+1}, \boldsymbol{\Phi}_r^{k+1}\right) - \left(p, \nabla \cdot \boldsymbol{\Phi}_r^{k+1}\right) = 0. \end{aligned} \tag{5.50}$$

By subtracting and adding the difference quotient term, $\left(\frac{\boldsymbol{u}^{k+1}-\boldsymbol{u}^k}{\Delta t}, \boldsymbol{\Phi}_r^{k+1}\right)$, in (5.50), and applying the decomposition (5.48), we have

$$\begin{aligned}&\left(\boldsymbol{u}_t^{k+1} - \frac{\boldsymbol{u}^{k+1} - \boldsymbol{u}^k}{\Delta t}, \boldsymbol{\Phi}_r^{k+1}\right) + \frac{1}{\Delta t}\left(\boldsymbol{\eta}^{k+1} - \boldsymbol{\Phi}_r^{k+1}, \boldsymbol{\Phi}_r^{k+1}\right)\\ &-\frac{1}{\Delta t}\left(\boldsymbol{\eta}^k - \boldsymbol{\Phi}_r^k, \boldsymbol{\Phi}_r^{k+1}\right) + \nu\left(\nabla\left(\boldsymbol{\eta}^{k+1} - \boldsymbol{\Phi}_r^{k+1}\right), \nabla \boldsymbol{\Phi}_r^{k+1}\right)\\ &+b^*\left(\boldsymbol{u}^{k+1}, \boldsymbol{u}^{k+1}, \boldsymbol{\Phi}_r^{k+1}\right) - b^*\left(\overline{\boldsymbol{u}_r^{k+1}}^r, \boldsymbol{u}_r^{k+1}, \boldsymbol{\Phi}_r^{k+1}\right) - \left(p, \nabla \cdot \boldsymbol{\Phi}_r^{k+1}\right) = 0. \end{aligned} \tag{5.51}$$

Note that (5.18) implies that $\left(\boldsymbol{\eta}^k, \boldsymbol{\Phi}_r^{k+1}\right) = 0$ and $\left(\boldsymbol{\eta}^{k+1}, \boldsymbol{\Phi}_r^{k+1}\right) = 0$. Letting $\boldsymbol{r}^k = \boldsymbol{u}_t^{k+1} - \frac{\boldsymbol{u}^{k+1}-\boldsymbol{u}^k}{\Delta t}$, we obtain

$$\begin{aligned}&\frac{1}{\Delta t}\left(\boldsymbol{\Phi}_r^{k+1}, \boldsymbol{\Phi}_r^{k+1}\right) - \frac{1}{\Delta t}\left(\boldsymbol{\Phi}_r^k, \boldsymbol{\Phi}_r^{k+1}\right) + \nu\left(\nabla \boldsymbol{\Phi}_r^{k+1}, \nabla \boldsymbol{\Phi}_r^{k+1}\right)\\ &= \left(\boldsymbol{r}^k, \boldsymbol{\Phi}_r^{k+1}\right) + \nu\left(\nabla \boldsymbol{\eta}^{k+1}, \nabla \boldsymbol{\Phi}_r^{k+1}\right) + b^*\left(\boldsymbol{u}^{k+1}, \boldsymbol{u}^{k+1}, \boldsymbol{\Phi}_r^{k+1}\right)\\ &-b^*\left(\overline{\boldsymbol{u}_r^{k+1}}^r, \boldsymbol{u}_r^{k+1}, \boldsymbol{\Phi}_r^{k+1}\right) - \left(p, \nabla \cdot \boldsymbol{\Phi}_r^{k+1}\right). \end{aligned} \tag{5.52}$$

We estimate the LHS of (5.52) by applying the Cauchy-Schwarz and Young inequalities:

$$\begin{aligned}\text{LHS} &= \frac{1}{\Delta t}\left\|\boldsymbol{\Phi}_r^{k+1}\right\|^2 - \frac{1}{\Delta t}\left(\boldsymbol{\Phi}_r^k, \boldsymbol{\Phi}_r^{k+1}\right) + \nu\left\|\nabla \boldsymbol{\Phi}_r^{k+1}\right\|^2\\ &\geq \frac{1}{2\Delta t}\left(\left\|\boldsymbol{\Phi}_r^{k+1}\right\|^2 - \left\|\boldsymbol{\Phi}_r^k\right\|^2\right) + \nu\left\|\nabla \boldsymbol{\Phi}_r^{k+1}\right\|^2. \end{aligned} \tag{5.53}$$

Using (5.52) and (5.53), we obtain

$$\begin{aligned}&\left\|\boldsymbol{\Phi}_r^{k+1}\right\|^2 - \left\|\boldsymbol{\Phi}_r^k\right\|^2 + 2\nu\Delta t\left\|\nabla \boldsymbol{\Phi}_r^{k+1}\right\|^2\\ \leq\ & 2\Delta t\left(\boldsymbol{r}^k, \boldsymbol{\Phi}_r^{k+1}\right) + 2\nu\Delta t\left(\nabla \boldsymbol{\eta}^{k+1}, \nabla \boldsymbol{\Phi}_r^{k+1}\right) + 2\Delta t\, b^*\left(\boldsymbol{u}^{k+1}, \boldsymbol{u}^{k+1}, \boldsymbol{\Phi}_r^{k+1}\right)\\ &-2\Delta t\, b^*\left(\overline{\boldsymbol{u}_r^{k+1}}^r, \boldsymbol{u}_r^{k+1}, \boldsymbol{\Phi}_r^{k+1}\right) - 2\Delta t\,\left(p, \nabla \cdot \boldsymbol{\Phi}_r^{k+1}\right). \end{aligned} \tag{5.54}$$

Using the Cauchy-Schwarz and Young inequalities in (5.54), we get

$$\left(\boldsymbol{r}^k, \boldsymbol{\Phi}_r^{k+1}\right) \leq \left\|\boldsymbol{r}^k\right\|_{-1}\left\|\nabla \boldsymbol{\Phi}_r^{k+1}\right\| \leq C_1\left(\left\|\boldsymbol{r}^k\right\|_{-1}^2 + \left\|\nabla \boldsymbol{\Phi}_r^{k+1}\right\|^2\right), \tag{5.55}$$

$$\nu\left(\nabla \boldsymbol{\eta}^{k+1}, \nabla \boldsymbol{\Phi}_r^{k+1}\right) \leq \nu\left\|\nabla \boldsymbol{\eta}^{k+1}\right\|\left\|\nabla \boldsymbol{\Phi}_r^{k+1}\right\| \leq C_2\left(\left\|\nabla \boldsymbol{\eta}^{k+1}\right\|^2 + \left\|\nabla \boldsymbol{\Phi}_r^{k+1}\right\|^2\right). \tag{5.56}$$

The nonlinear terms in (5.54) can be written as follows:

$$\begin{aligned} & b^*\left(\boldsymbol{u}^{k+1},\boldsymbol{u}^{k+1},\boldsymbol{\Phi}_r^{k+1}\right)-b^*\left(\overline{\boldsymbol{u}_r^{k+1}}^r,\boldsymbol{u}_r^{k+1},\boldsymbol{\Phi}_r^{k+1}\right)\\ = \;& b^*\left(\boldsymbol{u}^{k+1},\boldsymbol{u}^{k+1},\boldsymbol{\Phi}_r^{k+1}\right)-b^*\left(\overline{\boldsymbol{u}^{k+1}}^r,\boldsymbol{u}^{k+1},\boldsymbol{\Phi}_r^{k+1}\right)\\ & +b^*\left(\overline{\boldsymbol{u}^{k+1}}^r,\boldsymbol{u}^{k+1},\boldsymbol{\Phi}_r^{k+1}\right)-b^*\left(\overline{\boldsymbol{u}_r^{k+1}}^r,\boldsymbol{u}_r^{k+1},\boldsymbol{\Phi}_r^{k+1}\right) \end{aligned}$$

$$\begin{aligned} &= b^*\left(\boldsymbol{u}^{k+1}-\overline{\boldsymbol{u}^{k+1}}^r,\boldsymbol{u}^{k+1},\boldsymbol{\Phi}_r^{k+1}\right)+\;b^*\left(\overline{\boldsymbol{u}^{k+1}}^r,\boldsymbol{u}^{k+1},\boldsymbol{\Phi}_r^{k+1}\right)\\ &-b^*\left(\overline{\boldsymbol{u}_r^{k+1}}^r,\boldsymbol{u}^{k+1},\boldsymbol{\Phi}_r^{k+1}\right)+\;b^*\left(\overline{\boldsymbol{u}_r^{k+1}}^r,\boldsymbol{u}^{k+1},\boldsymbol{\Phi}_r^{k+1}\right)-b^*\left(\overline{\boldsymbol{u}_r^{k+1}}^r,\boldsymbol{u}_r^{k+1},\boldsymbol{\Phi}_r^{k+1}\right)\\ &= b^*\left(\boldsymbol{u}^{k+1}-\overline{\boldsymbol{u}^{k+1}}^r,\boldsymbol{u}^{k+1},\boldsymbol{\Phi}_r^{k+1}\right)\\ &+b^*\left(\overline{\boldsymbol{u}^{k+1}-\boldsymbol{u}_r^{k+1}}^r,\boldsymbol{u}^{k+1},\boldsymbol{\Phi}_r^{k+1}\right)+b^*\left(\overline{\boldsymbol{u}_r^{k+1}}^r,\boldsymbol{u}^{k+1}-\boldsymbol{u}_r^{k+1},\boldsymbol{\Phi}_r^{k+1}\right)\\ &= b^*\left(\boldsymbol{u}^{k+1}-\overline{\boldsymbol{u}^{k+1}}^r,\boldsymbol{u}^{k+1},\boldsymbol{\Phi}_r^{k+1}\right)+\;b^*\left(\overline{\boldsymbol{\eta}^{k+1}}^r,\boldsymbol{u}^{k+1},\boldsymbol{\Phi}_r^{k+1}\right)\\ &-b^*\left(\overline{\boldsymbol{\Phi}_r^{k+1}}^r,\boldsymbol{u}^{k+1},\boldsymbol{\Phi}_r^{k+1}\right)+\;b^*\left(\overline{\boldsymbol{u}_r^{k+1}}^r,\boldsymbol{\eta}^{k+1},\boldsymbol{\Phi}_r^{k+1}\right)-\cancelto{0}{b^*\left(\overline{\boldsymbol{u}_r^{k+1}}^r,\boldsymbol{\Phi}_r^{k+1},\boldsymbol{\Phi}_r^{k+1}\right)}, \end{aligned} \tag{5.57}$$

where in the last term we have used (5.3). In (5.57), we apply Lemma 5.2, Lemma 5.11, Lemma 5.9 and the Cauchy-Schwarz and Young inequalities:

$$\begin{aligned} b^*\left(\overline{\boldsymbol{u}_r^{k+1}}^r,\boldsymbol{\eta}^{k+1},\boldsymbol{\Phi}_r^{k+1}\right) &\overset{(5.5)}{\leq} C\left\|\overline{\boldsymbol{u}_r^{k+1}}^r\right\|^{1/2}\left\|\nabla\overline{\boldsymbol{u}_r^{k+1}}^r\right\|^{1/2}\left\|\nabla\boldsymbol{\eta}^{k+1}\right\|\left\|\nabla\boldsymbol{\Phi}_r^{k+1}\right\|\\ &\overset{(5.35),(5.37)}{\leq} C\left\|\boldsymbol{u}_r^{k+1}\right\|^{1/2}\left\|\nabla\boldsymbol{u}_r^{k+1}\right\|^{1/2}\left\|\nabla\boldsymbol{\eta}^{k+1}\right\|\left\|\nabla\boldsymbol{\Phi}_r^{k+1}\right\|\\ &\leq C_3\left(\left\|\boldsymbol{u}_r^{k+1}\right\|\left\|\nabla\boldsymbol{u}_r^{k+1}\right\|\left\|\nabla\boldsymbol{\eta}^{k+1}\right\|^2+\left\|\nabla\boldsymbol{\Phi}_r^{k+1}\right\|^2\right); \end{aligned} \tag{5.58}$$

$$\begin{aligned} b^*\left(\overline{\boldsymbol{\eta}^{k+1}}^r,\boldsymbol{u}^{k+1},\boldsymbol{\Phi}_r^{k+1}\right) &\overset{(5.5)}{\leq} C\left\|\overline{\boldsymbol{\eta}^{k+1}}^r\right\|^{1/2}\left\|\nabla\overline{\boldsymbol{\eta}^{k+1}}^r\right\|^{1/2}\left\|\nabla\boldsymbol{u}^{k+1}\right\|\left\|\nabla\boldsymbol{\Phi}_r^{k+1}\right\|\\ &\overset{(5.35),(5.36)}{\leq} C\left\|\boldsymbol{\eta}^{k+1}\right\|^{1/2}\|S_r\|_2^{1/4}\left\|\boldsymbol{\eta}^{k+1}\right\|^{1/2}\left\|\nabla\boldsymbol{u}^{k+1}\right\|\left\|\nabla\boldsymbol{\Phi}_r^{k+1}\right\|\\ &\leq C_4\left(\|S_r\|_2^{1/2}\left\|\boldsymbol{\eta}^{k+1}\right\|^2\left\|\nabla\boldsymbol{u}^{k+1}\right\|^2+\left\|\nabla\boldsymbol{\Phi}_r^{k+1}\right\|^2\right); \end{aligned} \tag{5.59}$$

$$\begin{aligned} b^*\left(\overline{\boldsymbol{\Phi}_r^{k+1}}^r,\boldsymbol{u}^{k+1},\boldsymbol{\Phi}_r^{k+1}\right) &\overset{(5.5)}{\leq} C\left\|\overline{\boldsymbol{\Phi}_r^{k+1}}^r\right\|^{1/2}\left\|\nabla\overline{\boldsymbol{\Phi}_r^{k+1}}^r\right\|^{1/2}\left\|\nabla\boldsymbol{u}^{k+1}\right\|\left\|\nabla\boldsymbol{\Phi}_r^{k+1}\right\|\\ &\overset{(5.35),(5.37)}{\leq} C\left\|\boldsymbol{\Phi}_r^{k+1}\right\|^{1/2}\left\|\nabla\boldsymbol{\Phi}_r^{k+1}\right\|^{1/2}\left\|\nabla\boldsymbol{u}^{k+1}\right\|\left\|\nabla\boldsymbol{\Phi}_r^{k+1}\right\|\\ &= C\left\|\boldsymbol{\Phi}_r^{k+1}\right\|^{\frac{1}{2}}\left\|\nabla\boldsymbol{u}^{k+1}\right\|\left\|\nabla\boldsymbol{\Phi}_r^{k+1}\right\|^{\frac{3}{2}}\\ &\leq C_5\left(\left\|\nabla\boldsymbol{u}^{k+1}\right\|^4\left\|\boldsymbol{\Phi}_r^{k+1}\right\|^2+\left\|\nabla\boldsymbol{\Phi}_r^{k+1}\right\|^2\right). \end{aligned} \tag{5.60}$$

$$
\begin{aligned}
& b^*\left(\boldsymbol{u}^{k+1}-\overline{\boldsymbol{u}^{k+1}}^r, \boldsymbol{u}^{k+1}, \boldsymbol{\Phi}_r^{k+1}\right) \\
& \overset{(5.5)}{\leq} C\left\|\boldsymbol{u}^{k+1}-\overline{\boldsymbol{u}^{k+1}}^r\right\|^{1/2}\left\|\nabla\left(\boldsymbol{u}^{k+1}-\overline{\boldsymbol{u}^{k+1}}^r\right)\right\|^{1/2}\left\|\nabla \boldsymbol{u}^{k+1}\right\|\left\|\nabla \boldsymbol{\Phi}_r^{k+1}\right\| \\
& \leq C_6\left(\left\|\boldsymbol{u}^{k+1}-\overline{\boldsymbol{u}^{k+1}}^r\right\|\left\|\nabla\left(\boldsymbol{u}^{k+1}-\overline{\boldsymbol{u}^{k+1}}^r\right)\right\|\left\|\nabla \boldsymbol{u}^{k+1}\right\|^2+\left\|\nabla \boldsymbol{\Phi}_r^{k+1}\right\|^2\right), \qquad (5.61)
\end{aligned}
$$

which can be bounded by using Lemma 5.9. Since $\boldsymbol{\Phi}_r^{k+1} \in \boldsymbol{X}^r \subset \boldsymbol{V}^h$, the pressure term on the RHS of (5.54) can be written as

$$
-\left(p, \nabla \cdot \boldsymbol{\Phi}_r^{k+1}\right)=-\left(p-q_h, \nabla \cdot \boldsymbol{\Phi}_r^{k+1}\right), \qquad (5.62)
$$

where q_h is any function in Q^h. Thus, the pressure term can be estimated as follows by using the Cauchy-Schwarz and Young inequalities:

$$
-\left(p, \nabla \cdot \boldsymbol{\Phi}_r^{k+1}\right) \leq C_7\left(\|p-q_h\|^2+\left\|\nabla \boldsymbol{\Phi}_r^{k+1}\right\|^2\right). \qquad (5.63)
$$

Choosing $C_1 - C_7$ appropriately, then substituting inequalities (5.55)–(5.56), (5.58)–(5.60), and (5.61) in (5.54), we obtain

$$
\begin{aligned}
& \left\|\boldsymbol{\Phi}_r^{k+1}\right\|^2-\left\|\boldsymbol{\Phi}_r^k\right\|^2+C_8\,\Delta t\left\|\nabla \boldsymbol{\Phi}_r^{k+1}\right\|^2 \\
& \leq C_9\Bigg(\Delta t\left\|\boldsymbol{r}^k\right\|_{-1}^2+\Delta t\left\|\nabla \boldsymbol{\eta}^{k+1}\right\|^2+\Delta t\left\|\boldsymbol{u}_r^{k+1}\right\|\left\|\nabla \boldsymbol{u}_r^{k+1}\right\|\left\|\nabla \boldsymbol{\eta}^{k+1}\right\|^2 \\
& +\Delta t\left\|S_r\right\|_2^{1/2}\left\|\nabla \boldsymbol{u}^{k+1}\right\|^2\left\|\boldsymbol{\eta}^{k+1}\right\|^2+\Delta t\left\|\nabla \boldsymbol{u}^{k+1}\right\|^4\left\|\boldsymbol{\Phi}_r^{k+1}\right\|^2 \\
& +\left\|\boldsymbol{u}^{k+1}-\overline{\boldsymbol{u}^{k+1}}^r\right\|\left\|\nabla\left(\boldsymbol{u}^{k+1}-\overline{\boldsymbol{u}^{k+1}}^r\right)\right\|\left\|\nabla \boldsymbol{u}^{k+1}\right\|^2+\Delta t\,\|p-q_h\|^2\Bigg). \qquad (5.64)
\end{aligned}
$$

Summing (5.64) from $k=0$ to $k=M-1$, we have

$$
\begin{aligned}
& \left\|\boldsymbol{\Phi}_r^M\right\|^2+C_8\,\Delta t\sum_{k=0}^{M-1}\left\|\nabla \boldsymbol{\Phi}_r^{k+1}\right\|^2 \leq\left\|\boldsymbol{\Phi}_r^0\right\|^2+C_9\,\Delta t\Bigg(\sum_{k=0}^{M-1}\left\|\boldsymbol{r}^k\right\|_{-1}^2 \\
& \quad+\sum_{k=0}^{M-1}\left\|\boldsymbol{u}_r^{k+1}\right\|\left\|\nabla \boldsymbol{u}_r^{k+1}\right\|\left\|\nabla \boldsymbol{\eta}^{k+1}\right\|^2+\|S_r\|_2^{1/2}\sum_{k=0}^{M-1}\left\|\nabla \boldsymbol{u}^{k+1}\right\|^2\left\|\boldsymbol{\eta}^{k+1}\right\|^2 \\
& \quad+\sum_{k=0}^{M-1}\left\|\nabla \boldsymbol{u}^{k+1}\right\|^4\left\|\boldsymbol{\Phi}_r^{k+1}\right\|^2+\sum_{k=0}^{M-1}\|p-q_h\|^2+\sum_{k=0}^{M-1}\left\|\nabla \boldsymbol{\eta}^{k+1}\right\|^2 \\
& \quad+\sum_{k=0}^{M-1}\left\|\boldsymbol{u}^{k+1}-\overline{\boldsymbol{u}^{k+1}}^r\right\|\left\|\nabla\left(\boldsymbol{u}^{k+1}-\overline{\boldsymbol{u}^{k+1}}^r\right)\right\|\left\|\nabla \boldsymbol{u}^{k+1}\right\|^2\Bigg). \qquad (5.65)
\end{aligned}
$$

The first term on the RHS of (5.65) vanishes, since $\boldsymbol{u}_r^0 = \boldsymbol{w}_r^0$. By using the Poincaré-Friedrichs inequality, the second term on the RHS of (5.65) can be estimated as follows (see, e.g., [91]):

$$
\Delta t\sum_{k=0}^{M-1}\left\|\boldsymbol{r}^k\right\|_{-1}^2 \leq C_{10}\,\Delta t\sum_{k=0}^{M-1}\left\|\boldsymbol{r}^k\right\|^2 \leq C_{11}\,\Delta t^2. \qquad (5.66)
$$

Using (5.21), the third term on the RHS of (5.65) can be estimated as follows:

$$\Delta t \sum_{k=0}^{M-1} \|\nabla \boldsymbol{\eta}^{k+1}\|^2 \leq C_{12} \Bigg(h^{2m} + \|S_r\|_2 h^{2m+2} + (1 + \|S_r\|_2)\, \Delta t^2 + \sum_{j=r+1}^{d} \|\boldsymbol{\varphi}_j\|_1^2\, \lambda_j \Bigg). \tag{5.67}$$

To estimate the fourth term on the RHS of (5.65), we use Lemma 5.12 and Assumption 5.4.1:

$$\begin{aligned} \Delta t \sum_{k=0}^{M-1} \left\|\mathbf{u}_r^{k+1}\right\| \left\|\nabla \mathbf{u}_r^{k+1}\right\| \left\|\nabla \boldsymbol{\eta}^{k+1}\right\|^2 &\overset{(5.42)}{\leq} \widetilde{C}_{13}\, \Delta t \sum_{k=0}^{M-1} \left\|\nabla \mathbf{u}_r^{k+1}\right\| \left\|\nabla \boldsymbol{\eta}^{k+1}\right\|^2 \\ \overset{(5.42)}{\leq} C_{13} \left\|\nabla \boldsymbol{\eta}^{k+1}\right\|^2 &\overset{(5.23)}{\leq} C_{13} \Bigg(h^{2m} + \|S_r\|_2 h^{2m+2} + (1 + \|S_r\|_2)\, \Delta t^2 \\ &\quad + \sum_{j=r+1}^{d} \|\boldsymbol{\varphi}_j\|_1^2\, \lambda_j \Bigg), \end{aligned} \tag{5.68}$$

where we used estimate (5.23) in the derivation of (5.68). Using (5.21) would not have been enough for the asymptotic convergence of (5.68).

Using Assumption 5.4.1, the fifth term on the RHS of (5.65) can be bounded as follows:

$$\begin{aligned} & \Delta t\, \|S_r\|_2^{1/2} \sum_{k=0}^{M-1} \left\|\nabla \mathbf{u}^{k+1}\right\|^2 \left\|\boldsymbol{\eta}^{k+1}\right\|^2 \\ \overset{(5.22)}{\leq}\ & \widetilde{C}_{14}\, \Delta t\, \|S_r\|_2^{1/2} \sum_{k=0}^{M-1} \left\|\nabla \mathbf{u}^{k+1}\right\|^2 \left(h^{m+1} + \Delta t + \sqrt{\sum_{j=r+1}^{d} \lambda_j} \right)^2 \\ \overset{(5.7)}{\leq}\ & C_{14}\, \|S_r\|_2^{1/2} \left(h^{m+1} + \Delta t + \sqrt{\sum_{j=r+1}^{d} \lambda_j} \right)^2 . \end{aligned} \tag{5.69}$$

To estimate the seventh term on the RHS of (5.65), we use Lemma 5.9:

$$\begin{aligned} & \Delta t \sum_{k=0}^{M-1} \left\|\boldsymbol{u}^{k+1} - \overline{\boldsymbol{u}^{k+1}}^{r}\right\| \left\|\nabla \left(\boldsymbol{u}^{k+1} - \overline{\boldsymbol{u}^{k+1}}^{r}\right)\right\| \left\|\nabla \boldsymbol{u}^{k+1}\right\|^2 \\ \overset{(5.24)}{\leq}\ & \widetilde{C}_{15}\, \Delta t \sum_{k=0}^{M-1} \left\|\nabla \boldsymbol{u}^{k+1}\right\|^2 \frac{1}{\delta} \Bigg[\left(h^{2m+2} + \Delta t^2 + \sum_{j=r+1}^{d} \lambda_j \right) \\ & + \delta^2 \left(h^{2m} + \|S_r\|_2\, h^{2m+2} + (1 + \|S_r\|_2)\, \Delta t^2 + \sum_{j=r+1}^{d} \|\boldsymbol{\varphi}_j\|_1^2\, \lambda_j \right) + \delta^4 \Bigg] \\ \overset{(5.7)}{\leq}\ & C_{15}\, \frac{1}{\delta} \Bigg[\left(h^{2m+2} + \Delta t^2 + \sum_{j=r+1}^{d} \lambda_j \right) \\ & + \delta^2 \left(h^{2m} + \|S_r\|_2\, h^{2m+2} + (1 + \|S_r\|_2)\, \Delta t^2 + \sum_{j=r+1}^{d} \|\boldsymbol{\varphi}_j\|_1^2\, \lambda_j \right) + \delta^4 \Bigg]. \end{aligned} \tag{5.70}$$

Since in (5.62) q_h was an arbitrary function in Q^h, we can use the approximation property (5.10) in Assumption 5.2.3 to bound the eighth term on the RHS of (5.65) as follows:

$$\Delta t \sum_{k=0}^{M-1} \|p - q_h\|^2 \leq C_{16}\, h^{2m}\,. \tag{5.71}$$

Collecting (5.66)-(5.71), equation (5.65) becomes

$$\begin{aligned}
&\left\|\boldsymbol{\Phi}_r^M\right\|^2 + C_8\,\Delta t \sum_{k=0}^{M-1} \left\|\nabla \boldsymbol{\Phi}_r^{k+1}\right\|^2 \\
\leq C_9 \Bigg\{ &\quad \Delta t \sum_{k=0}^{M-1} \left\|\nabla \boldsymbol{u}^{k+1}\right\|^4 \left\|\boldsymbol{\Phi}_r^{k+1}\right\|^2 \\
&+ \left(h^{2m} + \|S_r\|_2 h^{2m+2} + (1+\|S_r\|_2)\,\Delta t^2 + \sum_{j=r+1}^{d} \|\boldsymbol{\varphi}_j\|_1^2\,\lambda_j\right) \\
&+ \|S_r\|_2^{1/2} \left(h^{m+1} + \Delta t + \sqrt{\sum_{j=r+1}^{d} \lambda_j}\right)^2 + h^{2m} + \Delta t^2 \\
&+ \frac{1}{\delta}\Bigg[\left(h^{2m+2} + \Delta t^2 + \sum_{j=r+1}^{d} \lambda_j\right) \\
&+ \delta^2 \left(h^{2m} + \|S_r\|_2\, h^{2m+2} + (1+\|S_r\|_2)\,\Delta t^2 + \sum_{j=r+1}^{d} \|\boldsymbol{\varphi}_j\|_1^2\,\lambda_j\right) + \delta^4\Bigg]\Bigg\} \\
= C_9 \Bigg\{ &\Delta t \sum_{k=0}^{M-1} \left\|\nabla \boldsymbol{u}^{k+1}\right\|^4 \left\|\boldsymbol{\Phi}_r^{k+1}\right\|^2 \\
&+ \left(h^{2m} + \|S_r\|_2 h^{2m+2} + (1+\|S_r\|_2)\,\Delta t^2 + \sum_{j=r+1}^{d} \|\boldsymbol{\varphi}_j\|_1^2\,\lambda_j\right) \\
&+ \|S_r\|_2^{1/2} \left(h^{2m+2} + \Delta t^2 + \sum_{j=r+1}^{d} \lambda_j\right) + \frac{1}{\delta}\left(h^{2m+2} + \Delta t^2 + \sum_{j=r+1}^{d} \lambda_j\right) \\
&+ \delta \left(h^{2m} + \|S_r\|_2\, h^{2m+2} + (1+\|S_r\|_2)\,\Delta t^2 + \sum_{j=r+1}^{d} \|\boldsymbol{\varphi}_j\|_1^2\,\lambda_j\right) + \delta^3 \Bigg\} \\
\overset{notation}{=} &\quad C_9 \Bigg\{ \Delta t \sum_{k=0}^{M-1} \left\|\nabla \boldsymbol{u}^{k+1}\right\|^4 \left\|\boldsymbol{\Phi}_r^{k+1}\right\|^2 \\
&+ \mathcal{F}\left(\delta, h, \Delta t, \|S_r\|_2, \{\lambda_j\}_{j=r+1}^d, \{\|\boldsymbol{\varphi}_j\|_1\}_{j=r+1}^d\right)\Bigg\}\,.
\end{aligned} \tag{5.72}$$

A discrete Gronwall lemma (see Lemma 27 in [106] and Lemma 5.1 in [84]) implies that, for

small enough Δt (i.e., $\Delta t < \left(C_9 \max_{1\leq k\leq M} \|\nabla \boldsymbol{u}^k\|^4\right)^{-1}$), the following inequality holds:

$$\begin{aligned} &\left\|\boldsymbol{\Phi}_r^M\right\|^2 + C_8\,\Delta t \sum_{k=0}^{M-1} \left\|\nabla \boldsymbol{\Phi}_r^{k+1}\right\|^2 \\ \leq\ & C_{17}\mathcal{F}\left(\delta, h, \Delta t, \|S_r\|_2, \{\lambda_j\}_{j=r+1}^d, \{\|\boldsymbol{\varphi}_j\|_1\}_{j=r+1}^d\right). \end{aligned} \tag{5.73}$$

By using (5.73), the triangle inequality, and (5.21)–(5.22), yields

$$\begin{aligned} &\left\|\mathbf{u}^M - \mathbf{u}_r^M\right\|^2 + \Delta t \sum_{k=0}^{M-1} \left\|\nabla\left(\mathbf{u}^{k+1} - \mathbf{u}_r^{k+1}\right)\right\|^2 \\ \leq\ & C\,\mathcal{F}\left(\delta, h, \Delta t, \|S_r\|_2, \{\lambda_j\}_{j=r+1}^d, \{\|\boldsymbol{\varphi}_j\|_1\}_{j=r+1}^d\right). \end{aligned} \tag{5.74}$$

This completes the proof. □

5.5 Numerical Results

In this section, we perform a numerical investigation of the theoretical results obtained in Section 5.4. To this end, we investigate whether the ROM filtering error estimate in Lemma 5.9 and the L-ROM approximation error estimate in Theorem 5.13 are recovered numerically.

In the numerical investigation, we consider the same test problem and computational setting as those used in Section IV.B in [91]. The problem is governed by the 2D incompressible NSE with an analytical solution. The exact velocity, $\boldsymbol{u} = (u, v)$, has the components $u = \frac{2}{\pi}\arctan(-500(y-t))\sin(\pi y)$, $v = \frac{2}{\pi}\arctan(-500(x-t))\sin(\pi x)$, and the exact pressure is given by $p = 0$. The diffusion coefficient is $\nu = 10^{-3}$ and the forcing term is chosen to match the exact solution. The spatial domain $[0,1]\times[0,1]$ is discretized by the Taylor-Hood FEs with the mesh size $h = 1/64$. To generate the POD basis, snapshots are collected over the time interval $[0,1]$ at every $\Delta T = 10^{-2}$ by recording the exact values of u and v on the FE mesh. We do not use the common centering trajectory approach; instead, we apply the method of snapshots to the snapshot data directly. The dimension of the POD basis is 101.

5.5.1 ROM Filtering Error

In this section, we perform a numerical investigation of the ROM filtering error (5.24) in Lemma 5.9. We define the following average squared filtering errors: $\mathcal{E}_{L^2} = \frac{1}{M+1}\sum_{k=0}^{M} \|\boldsymbol{u}^k - \overline{\boldsymbol{u}^k}^r\|^2, \mathcal{E}_{H^1} = \frac{1}{M+1}\sum_{k=0}^{M} \|\nabla(\boldsymbol{u}^k - \overline{\boldsymbol{u}^k}^r)\|^2$. The ROM filtering error bound (5.24) in Lemma 5.9 depends on the parameters $h, \Delta t, \delta$ as well as the ROM truncation errors $\Lambda_{L^2}^r = \sum_{j=r+1}^{d} \lambda_j$, $\Lambda_{H^1}^r =$

$\sum_{j=r+1}^{d} \|\boldsymbol{\varphi}_j\|_1^2 \lambda_j$. We numerically investigate the rates of convergence of $\mathcal{E}_{L^2}$ and $\mathcal{E}_{H^1}$ with respect to the time step Δt, filter radius δ, and ROM truncation error $\Lambda_{H^1}^r$.

First, we investigate the convergence rates with respect to δ. To this end, we fix $h = 1/64$, $r = 95$ and $\Delta t = 10^{-4}$ (note that the time step size should not matter in this test because the snapshots are FE interpolants of the exact solutions), and vary δ. With these choices, $h^{2m} = \mathcal{O}(10^{-8})$, $\Lambda_{L^2}^r = \mathcal{O}(10^{-8})$, $\Lambda_{H^1}^r = \mathcal{O}(10^{-3})$ and $\|S_r\|_2 = \mathcal{O}(10^5)$. Thus, the theoretical error estimate (5.24) in Lemma 5.9 yields the following asymptotic scaling:

$$\mathcal{E}_{L^2} \overset{(5.24)}{\sim} \mathcal{O}(\delta^2). \tag{5.75}$$

Note that (5.24) does not provide a scaling between $\mathcal{E}_{H^1}$ and δ.

We apply the DF (5.2) to the snapshot data and measure the numerical errors $\mathcal{E}_{L^2}$ and $\mathcal{E}_{H^1}$, which are listed in Table 5.1. Linear regressions of the errors, which are plotted in Fig. 5.1 for decreasing δ values, show the following scalings for the ROM filtering errors:

$$\mathcal{E}_{L^2} \sim \mathcal{O}(\delta^{2.52}) \tag{5.76}$$

$$\mathcal{E}_{H^1} \sim \mathcal{O}(\delta^{1.96}). \tag{5.77}$$

Thus, the theoretical scaling (5.75) is numerically recovered. On the other hand, although not verified theoretically in (5.24), we do observe the almost quadratic convergence of $\mathcal{E}_{H^1}$ with respect to δ (see Remark 5.10).

δ	$\mathcal{E}_{L^2}$	$\mathcal{E}_{H^1}$
1×10^{-2}	3.54×10^{-3}	9.87×10^{1}
5×10^{-3}	9.14×10^{-4}	4.65×10^{1}
2.5×10^{-3}	1.63×10^{-4}	1.22×10^{1}
2.0×10^{-3}	8.41×10^{-5}	6.79×10^{0}
1.67×10^{-3}	4.71×10^{-5}	3.97×10^{0}
1.25×10^{-3}	1.77×10^{-5}	1.56×10^{0}

Table 5.1: Average ROM filtering errors $\mathcal{E}_{L^2}$ and $\mathcal{E}_{H^1}$ for decreasing δ values.

Next, we investigate the convergence rates with respect to $\Lambda_{H^1}^r$. To this end, we fix $h = 1/64$, $\Delta t = 10^{-4}$, $\delta = 10^{-3}$ and vary r. With these choices, $h^{2m} = \mathcal{O}(10^{-8})$, $\delta^2 = \mathcal{O}(10^{-6})$ and $\|S_r\|_2 = \mathcal{O}(10^4) - \mathcal{O}(10^5)$. Thus, the theoretical error estimate (5.24) in Lemma 5.9 yields the following asymptotic scalings:

$$\mathcal{E}_{L^2} \overset{(5.24)}{\sim} \mathcal{O}(\Lambda_{H^1}^r) \tag{5.78}$$

$$\mathcal{E}_{H^1} \overset{(5.24)}{\sim} \mathcal{O}(\Lambda_{H^1}^r). \tag{5.79}$$

The numerical errors $\mathcal{E}_{L^2}$ and $\mathcal{E}_{H^1}$ are listed in Table 5.2 for increasing r values. The corresponding linear regressions, which are shown in Fig. 5.2, indicate the following scalings between the average ROM filtering errors and the ROM truncation error:

$$\mathcal{E}_{L^2} \sim \mathcal{O}\left((\Lambda_{H^1}^r)^{1.20}\right) \tag{5.80}$$

$$\mathcal{E}_{H^1} \sim \mathcal{O}\left((\Lambda_{H^1}^r)^{0.97}\right). \tag{5.81}$$

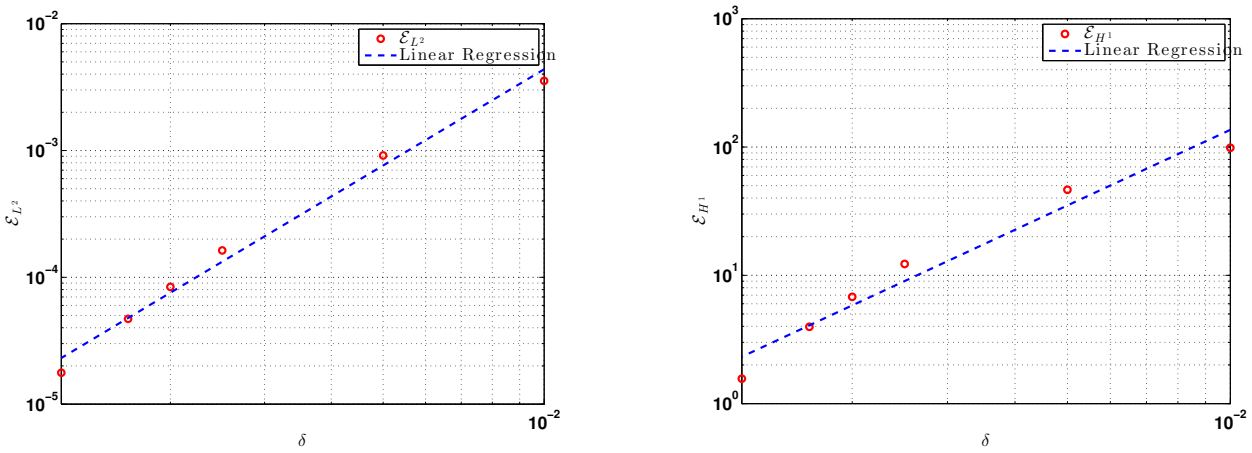


Figure 5.1: Linear regression of $\mathcal{E}_{L^2}$ and $\mathcal{E}_{H^1}$ with respect to δ.

r	$\Lambda^r_{H^1}$	$\mathcal{E}_{L^2}$	$\mathcal{E}_{H^1}$
30	1.23×10^{2}	3.29×10^{-3}	1.23×10^{2}
40	9.26×10^{1}	1.70×10^{-3}	9.27×10^{1}
50	6.73×10^{1}	9.05×10^{-4}	6.74×10^{1}
60	4.44×10^{1}	4.91×10^{-4}	4.46×10^{1}
70	2.09×10^{1}	2.39×10^{-4}	2.14×10^{1}
80	6.42×10^{0}	8.11×10^{-5}	7.06×10^{0}

Table 5.2: Average ROM filtering errors $\mathcal{E}_{L^2}$ and $\mathcal{E}_{H^1}$ for increasing r values.

Thus, the theoretical scalings (5.78) and (5.79) are numerically recovered.

5.5.2 L-ROM Approximation Error

In this section, we perform a numerical investigation of the L-ROM approximation error estimate (5.46) in Theorem 5.13. The L-ROM approximation error at the final time step is $\mathcal{E}^M_{L^2} = \|\boldsymbol{u}^M - \overline{\boldsymbol{u}^M}^r\|$. We numerically investigate the rates of convergence of $\mathcal{E}^M_{L^2}$ with respect to the time step Δt, filter radius δ and ROM truncation error $\Lambda^r_{H^1}$. To this end, we first note that in our numerical investigation $\delta < 1$ and $\|S_r\|_2 \gg 1$. Thus, the L-ROM approximation error bound (5.47) in Theorem 5.13 simplifies to the following:

$$\begin{aligned} & \mathcal{F}\left(\delta, h, \Delta t, \|S_r\|_2, \{\lambda_j\}_{j=r+1}^d, \{\|\boldsymbol{\varphi}_j\|_1\}_{j=r+1}^d\right) \\ = \;& h^{2m} + \|S_r\|_2 h^{2m+2} + \|S_r\|_2 \Delta t^2 + \Lambda^r_{H^1} \\ + \;& \|S_r\|_2^{1/2} \Lambda^r_{L^2} + \frac{1}{\delta}\left(h^{2m+2} + \Delta t^2 + \Lambda^r_{L^2}\right) + \delta^3 . \end{aligned} \tag{5.82}$$

The control parameters in the L-ROM approximation error rates of convergence in (5.82) are $h, \Delta t, r$, and δ.

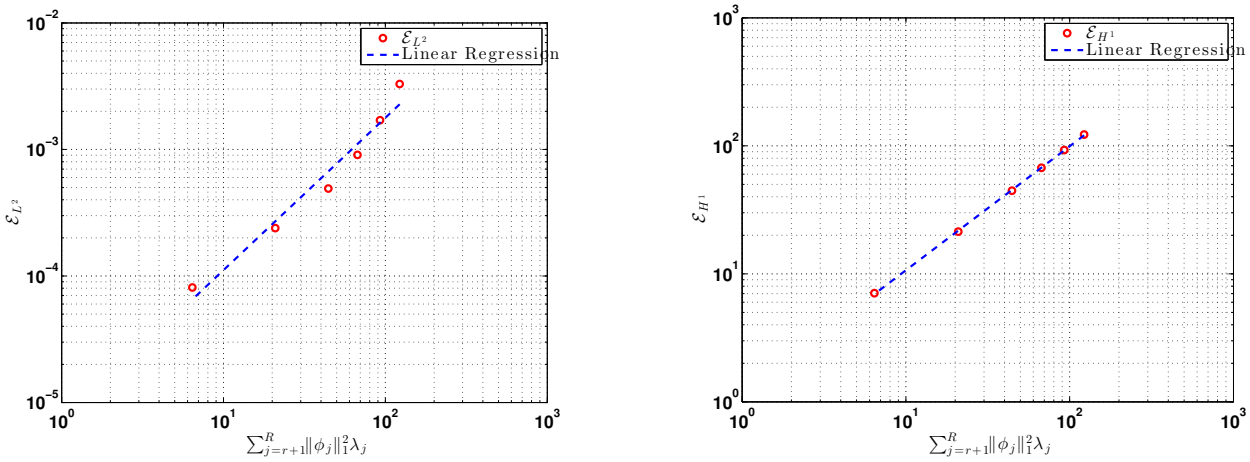


Figure 5.2: Linear regression of $\mathcal{E}_{L^2}$ and $\mathcal{E}_{H^1}$ with respect to $\Lambda_{H^1}^r$.

To determine the L-ROM approximation error rate of convergence with respect to Δt, we fix $h = 1/64$, $r = 99$, $\delta = 10^{-4}$ and vary Δt. With these choices, $h^{2m} = \mathcal{O}(10^{-8})$, $\Lambda_{L^2}^r = \mathcal{O}(10^{-8})$, $\Lambda_{H^1}^r = \mathcal{O}(10^{-3})$ and $\|S_r\|_2 = \mathcal{O}(10^5)$. Thus, the theoretical L-ROM approximation error bound (5.82) predicts the following rate of convergence of $\mathcal{E}_{L^2}^M$ with respect to Δt:

$$\mathcal{E}_{L^2}^M = \mathcal{O}(\Delta t)\,. \tag{5.83}$$

The L-ROM approximation error $\mathcal{E}_{L^2}^M$ is listed in Table 5.3 for decreasing Δt values. The corresponding linear regression, which is shown in Fig. 5.3, indicates the following L-ROM approximation error rate of convergence with respect to Δt:

$$\mathcal{E}_{L^2}^M = \mathcal{O}(\Delta t^{0.99})\,. \tag{5.84}$$

Thus, the theoretical rate of convergence (5.83) is numerically recovered.

Δt	$\mathcal{E}_{L^2}^M$
1×10^{-2}	2.36×10^{-2}
5×10^{-3}	2.33×10^{-2}
2.5×10^{-3}	6.49×10^{-3}
1.25×10^{-3}	3.49×10^{-3}
6.25×10^{-4}	1.96×10^{-3}

Table 5.3: L-ROM approximation error $\mathcal{E}_{L^2}^M$ for decreasing Δt values.

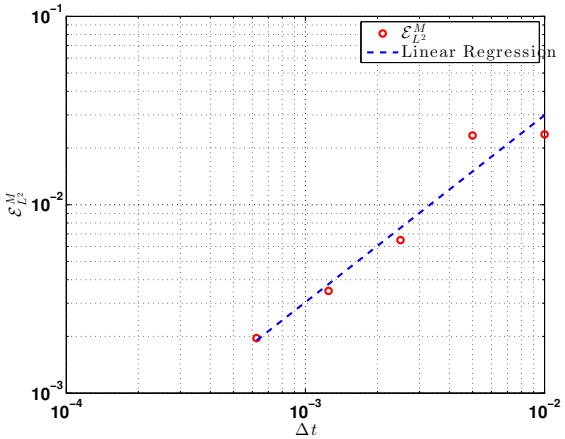


Figure 5.3: Linear regression of $\mathcal{E}_{L^2}^M$ with respect to Δt.

To determine the L-ROM approximation error rate of convergence with respect to δ, we fix $h = 1/64$, $r = 99$, $\Delta t = 10^{-4}$ and vary δ. With these choices, $h^{2m} = \mathcal{O}(10^{-8})$,

$\Delta t^2 = \mathcal{O}(10^{-8})$, $\Lambda^r_{L^2} = \mathcal{O}(10^{-8})$, $\Lambda^r_{H^1} = \mathcal{O}(10^{-3})$ and $\|S_r\|_2 = \mathcal{O}(10^5)$. Thus, the theoretical L-ROM approximation error bound (5.82) predicts the following rate of convergence of $\mathcal{E}^M_{L^2}$ with respect to δ:

$$\mathcal{E}^M_{L^2} = \mathcal{O}(\delta^{3/2})\,. \tag{5.85}$$

The L-ROM approximation error $\mathcal{E}^M_{L^2}$ is listed in Table 5.4 for decreasing δ values. The corresponding linear regression, which is shown in Fig. 5.4, indicates the following L-ROM approximation error rate of convergence with respect to δ:

$$\mathcal{E}^M_{L^2} = \mathcal{O}(\delta^{2.09})\,. \tag{5.86}$$

We note that the numerical rate of convergence (5.86) is higher than the theoretical rate of convergence (5.85) (see Remark 5.10).

δ	$\mathcal{E}^M_{L^2}$
5×10^{-1}	8.47×10^{-1}
2.5×10^{-1}	4.15×10^{-1}
1.25×10^{-1}	1.14×10^{-1}
6.25×10^{-2}	1.96×10^{-2}
3.12×10^{-2}	2.81×10^{-3}
1.56×10^{-2}	9.59×10^{-4}

Table 5.4: L-ROM approximation error $\mathcal{E}^M_{L^2}$ for decreasing δ values.

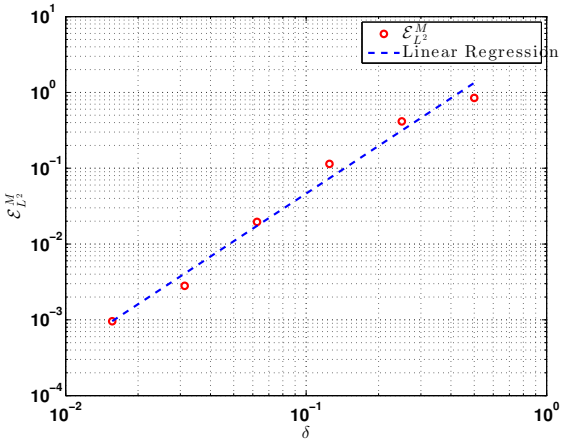


Figure 5.4: Linear regression of $\mathcal{E}^M_{L^2}$ with respect to δ.

To determine the L-ROM approximation error rate of convergence with respect to $\Lambda^r_{H^1}$, we fix $h = 1/64$, $\Delta t = 10^{-4}$, $\delta = 10^{-2}$ and vary r. With these choices, $h^{2m} = \mathcal{O}(10^{-8})$ and $\|S_r\|_2 = \mathcal{O}(10^2) - \mathcal{O}(10^5)$. Thus, the theoretical L-ROM approximation error bound (5.82) predicts the following rate of convergence of $\mathcal{E}^M_{L^2}$ with respect to $\Lambda^r_{H^1}$:

$$\mathcal{E}^M_{L^2} = \mathcal{O}(\Lambda^r_{H^1})\,. \tag{5.87}$$

The L-ROM approximation error $\mathcal{E}^M_{L^2}$ is listed in Table 5.5 for increasing r values. The corresponding linear regression, which is shown in Fig. 5.5, indicates the following L-ROM approximation error rate of convergence with respect to $\Lambda^r_{H^1}$:

$$\mathcal{E}^M_{L^2} = \mathcal{O}((\Lambda^r_{H^1})^{1.53})\,. \tag{5.88}$$

Thus, the theoretical rate of convergence (5.87) is numerically recovered.

r	$\Lambda^r_{H^1}$	$\mathcal{E}^M_{L^2}$
10	1.99×10^2	9.62×10^{-2}
20	1.57×10^2	5.15×10^{-2}
30	1.22×10^2	3.05×10^{-2}
40	9.26×10^1	2.09×10^{-2}
50	6.73×10^1	1.83×10^{-2}

Table 5.5: L-ROM approximation error $\mathcal{E}^M_{L^2}$ for increasing r values.

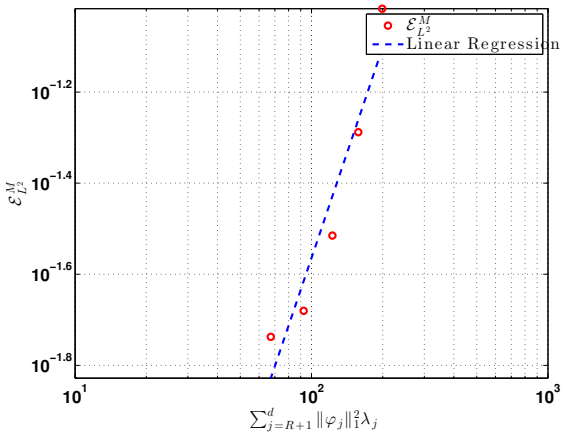


Figure 5.5: Linear regression of $\mathcal{E}^M_{L^2}$ with respect to $\Lambda^r_{H^1}$.

5.6 Summary

In this chapter, we took a first step in the numerical analysis of the Reg-ROMs and the explicit ROM spatial filter. In Theorem 5.13, we proved error estimates for the FE discretization of one such Reg-ROM, the L-ROM [88, 138, 165]. In Lemma 5.9, we also proved error estimates for the FE discretization of the ROM differential filter, which is the explicit ROM spatial filter that we used in the construction of the L-ROM. Finally, in Section 5.5, we provided a numerical verification of the ROM filtering error estimate derived in Lemma 5.9 and the L-ROM approximation error estimate in Theorem 5.13. In our numerical investigation, we considered the 2D incompressible NSE with an analytical solution and small diffusion coefficient $\nu = 10^{-3}$, which is the computational setting used in [91].

There are several research directions that could be pursued. As noted in Remark 5.10, one could try to extend from the FE setting to the ROM setting Lemma 2.4 in [62] instead of Lemma 2.12 in [105], as we did in Lemma 5.9, since the former could yield better δ scalings of the H^1 seminorm of the filtering error. However, one would probably first have to prove the H^1 stability of the ROM L^2 projection, which, to our knowledge, has not been achieved yet. Another research direction is the extension of the numerical analysis for the L-ROM to other Reg-ROMs, such as the evolve-then-filter ROM [165]. Finally, one could also try to prove error estimates for the novel large eddy simulation ROMs introduced in [168], in which the explicit ROM filter error plays a central role.

Chapter 6

Leray ROM (L-ROM): Stochastic Systems

In this chapter, we extend the ROM spatial filtering and the LES-ROM framework to stochastic systems. As a first step in this direction, we consider a simplified setting: a stochastic Burgers equation with linear multiplicative noise. Our preliminary results suggests that, in a convection-dominated regime, standard ROMs yield inaccurate results in the form of spurious numerical oscillations. To alleviate these oscillations, we use the Leray ROM, which increases the numerical stability of the standard ROM by smoothing (regularizing) the convective term with an explicit spatial filter. The Leray ROM yields significantly better results than the standard ROM and is more robust with respect to changes in the strength of the noise.

6.1 Introduction

Numerical instability, usually in the form of unphysical numerical oscillations, is one of the main challenges for ROMs of fluid flows. There are several sources of numerical instability of ROMs for fluid flows [42], such as (i) the convection-dominated (high Reynolds number) regime, in which the convection nonlinear term plays a central role [12, 85, 118]; and (ii) the inf-sup condition, which imposes a constraint on the ROM velocity and pressure spaces [16, 42]. To mitigate the spurious numerical oscillations created by these sources of numerical instability, various stabilized ROMs have been proposed (see, e.g., [7, 12, 13, 16, 18, 26, 44, 54, 76, 94, 126, 133, 162, 166] for such examples). A promising recent development in this class of methods is regularized ROMs [138, 165], which use explicit spatial filtering to increase the numerical stability of the ROM approximation.

Recently, the development of ROMs for systems involving random components has also received increased attention. For instance, ROMs for partial differential equations (PDEs) subject to random inputs acting on the boundary as well as PDEs with random coefficients have been considered in various contexts [33, 34, 51, 53, 65, 103, 52, 80]. However, ROMs for evolutionary PDEs driven by stochastic processes such as Brownian motions seem to be much less investigated. To our knowledge, only a few works are available [40]; see also [49].

In this work, we consider ROMs within the context of nonlinear stochastic PDEs (SPDEs)

that are of relevance to fluid dynamics. The main purpose is to investigate within a simple relevant setting—a stochastic Burgers equation (SBE) driven by linear multiplicative noise—the stabilization of the standard G-ROM in a convection-dominated regime. It is numerically illustrated that spurious oscillations developed in a G-ROM persist as the noise is turned on, and the oscillations worsen as the noise amplitude increases. A Leray regularized ROM (L-ROM) is then tested. The L-ROM provides more accurate modeling of the SBE dynamics by greatly reducing the artificial oscillations of the G-ROM, especially when the dimensions of the reduced models are low; cf. Figs. 6.3–6.6. It is further illustrated that the L-ROM is much more robust than the G-ROM with respect to the noise amplitude as revealed by the statistics of the corresponding modeling errors, which have significantly lower mean and variance; cf. Fig. 6.7.

6.2 ROMs for a Stochastic Burgers Equation

The viscous Burgers equation and its stochastic versions have been used previously to test new techniques in reduced order modeling and related contexts; see among many others [47, 48, 49, 99, 100, 117]. In this work, we will focus on a stochastic Burgers equation (SBE) driven by linear multiplicative noise, which is presented briefly in Section 6.2.1. To fix ideas, the ROMs explored in this chapter will be derived based on the POD. In Section 6.2.2, we outline the main steps in the derivation of the POD basis and standard G-ROM for the SBE. In Section 5.3, we develop the L-ROM.

6.2.1 Stochastic Burgers Equation (SBE)

We focus on the following stochastic Burgers equation (SBE) driven by linear multiplicative noise:

$$\begin{aligned} &\mathrm{d}u = \Big(\nu u_{xx} - u u_x\Big)\mathrm{d}t + \sigma u \circ \mathrm{d}W_t, \\ &u(0,t) = u(1,t) = 0, \quad t \geq 0, \\ &u(x,0) = u_0(x), \qquad x \in (0,1), \end{aligned} \tag{6.1}$$

where ν is a positive diffusion coefficient, W_t is a two-sided one-dimensional Wiener process, σ is a positive constant which measures the "amplitude" of the noise, and u_0 is some appropriate initial datum to be specified below. To fix ideas, the multiplicative noise term $\sigma u \circ \mathrm{d}W_t$ is understood in the sense of Stratonovich [123].

SPDEs driven by linear multiplicative noise such as the SBE (6.1) arise in various contexts, including turbulence theory or non-equilibrium phase transitions [29, 58, 116], the modeling of randomly fluctuating environment [21] in spatially-extended harvesting models [50, 83, 114, 115, 135, 136], or simply the modeling of parameter disturbances [30].

6.2.2 Galerkin ROM (G-ROM) for SBE

In this section, we briefly describe the POD and G-ROM for the SBE. The POD starts with the snapshots, which, in this chapter, are numerical approximations of the SBE (6.1) at different time instances. The POD seeks a low-dimensional basis that approximates the

snapshots optimally with respect to a certain norm. In this work, we employ the commonly used L^2-norm (see, e.g., [100] for alternatives). The solution of the minimization problem is equivalent to the solution of an eigenvalue problem [42]. The POD subspace of a given dimension r is spanned by the first r POD basis functions, which are the normalized functions $\{\boldsymbol{\varphi}_j\}_{j=1}^r$ that correspond to the first r largest eigenvalues of the aforementioned eigenvalue problem:

$$\boldsymbol{X}^r := \text{span}\{\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_r\}. \tag{6.2}$$

Note that the POD functions are orthogonal to each other with respect to the L^2-inner product $\langle \cdot, \cdot \rangle$ on the underlying phase space:

$$\langle \boldsymbol{\varphi}_i, \boldsymbol{\varphi}_j \rangle = \delta_{ij}, \tag{6.3}$$

where δ_{ij} denotes the Kronecker-delta. Note also that in (6.3) and the remainder of the chapter, the POD basis functions are considered as continuous functions on the spatial domain, since they are linear combinations of finite element basis functions.

The derivation of the POD-based Galerkin ROM (G-ROM) follows the standard Galerkin approximation procedure with the underlying basis taken to be the POD basis. For the sake of clarity, we sketch this derivation for the SBE (6.1) below. Given a positive integer r, the r-dimensional POD Galerkin approximation u_r of the SBE solution u takes the following form:

$$\boldsymbol{u}_r(\boldsymbol{x}, t; \omega) := \sum_{j=1}^{r} a_j(t; \omega) \boldsymbol{\varphi}_j(\boldsymbol{x}), \tag{6.4}$$

where the time-varying random coefficients $\{a_j(t, \omega)\}_{j=1}^r$ are determined by solving:

$$\Big\langle \mathrm{d}u_r, \boldsymbol{\varphi}_j \Big\rangle = \Big\langle \Big(\nu(u_r)_{xx} - u_r(u_r)_x\Big), \boldsymbol{\varphi}_j \Big\rangle \mathrm{d}t + \sigma \langle u_r, \boldsymbol{\varphi}_j \rangle \circ \mathrm{d}W_t, \qquad j = 1, \cdots, r. \tag{6.5}$$

The above system can be recast into the following more explicit form by using the expansion of u_r given in (6.4) and the orthogonality property satisfied by the POD basis functions given in (6.3):

$$\mathrm{d}a_j = \Big[-\nu \sum_{k=1}^{r} a_k \Big\langle \Big((\boldsymbol{\varphi}_k)_x, (\boldsymbol{\varphi}_j)_x \Big\rangle + \sum_{k,l=1}^{r} a_k a_l \Big\langle \boldsymbol{\varphi}_k (\boldsymbol{\varphi}_l)_x, \boldsymbol{\varphi}_j \Big\rangle \Big] \mathrm{d}t + \sigma a_j \circ \mathrm{d}W_t, \tag{6.6}$$

where $j = 1, \cdots, r$. This system of stochastic differential equations (SDEs) is the r-dimensional Galerkin ROM for the SBE (6.1).

6.2.3 Leray ROM (L-ROM) for SBE

To investigate fixes for G-ROM's potential numerical instability in the convection-dominated regime of the SBE (6.1), we draw inspiration from the deterministic case and consider *regularized ROMs (Reg-ROMs)*. These Reg-ROMs belong to the wide class of stabilized ROMs (see, e.g., [7, 12, 13, 16, 18, 26, 44, 54, 76, 85, 94, 118, 126, 133, 162, 166] for such examples). What distinguishes the Reg-ROMs from the other stabilized ROMs is that they increase the numerical stability of the model by using *explicit spatial filtering*, which is a relatively new concept in the ROM field [162, 138, 165]. In this study, we will use the simplest such

Reg-ROM, the *Leray ROM (L-ROM)* [138, 165], which is based on a specific way of filtering the convective term in the SBE (6.1) as explained below.

The extension of the deterministic L-ROM proposed in [138, 165] to the stochastic problem (6.1) at hand is straightforward. There is only one crucial difference in its derivation compared to the derivation of the G-ROM as outlined in Section 6.2.2, which consists of replacing the nonlinear term $u_r(u_r)_x$ in (6.5) by a regularized term $\overline{u}_r(u_r)_x$ here. This regularized version, $\overline{u}_r$, of u_r is obtained based on the usage of the DF[1] : Let δ be the radius of the DF. For a given $u_r \in \boldsymbol{X}^r$, find $\overline{u}_r \in \boldsymbol{X}^r$ such that

$$\left\langle \left(I - \delta^2\Delta\right)\overline{u}_r, \boldsymbol{\varphi}_j\right\rangle = \langle u_r, \boldsymbol{\varphi}_j\rangle, \quad \forall\, j = 1, \dots r\,. \tag{6.7}$$

Namely, the r-dimensional L-ROM approximation u_r of the SBE solution u takes the following form:

$$u_r(\boldsymbol{x}, t; \omega) := \sum_{j=1}^{r} a_j(t;\omega)\boldsymbol{\varphi}_j(\boldsymbol{x}), \tag{6.8}$$

where the time-varying random coefficients $\{a_j(t,\omega)\}_{j=1}^r$ are determined by solving:

$$\left\langle \mathrm{d}u_r, \boldsymbol{\varphi}_j\right\rangle = \left\langle \Big(\nu(u_r)_{xx} - \overline{u}_r(u_r)_x\Big), \boldsymbol{\varphi}_j\right\rangle \mathrm{d}t + \sigma\langle u_r, \boldsymbol{\varphi}_j\rangle \circ \mathrm{d}W_t, \qquad j = 1, \cdots, r. \tag{6.9}$$

Since at each time instance t, the sought regularization $\overline{\boldsymbol{u}}^r(t, \cdot; \omega)$ lives in $\boldsymbol{X}^r$, it admits the following expansion:

$$\overline{u}_r(t, x; \omega) \equiv \sum_{k=1}^{r} \overline{a}_k(t;\omega)\boldsymbol{\varphi}_k(\boldsymbol{x})\,, \tag{6.10}$$

where $\overline{a}_j$ can be determined by using the expansion (6.10) in (6.7), which leads to

$$\sum_{k=1}^{r} \overline{a}_k(t;\omega)\boldsymbol{\varphi}_k = \overline{\sum_{k=1}^{r} a_k(t;\omega)\boldsymbol{\varphi}_k} = \sum_{k=1}^{r} a_k(t;\omega)\overline{\boldsymbol{\varphi}}_k\,, \tag{6.11}$$

and the filtered POD mode $\overline{\boldsymbol{\varphi}}_k$, $1 \le k \le r$, is determined via

$$\left\langle \left(I - \delta^2\Delta\right)\overline{\boldsymbol{\varphi}}_k, \boldsymbol{\varphi}_j\right\rangle = \langle \boldsymbol{\varphi}_k, \boldsymbol{\varphi}_j\rangle, \quad \forall\, j = 1, \dots r\,. \tag{6.12}$$

Consequently, in contrast to the G-ROM given in (6.6), the r-dimensional L-ROM for SBE (6.1) is given by:

$$\mathrm{d}a_j = \Big[-\nu \sum_{k=1}^{r} a_k \Big\langle \big((\boldsymbol{\varphi}_k)_x, (\boldsymbol{\varphi}_j)_x\big\rangle + \sum_{k,l=1}^{r} a_k a_l \Big\langle \overline{\boldsymbol{\varphi}}_k (\boldsymbol{\varphi}_l)_x, \boldsymbol{\varphi}_j\Big\rangle\Big] \mathrm{d}t + \sigma a_j \circ \mathrm{d}W_t, \tag{6.13}$$

where $j = 1, \cdots, r$.

[1]Differential filters have been used in the simulation of convection-dominated flows with standard numerical methods [71, 72]. In reduced order modeling, the DF was first used in [138] and extended in [165].

6.3 Computational Investigation

In this section, we present a computational investigation on potential numerical instability of the standard G-ROM (6.6) for the SBE (6.1) and on a possible alleviation of such instability achieved by the L-ROM (6.13).

It has been observed in a previous study [165] that, for the deterministic Burgers equation placed in a convection-dominated regime, the G-ROM yields excessive spurious oscillations, especially when the dimension of the G-ROM is low. Similar to [165], we set up the numerical experiments for the SBE (6.1) in a regime with a small diffusion coefficient ($\nu = 10^{-3}$) and a steep internal layer; see Section 6.3.1. In Section 6.3.2, the emergence of such oscillations is confirmed in the current stochastic setting as well. The improvement achieved by the L-ROM in the form of significant reduction of the spurious oscillations is then presented in Section 6.3.3. Finally, some preliminary statistical tests are presented in Section 6.3.4, which also shows the robustness of the L-ROM with respect to the strength of the noise.

6.3.1 Setup of the Numerical Experiments

In this section, we present a short description of the setup of the numerical experiments.

Numerical Discretization of the SBE. The SBE (6.1) is solved by a semi-implicit Euler scheme as given in [49, Section 6.1]. For the reader's convenience, we briefly describe the numerical discretization below, and refer to [49, Section 6.1] for more details. We also refer the reader to [5, 31, 40, 86, 92, 109] for other numerical approximation schemes of nonlinear SPDEs.

At each time step the nonlinearity $uu_x = (u^2)_x/2$ and the noise term $\sigma u \circ \mathrm{d}W_t$ are treated explicitly, and the other terms are treated implicitly. The Laplacian operator is discretized using the standard second-order central difference approximation. The resulting semi-implicit scheme reads as follows:

$$u_j^{n+1} - u_j^n = \Big(\nu\Delta_d u_j^{n+1} + \frac{\sigma^2}{2}u_j^n - \frac{1}{2}\nabla_d\big((u_j^n)^2\big)\Big)\Delta t + \sigma\zeta_n u_j^n\sqrt{\Delta t}\,, \tag{6.14}$$

where u_j^n is the discrete approximation of $u(j\Delta x, n\Delta t)$, Δx the mesh size of the spatial discretization, and Δt the time step. The discretized Laplacian Δ_d and the discretized spatial derivative ∇_d in (6.14) are given by

$$\Delta_d u_j^n = \frac{u_{j-1}^n - 2u_j^n + u_{j+1}^n}{(\Delta x)^2};\quad \nabla_d\big((u_j^n)^2\big) = \frac{(u_{j+1}^n)^2 - (u_j^n)^2}{\Delta x},\quad j \in \{1, \cdots, N_x - 2\}\,.$$

The boundary conditions in (6.14) are $u_0^n = u_{N_x-1}^n = 0$, where N_x is the total number of grid points used for the discretization of the spatial domain $[0, 1]$. The ζ_n in (6.14) are random variables drawn independently from a normal distribution $\mathcal{N}(0, 1)$. Note that the additional drift term $\sigma^2 u_j^n/2$ in the RHS of (6.14) is due to the conversion of the Stratonovich noise term $\sigma u \circ \mathrm{d}W_t$ into its Itô form. Throughout the chapter, the simulations of the SBE (6.1) are performed for $\Delta t = 10^{-4}$ and $N_x = 1025$ so that $\Delta x \approx 9.8 \times 10^{-4}$. The diffusion coefficient ν is set to be 0.001. The values of the parameter σ will be specified below.

Choice of the Initial Data. The initial condition is chosen to be a mollified and slightly shifted version of the step function used in [99], which is given by

$$\boldsymbol{u}_0(x) = \int_{-\infty}^{\infty} \xi(y)\phi_\epsilon(x-y)\,\mathrm{d}y, \qquad x \in [0,1]. \tag{6.15}$$

Here, ξ is the step function defined by $\xi(x) = 1$ if $x \in (0.05, 0.55)$ and $\xi(x) = 0$ otherwise. The mollifier ϕ_ϵ is given by $\phi_\epsilon(x) = \frac{1}{\epsilon}\phi(\frac{x}{\epsilon})$ with

$$\phi(x) = \begin{cases} C\exp\big(-\frac{1}{(1-x^2)}\big) & \text{if } |x| < 1, \\ 0 & \text{otherwise,} \end{cases}$$

and the normalization constant C is chosen such that $\int_{-1}^{1}\phi(x)\,\mathrm{d}x = 0$. Throughout our numerical experiments, the parameter ϵ in the mollifier ϕ_ϵ is set to be $\epsilon = 0.01$.

The modification adopted here is mainly intended to enforce the compatibility of the initial and boundary condition at the left boundary point ($x = 0$) and to avoid any potential regularity issues that may arise in our numerical discretization of the SBE in (6.14) due to the discontinuity in the step function.

As will be seen below, by choosing such a step-function like initial profile and by setting the diffusion constant ν sufficiently small, the SBE exhibits interesting transient dynamics that will turn out to be a good laboratory to study the potential instability of the G-ROM; cf. Fig. 6.1.

Numerical Integration of the ROMs. The discretization of the G-ROM (6.6) and the L-ROM (6.13) are carried out by using a standard Euler-Maruyama scheme (see, e.g., [96, p. 305]). For instance, the corresponding G-ROM discretization is given by:

$$\begin{aligned} a_j^{n+1} - a_j^n = \Big[&-\nu\sum_{k=1}^{r} a_k^n \Big\langle \big((\boldsymbol{\varphi}_k)_x, (\boldsymbol{\varphi}_j)_x\big\rangle + \frac{\sigma^2}{2}a_j^n \\ &+ \sum_{k,l=1}^{r} a_k^n a_l^n \Big\langle \boldsymbol{\varphi}_k(\boldsymbol{\varphi}_l)_x, \boldsymbol{\varphi}_j\Big\rangle\Big]\Delta t + \sigma\zeta_n a_j^n\sqrt{\Delta t}\,, \quad j = 1,\cdots,r, \end{aligned} \tag{6.16}$$

where, as in (6.14), ζ_n are random variables drawn independently from a normal distribution $\mathcal{N}(0,1)$, and $n = 1,\cdots,N$, with N being the total number of time steps.

6.3.2 G-ROM Results: Spurious Oscillations

In this section, we assess the performance of the G-ROM in its ability to reproduce the SBE's spatio-temporal field for a fixed noise amplitude $\sigma = 0.3$ and an arbitrarily fixed realization of the noise. The statistical relevance of the results presented in this section is confirmed in Section 6.3.4.

For this purpose, we first simulate the SBE (6.1) over the time interval $[0,1]$ following the numerical setup presented in Section 6.3.1 and construct the POD basis functions used in the derivation of the G-ROM (6.6). In Fig. 6.1, the numerically simulated spatio-temporal field of the SBE (6.1) as well as the initial profile and the final time solution profile are plotted.

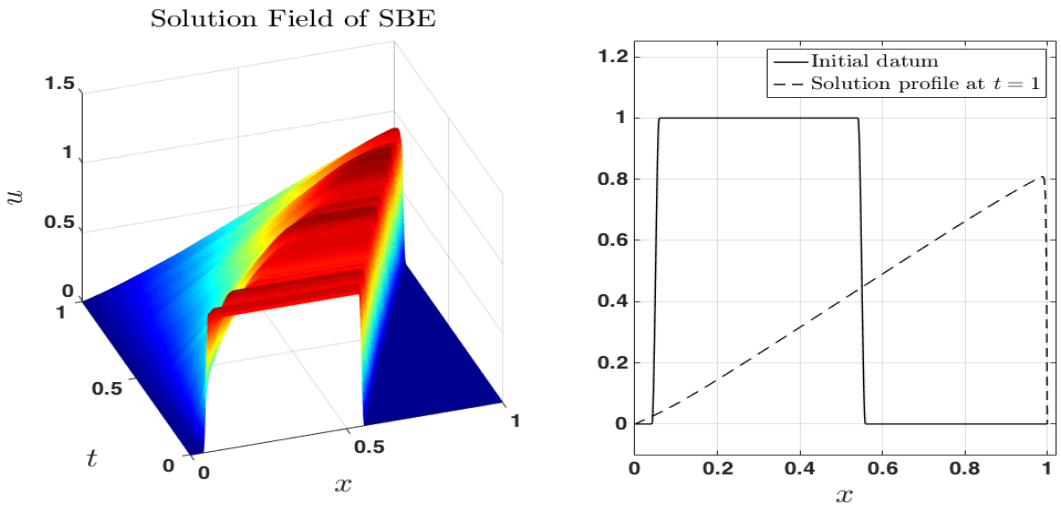


Figure 6.1: The numerically simulated spatio-temporal field of the SBE (6.1) with $\sigma = 0.3$ forced by an arbitrary realization of the noise (left panel), and the initial profile given by (6.15) with $\epsilon = 0.01$ (right panel, solid line) as well as the solution profile at time $t = 1$ (right panel, dashed line).

To construct the POD basis functions used in the derivation of the G-ROM (6.6), we collected 101 equally spaced snapshots (without subtracting the centering trajectory) from the simulated SBE spatio-temporal field, and we used the method of snapshots [145]. For illustration purposes, we plot four POD basis functions in Fig. 6.2.

The tests for the G-ROM are carried out with dimension $r = 6$ as well as $r = 10$; the results are plotted in Fig. 6.3. In both cases, the percentage of the total kinetic energy contained in the first r modes is already high: 98.5% for $r = 6$ and 99.3% for $r = 10$. Despite such a high percentage of energy captured by the first r POD modes, the corresponding G-ROM exhibits very strong spurious oscillations, as can be observed from both the reconstructed spatio-temporal fields and the final-time solution profiles in Fig. 6.3. On the other hand, an inspection on the evolution of the projected dynamics onto each POD mode reveals that the G-ROM is performing actually quite well in modeling the dynamics of the first two modes, while its performance deteriorates for higher frequency modes; see Fig. 6.4 for the case $r = 6$.

For the SBE problem studied here, as the dimension of the G-ROM increases, the overall accuracy also improves, as can already be seen in Fig. 6.3. Note also that the G-ROM performance improves as the diffusion coefficient ν increases (results not shown). This behavior is expected since increasing ν increases the diffusion effects, which, in turn, reduces the steepness of the localized internal layer. These numerical results suggest the convection-dominated regime to be a primary cause of the G-ROM's numerical instability observed here, just as in the deterministic case [165].

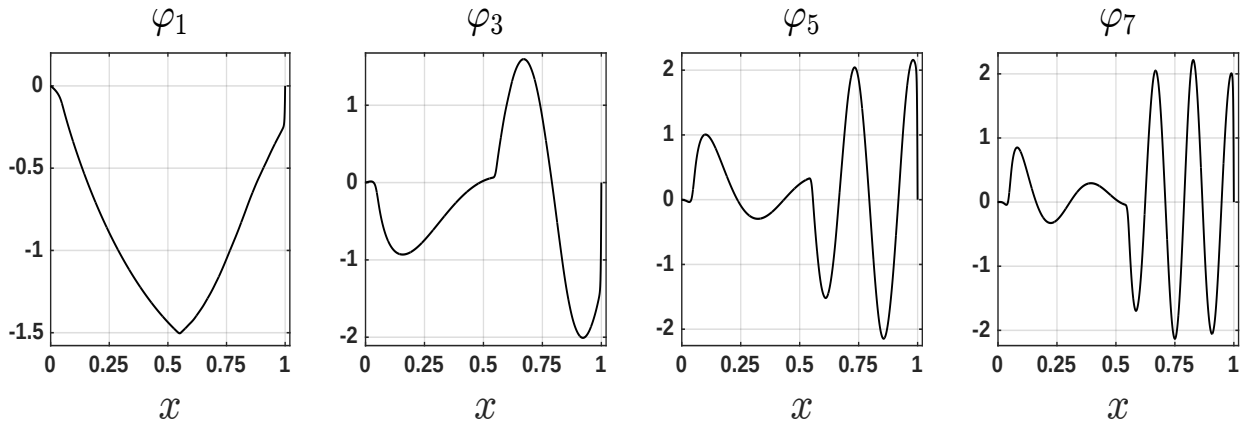


Figure 6.2: A few POD basis functions constructed based on the spatio-temporal field plotted in Fig. 6.1.

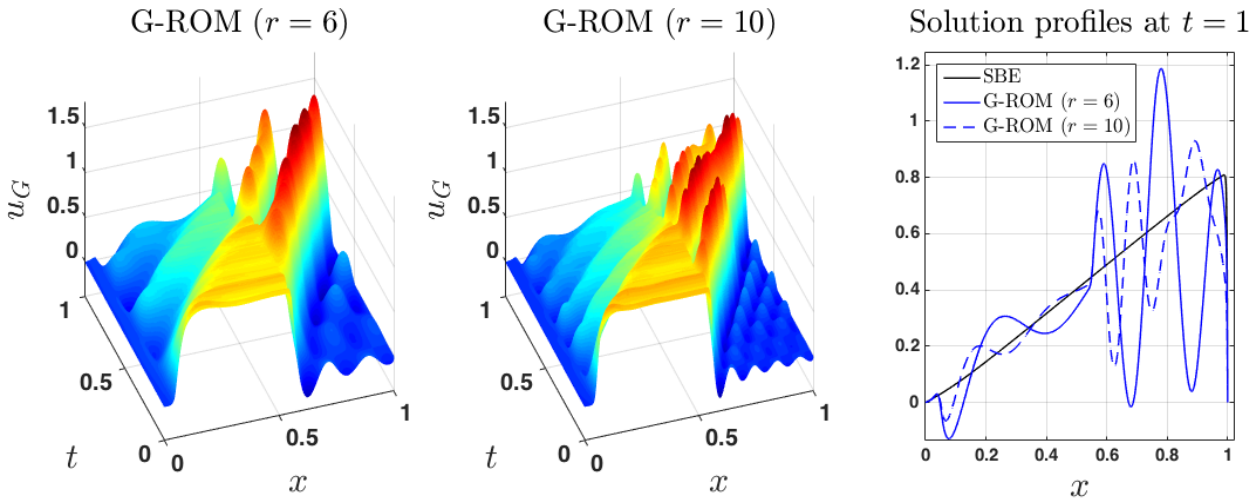


Figure 6.3: Spatio-temporal field, $u_G := \sum_{j=1}^{r} a_j \varphi_j$, reconstructed from the numerical simulation of the G-ROM (6.6) with dimension $r = 6$ (left panel) and $r = 10$ (middle panel), respectively. The noise path is the same as that used to generate the SBE's spatio-temporal field plotted in Fig. 6.1; $\sigma = 0.3$. The corresponding solution profiles at time $t = 1$ are shown in the right panel.

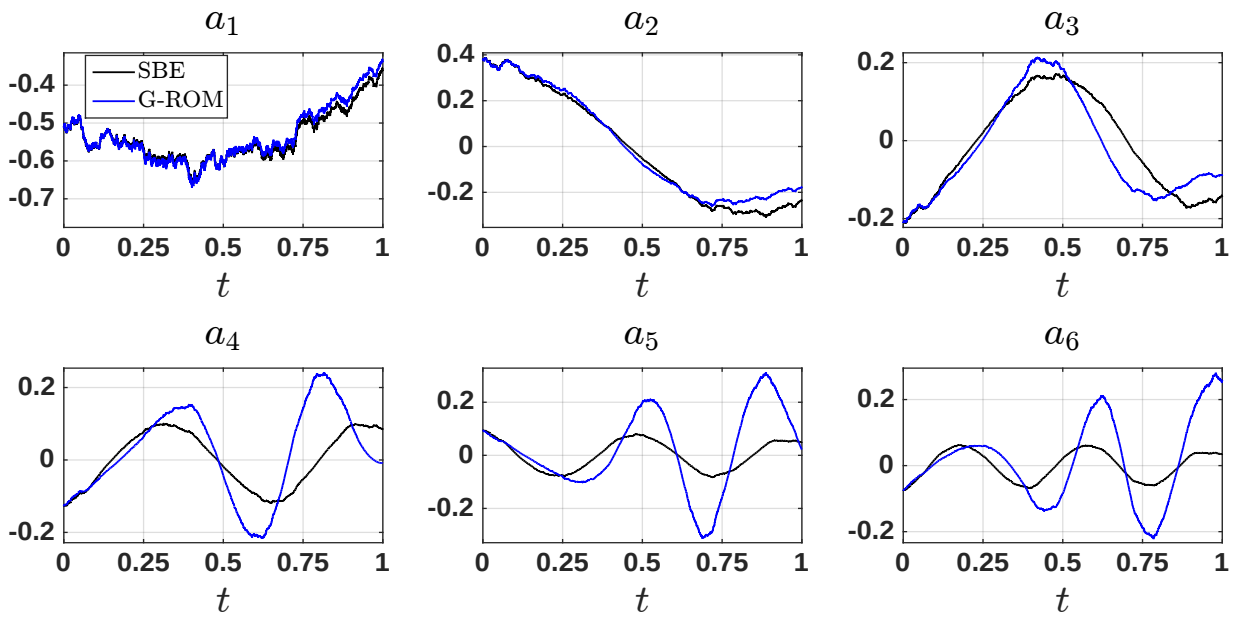


Figure 6.4: The time series a_j, $1 \le j \le r$, as modeled by the G-ROM (6.6) with dimension $r = 6$ (blue curves). Also plotted are the SBE solution projections onto the first r POD modes (black curves).

6.3.3 L-ROM Results: Alleviation of G-ROM's Spurious Oscillations

In this section, we illustrate that the G-ROM's spurious oscillations such as those illustrated in the previous section can be alleviated by using the L-ROM (6.13) derived in Section 5.3 when the spatial filtering parameter δ is appropriately calibrated; cf. (6.12).

We choose the optimal value of this free parameter δ to be the value that minimizes the L^2-error of the corresponding L-ROM in reconstructing the SBE's spatio-temporal field. In our numerical experiments, we find the optimal value by trial and error. To reduce the numerical efforts, especially in view of the statistical test given in the next section, all the numerical results related to the L-ROM (6.13) are obtained for $\delta = 0.12$, which is a nearly optimal δ value for the $r = 10$ and $\sigma = 0$ case.[2] The L-ROM results corresponding to those plotted in Figs. 6.3 and 6.4 for the G-ROM are shown in Figs. 6.5 and 6.6, respectively. As can be observed from these results, the spurious oscillations are indeed significantly reduced in the L-ROM dynamics, and an improvement in the modeling of the SBE's spatio-temporal field is also achieved.

It is also interesting to note that although the regularization used in the L-ROM successfully reduces the spurious oscillation observed in the G-ROM's high-frequency modes, it leads to a slight deterioration on the modeling of the projected dynamics onto the first POD mode as can be seen by comparing the upper left panels of Fig. 6.6 and Fig. 6.4. This

[2]We have checked that, under the parameter setting used to generate Figs. 6.5 and 6.6, the δ value we chose ($\delta = 0.12$) is close to the optimal δ values for both the $r = 6$ and $r = 10$ cases.

deterioration is also observed even if the optimal δ value is used. Of course, the deterioration is reduced when the dimension of the L-ROM is increased. We intend to further investigate this issue (together with potential L-ROM improvements) in a separate communication.

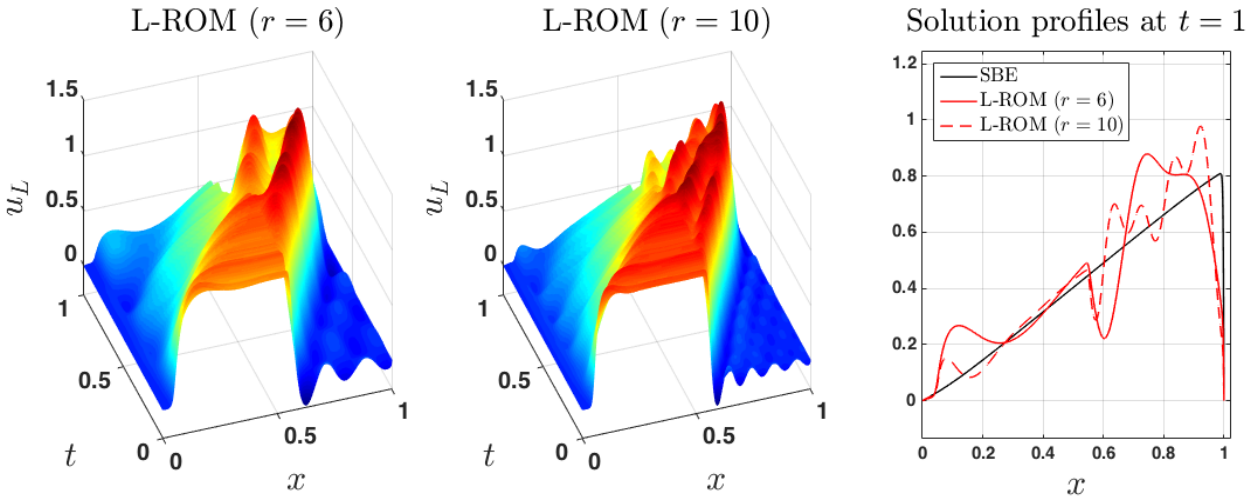


Figure 6.5: Results corresponding to Fig. 6.3 for the L-ROM (6.13), where the spatio-temporal field $u_L := \sum_{j=1}^{r} a_j \varphi_j$ is reconstructed from the numerical simulation of (6.13) with dimension $r = 6$ (left panel) and $r = 10$ (middle panel).

6.3.4 Robustness of the L-ROM results

In this section, we present some further numerical results regarding the statistical relevance of the results given in Sections 6.3.2 and 6.3.3. We also investigate the effect of the magnitude of the noise on the results.

For this purpose, the performances of the G-ROM and L-ROM are assessed by using the relative L^2-errors computed as follows:

$$E(\omega) = \frac{\sqrt{\int_0^1 \int_0^1 |u(\cdot,\cdot;\omega) - u_r(\cdot,\cdot;\omega)|^2 \, \mathrm{d}x \, \mathrm{d}t}}{\sqrt{\int_0^1 \int_0^1 |u(\cdot,\cdot;\omega)|^2 \, \mathrm{d}x \, \mathrm{d}t}} \times 100\%, \tag{6.17}$$

where for each sample path ω, $u(\cdot,\cdot;\omega)$ denotes the solution to the SBE (6.1), and $u_r(\cdot,\cdot;\omega)$ denotes the solution to either the G-ROM (6.6) or the L-ROM (6.13) with dimension r.

We consider 13 noise magnitude σ values equally spaced between 0 and 0.6. For each of these σ values, we perform 3000 numerical simulations of the fine resolution discretization of the SBE (to obtain u) and the two ROMs (to obtain u_r). The dimension of the ROMs is chosen to be $r = 10$, and the parameter δ used in the differential filter involved in the L-ROM (6.13) is fixed to be 0.12 (cf. Section 6.3.3). In Fig. 6.7, the ensemble averages of the relative errors are plotted; the error bars indicate the standard deviations.[3] This figure shows that the L-ROM is not only more accurate but also more robust to noise variations

[3]We checked that the statistical results plotted in Fig. 6.7 have already converged by comparing the results estimated from 1500 sample points of the relative errors for each of the σ values.

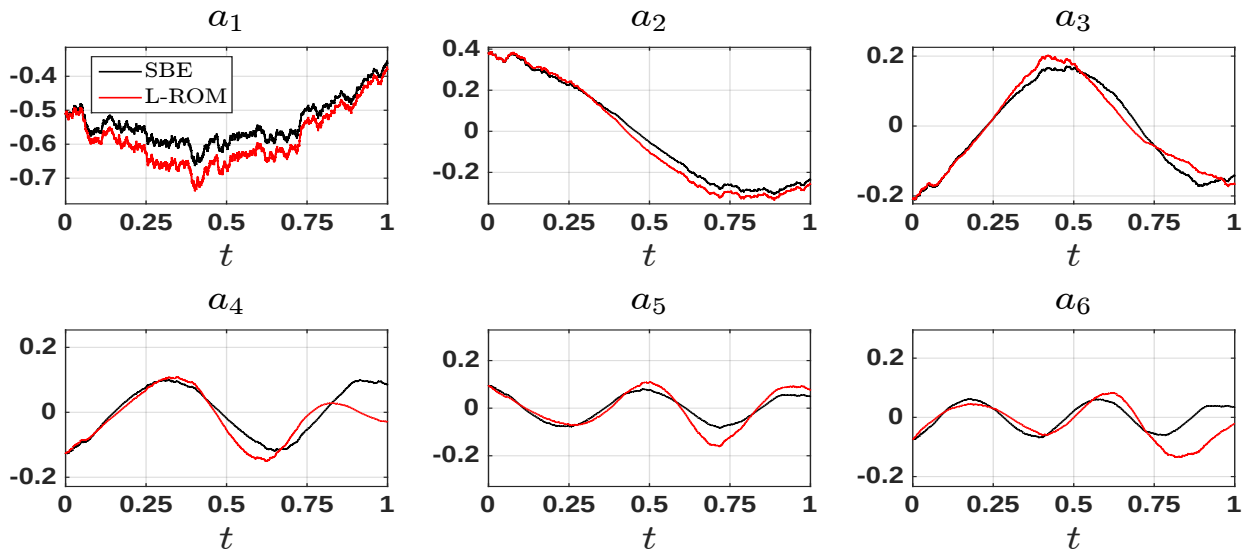


Figure 6.6: The time series a_j, $1 \leq j \leq r$, as modeled by the L-ROM (6.13) with dimension $r = 6$ (red curves). Also plotted are the SBE solution projections onto the first r POD modes (black curves).

than the G-ROM. Indeed, for the larger σ values in Fig. 6.7, the standard deviations of the relative L^2-errors associated with the G-ROM are significantly larger than those associated with the L-ROM as indicated by the corresponding error bars.

Finally, we mention that for the simulation of the G-ROM and L-ROM, instead of updating the POD basis for each σ value and for each realization of the noise, we have fixed the POD basis to be the one constructed from the spatio-temporal field of the SBE at $\sigma = 0$ (i.e., the deterministic Burgers equation). We made this choice based on the observation that the POD bases for different σ (within the explored range $[0, 0.6]$) and different noise paths actually resemble the POD basis for the $\sigma = 0$ case, which is a feature that is specific to the linear multiplicative noise. When the POD basis is updated for each noise path and each σ, we obtain results that are similar to those plotted in Fig. 6.7, although the standard deviations of the G-ROM errors are slightly reduced and the standard deviations of the L-ROM errors are slightly increased.

6.4 Summary

Numerical instability is a significant challenge for standard ROMs of deterministic convection-dominated fluid flows. A natural question is how (if at all) this challenge translates to ROMs of stochastic fluid flows. In this chapter, we took a modest step toward investigating this question by performing a computational study of the SBE (6.1) with a small diffusion coefficient ($\nu = 10^{-3}$) and in the presence of a steep internal layer. The numerical results suggested that standard (Galerkin) ROMs display spurious numerical oscillations in this convection-dominated regime. To alleviate these oscillations, we tested the L-ROM, which is a regularized ROM that uses explicit spatial filtering to smooth (regularize) the convective

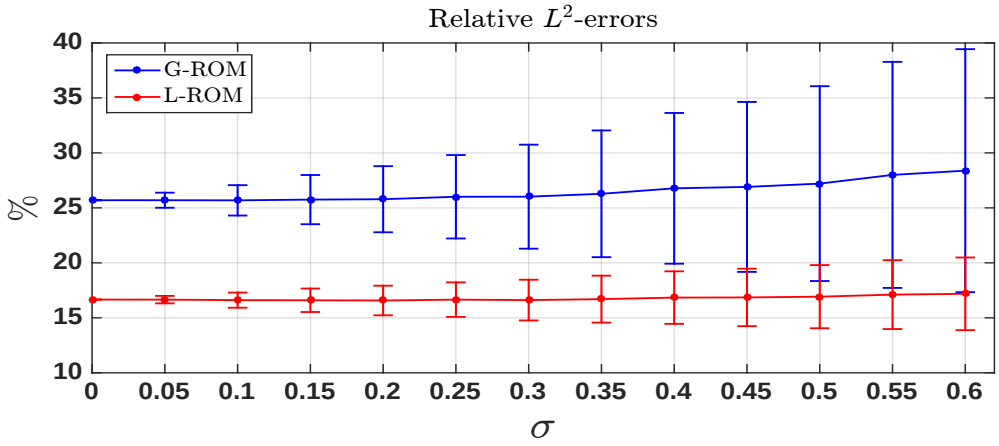


Figure 6.7: Relative L^2-errors associated with the G-ROM (6.6) and the L-ROM (6.13) as computed via (6.17) for $r = 10$. The errors are computed for 13 values of the noise amplitude parameter σ equally spaced between 0 and 0.6. An ensemble simulation of size 3000 is carried out for the SBE (6.1) and the two ROMs (6.6) and (6.13) for each σ value. The ensemble averages of the relative errors are plotted. The error bars indicate the standard deviations. The parameter δ used in the differential filter involved in the L-ROM (6.13) is fixed to be 0.12 for all the simulations.

term in the SBE. The L-ROM results were significantly more accurate than the G-ROM results. In particular, the G-ROM numerical oscillations were significantly decreased by the L-ROM; compare Figs. 6.3 and 6.4 with Figs. 6.5 and 6.6. Furthermore, the L-ROM results were less sensitive to noise magnitude variations than the G-ROM results; see Fig. 6.7.

Chapter 7

Conclusions and Future Work

The main goal of this dissertation is to develop closure models for ROMs of fluid flows, i.e., to model the interaction between resolved and unresolved ROM modes. ROM closure models are needed in realistic fluid flows, since standard ROMs that do not include closure models generally yield inaccurate results. The vast majority of ROM closure models add numerical dissipation. The argument generally used in their development is the energy cascade, which states that the main role of the unresolved ROM modes is to extract energy from the system.

In this dissertation we put forth a *fundamentally different* ROM closure philosophy that is not based on any phenomenological arguments (e.g., energy cascade). We emphasize that the ROM closure models that we propose do not include any explicit energy dissipation mechanism. We construct the new ROM closure models in two stages: First, we use explicit ROM spatial filtering to develop an LES-ROM framework in which the term that needs to be approximated (i.e., the ROM subfilter-scale stress tensor) is explicitly defined. We note that this is in contrast with the vast majority of existing ROM closure models in which the term to be approximated and the spatial filter used are generally unknown. In the second step of our new framework, we propose two novel ROM closure models: the AD-ROM and the CF-ROM. The AD-ROM employs approximate deconvolution to approximate the ROM subfilter-scale stress tensor and solve the ROM closure problem. Approximate deconovlution is an idea developed in the image processing and inverse problems communities; it is entirely mathematical and does not include any phenomenological arguments (e.g., energy cascade). In the numerical simulation of a 3D flow past a circular cylinder at Reynolds number $Re = 1000$, the AD-ROM yields accurate results at a low computational cost. The second novel ROM closure model that we propose is the CF-ROM, which is a data-driven model. The ROM closure model in the CF-ROM is constructed by solving an offline optimization problem (similar to [38, 129]) that aims at minimizing the difference between the ROM subfilter-scale stress tensor and the available full order model data. In the numerical simulation of the 1D Burgers equation with a small diffusion coefficient, the CF-ROM is as accurate as and significantly more efficient than state-of-the-art ROM closure models.

To our knowledge, there are no published results on the error analysis of ROM spatial filtering and LES-ROMs. In this dissertation, we take a first step in this direction and prove error estimates for the finite element discretization of ROM spatial filtering. We then use these error estimates to prove error bounds for the finite element discretization of the Leray ROM, which is one of the simplest spatially filtered ROMs.

Finally, we start extending our ROM spatial filtering framework from the deterministic case to the stochastic case. Specifically, we use the spatially filtered Leray ROM in the numerical simulation of a stochastic 1D Burgers equation. The numerical results show that the Leray ROM yields more accurate results than the standard (unfiltered) ROM.

Although these first results from our new LES-ROM framework and ROM closure models are extremely encouraging, there are numerous outstanding open questions. Below, we outline several research avenues that we plan to pursue.

LES-ROM Framework Open Questions We believe that the choice of the *explicit ROM spatial filter* used to develop the LES-ROM framework should be thoroughly investigated. Indeed, in the LES field, several spatial filters are commonly used, e.g., the Gaussian, box, sharp cutoff and differential filters (see [27, 139] for more details). The effect of spatial filtering on the resulting LES model is relatively well understood. In the ROM, however, to our knowledge, only two ROM spatial filters have been used so far: the ROM projection and the ROM differential filter. Their effect on the resulting ROMs has been investigated in [138, 162, 164, 165, 167]. We believe, however, that finding the best explicit ROM spatial filter for the LES-ROM framework is an important open question.

Once the ROM spatial filter is chosen, another open question for the LES-ROM framework is finding the *optimal filter radius,* δ. We emphasize that, although heuristic scalings (e.g., $\delta \sim h$, where h is the mesh size) are often used in LES, the optimal scaling is, to our knowledge, still an open question in the LES field [27, 139]. To answer this question in our LES-ROM framework, we will use the LES-ROM discretization error (which consists of the error of the standard numerical discretization, the filtering error [27, 93, 107] and the ROM truncation [100]) to develop δ scalings, which we will then numerically investigate for different test problems.

In the derivation of the LES-ROM framework in Section 2.2, we assumed that the differentiation and spatial filtering operators commute. We note that in the LES field, this assumption fails in certain situations (for example, when the radius of the spatial filter, δ, is variable) [27, 139]. In this cases, the *LES-ROM commutation error*, i.e., the error made by using the above assumption, must be addressed. Using a variable δ seems appropriate for our LES-ROM framework as well, especially when the POD modes display highly localized spatial features. Thus, we plan to investigate the magnitude of the LES-ROM commutation error when a variable δ is used. If the commutation error is relatively large compared with the discretization and modeling errors, then we will use various strategies to model it, just as in LES [27].

LES: Before or After? A natural question is whether spatial filtering and LES should be used to generate the snapshot data (i.e., before the ROM is used) or only to develop the ROM. We believe that in realistic, turbulent flow applications, spatial filtering and LES should be used both in the data generation and the ROM development (i.e., *both "before" and "after"*). The same conclusion seems to be yielded by the numerical investigation of a flow past the Ahmed body at $Re = 300,000$ in [125], in which a standard LES model (the Smagorinsky model) was used to generate the snapshots and a functional ROM closure model was utilized to stabilize the ROM. Since this work represents the first step in the investigation of the new AD-ROM, we decided to separate the spatial filtering effects in the data generation from

those in the ROM development. Thus, in Chapter 3 we only considered the "after" case. However, for the numerical simulation of realistic, high Reynolds number flows, we plan to use spatial filtering and LES both in the data generation and ROM development (i.e., both "before" and "after").

AD-ROM Open Questions The choice of other *regularization methods* in the AD-ROM should be carefully investigated. Indeed, as a first step, in Chapter 3 we considered only one of the simplest regularization method, the Lavrentiev regularization. We plan, however, to investigate other regularization strategies, both variational and iterative [28, 81, 158].

Probably the most important research direction that we plan to pursue is to investigate the AD-ROM in the numerical simulation of *realistic turbulent flows that display recurrent structures* (see, e.g., [124, 125]). We emphasize, however, that in these challenging settings, the current AD-ROM should probably be supplemented with a *numerical dissipation mechanism* (e.g., time relaxation [107]), just as in LES [148, 149, 150, 151]. In this dissertation we investigated the AD-ROM without any additional numerical dissipation mechanism. We believe that decoupling the AD and dissipation mechanisms was necessary in order to understand the effect of the former, which, to our knowledge, is new in a ROM setting. We note, however, that for higher Reynolds number flows, adding numerical dissipation would probably be needed in order to account for the subgrid-scale effects due to the inherently coarse meshes used in the data generation.

In our numerical investigation, the accuracy of the new AD-ROM was similar to that of the EV-ROMs [162] and Reg-ROMs [165]. We believe that this is impressive, given that the new AD-ROM did not use any explicit numerical dissipation mechanisms, whereas the other ROMs did. For high Reynolds number flows, we plan to add numerical dissipation (e.g., time relaxation) to the AD-ROM and perform a *thorough comparison with other types of ROMs* (e.g, EV-ROMs [12, 85, 162], Reg-ROMs [138, 165] and calibrated ROMs [6, 54, 160, 163]).

CF-ROM Open Questions Although the preliminary results are encouraging, the new CF-ROM framework's full potential still needs to be explored. Next, we outline several research directions that could be pursued.

Quadratic Ansatz: In the CF-ROM's numerical investigation in Section 4.5, we used the linear ansatz (i.e., we considered $\widetilde{B}_{imn} = 0$ in (4.32)) to model the ROM stress tensor $\boldsymbol{\tau}_r$. We emphasize, however, that from a physical point of view, the quadratic ansatz (i.e., $\widetilde{B}_{imn} \neq 0$ in (4.32)) is more appropriate. Indeed, the ROM stress tensor $\boldsymbol{\tau}_r$ is closely connected to the quadratic nonlinearity in the NSE. Thus, we expect the ROM stress tensor be best approximated by a quadratic term, too, just as in the quadratic ansatz. In fact, given the low physical accuracy of the linear ansatz used in the numerical investigation in Section 4.5, the fact the CF-ROM results were qualitatively comparable to those from state-of-the-art LES-ROMs is impressive.

CF-ROM's Potential Generality: Although the particular form of the CF-ROM will change with the underlying physical system, certain classes of physical systems might yield similar types of CF-ROMs. For example, for some classes of systems the linear ansatz could suffice, whereas for some other classes of systems the quadratic ansatz could be needed. Furthermore, for certain classes of systems, one could determine variation intervals for the

CF-ROM coefficients.

CF-ROM's Universality: We constructed and tested the CF-ROM in a fluid dynamics setting: We used the NSE to build the CF-ROM and the Burgers equation to numerically investigate it. We emphasize, however, that the CF-ROM framework can be applied to any type of nonlinear PDE that is amenable to reduced order modeling. Indeed, the only input needed in the CF-ROM framework is the FOM data. Once those are supplied, the CF-ROM proceeds in two steps. (i) First, the given nonlinear PDE is spatially filtered. The nonlinearity yields a nonlinear stress tensor (which will generally be different from the stress tensor $\boldsymbol{\tau}_r$ used in this chapter). (ii) In the second step of the CF-ROM construction, the available FOM data is used to compute an approximation for the true stress tensor in the filtered ROM in (i) and an optimization problem is employed to calibrate the CF-ROM accordingly. We emphasize again that the entire CF-ROM procedure does not use any phenomenological arguments that would restrict it to the particular physical system modeled by the given nonlinear PDE. This is in stark contrast with, e.g., ROM closure models of eddy viscosity type [85, 125, 162], which cannot be directly applied to other classes of PDEs. Since the CF-ROM is built upon general principles (i.e., filtering and calibration), we expect it to be successful in the numerical simulation of general mathematical models (e.g., from elasticity or bioengineering).

Spatial Filtering: In Section 4.2, we used the ROM projection (4.3) as a ROM spatial filter to construct the F-ROM (4.9). This choice of ROM spatial filter was essential for being able to write the F-ROM decomposition in (4.11) and to explain why most ROM closure models amount to adding extra terms to the standard G-ROM. (Indeed, the ROM projection yielded identity (4.8), which in turn allowed the formal F-ROM decomposition (4.11).) We emphasize, however, that other ROM spatial filters could be used in the new CF-ROM framework. For example, the ROM differential filter (which was successfully used in developing LES-ROMs [165, 168]) could be used as a ROM spatial filter to construct the F-ROM (4.9).

Stochastic ROM Open Questions We emphasize that much more remains to be done for a clear understanding of the potential numerical instability of ROMs and possible remedies of such instability for convection-dominated stochastic flows. For example, it is interesting to explore whether the results of this computational study are valid for other types of noise (e.g, additive noise or correlated additive and multiplicative noise) and more realistic settings (e.g., 3D fluid flows modeled by the NSE). Furthermore, it is also interesting to investigate the performance of other regularized ROMs (e.g., the evolve-then-filter ROM [165]) and stabilized ROMs.

Another important research direction is the investigation of the robustness of the proposed ROMs. For example, it would be interesting to use more systematic approaches (such as numerical analysis [76]) to determine general scalings for the model parameters, such as the spatial filtering parameter δ used in the L-ROM. One could also perform thorough sensitivity studies of these ROMs with respect to model parameters, such as the number of basis functions (r) or the filtering parameter (δ).

Bibliography

[1] B. Abou El Majd and L. Cordier. New regularization method for calibrated POD reduced-order models. *Math. Model. Anal.*, 21(1):47–62, 2016.

[2] I. Akhtar. *Parallel Simulation, Reduced-Order Modeling, and Feedback Control of Vortex Shedding using Fluidic Actuators*. PhD thesis, Virginia Tech, 2008.

[3] I. Akhtar, J. Borggaard, J. A. Burns, H. Imtiaz, and L. Zietsman. Using functional gains for effective sensor location in flow control: a reduced-order modelling approach. *J Fluid Mech.*, 781:622–656, 2015.

[4] I. Akhtar, A. H. Nayfeh, and C. J. Ribbens. On the stability and extension of reduced-order Galerkin models in incompressible flows. *Theor. Comp. Fluid Dyn.*, 23(3):213–237, 2009.

[5] A. Alabert and I. Gyöngy. On numerical approximation of stochastic Burgers' equation. In *From Stochastic Calculus to Mathematical Finance*, pages 1–15. Springer, Berlin, 2006.

[6] A. K. Alekseev and I. M. Navon. The analysis of an ill-posed problem using multi-scale resolution and second-order adjoint techniques. *Comput. Meth. Appl. Mech. Eng.*, 190(15):1937–1953, 2001.

[7] D. Amsallem and C. Farhat. Stabilization of projection-based reduced-order models. *Int. J. Num. Meth. Eng.*, 91(4):358–377, 2012.

[8] A. C. Antoulas. *Approximation of large-scale dynamical systems*, volume 6. SIAM, 2005.

[9] A. C. Antoulas, C. A. Beattie, and S. Gugercin. Interpolatory model reduction of large-scale dynamical systems. In *Efficient modeling and control of large-scale systems*, pages 3–58. Springer, 2010.

[10] S. Aradag, S. Siegel, J. Seidel, K. Cohen, and T. McLaughlin. Filtered POD-based low-dimensional modeling of the 3D turbulent flow behind a circular cylinder. *Int. J. Num. Meth. Fluids*, 66(1):1–16, 2011.

[11] Uri M Ascher and Linda R Petzold. *Computer methods for ordinary differential equations and differential-algebraic equations*, volume 61. Siam, 1998.

[12] N. Aubry, P. Holmes, J. L. Lumley, and E. Stone. The dynamics of coherent structures in the wall region of a turbulent boundary layer. *J. Fluid Mech.*, 192:115–173, 1988.

[13] M. J. Balajewicz, E. H. Dowell, and B. R. Noack. Low-dimensional modelling of high-Reynolds-number shear flows incorporating constraints from the Navier–Stokes equation. *J. Fluid Mech.*, 729:285–308, 2013.

[14] M. J. Balajewicz, I. Tezaur, and E. H. Dowell. Minimal subspace rotation on the Stiefel manifold for stabilization and enhancement of projection-based reduced order models for the compressible Navier–Stokes equations. *J. Comput. Phys.*, 321:224–241, 2016.

[15] F. Ballarin, E. Faggiano, S. Ippolito, A. Manzoni, A. Quarteroni, G. Rozza, and R. Scrofani. Fast simulations of patient-specific haemodynamics of coronary artery bypass grafts based on a POD–Galerkin method and a vascular shape parametrization. *J. Comput. Phys.*, 315:609–628, 2016.

[16] F. Ballarin, A. Manzoni, A. Quarteroni, and G. Rozza. Supremizer stabilization of POD–Galerkin approximation of parametrized steady incompressible Navier–Stokes equations. *Int. J. Numer. Meth. Engng.*, 102:1136–1161, 2015.

[17] R. E. Bank and T. Dupont. An optimal order process for solving finite element equations. *Math. Comput.*, 36(153):35–51, 1981.

[18] M. F. Barone, I. Kalashnikova, D. J. Segalman, and H. K. Thornquist. Stable Galerkin reduced order models for linearized compressible flow. *J. Comput. Phys.*, 228(6):1932–1946, 2009.

[19] U. Baur, P. Benner, and L. Feng. Model order reduction for linear and nonlinear systems: a system-theoretic perspective. *Arch. Comput. Method. E.*, 21(4):331–358, 2014.

[20] J. Bec and K. Khanin. Burgers turbulence. *Phys. Rep.*, 447:1–66, 2007.

[21] J. R. Beddington and R. M. May. Harvesting natural populations in a randomly fluctuating environment. *Science*, 197(4302):463–465, 1977.

[22] P. Benner and T. Breiten. Two-sided projection methods for nonlinear model order reduction. *SIAM J. Sci. Comput.*, 37(2):B239–B260, 2015.

[23] P. Benner, S. Gugercin, and K. Willcox. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Rev.*, 57(4):483–531, 2015.

[24] Peter Benner, Pawan Goyal, and Serkan Gugercin. h_2-quasi-optimal model order reduction for quadratic-bilinear control systems. *arXiv preprint arXiv:1610.03279*, 2016.

[25] M. Benosman, J. Borggaard, and B. Kramer. Robust reduced-order model stabilization for partial differential equations based on Lyapunov theory and extremum seeking with application to the 3D Boussinesq equations. *arXiv preprint* ***http://arxiv.org/abs/1604.04586***, 2016.

[26] M. Bergmann, C. H. Bruneau, and A. Iollo. Enablers for robust POD models. *J. Comput. Phys.*, 228(2):516–538, 2009.

[27] L. C. Berselli, T. Iliescu, and W. J. Layton. *Mathematics of Large Eddy Simulation of Turbulent Flows*. Scientific Computation. Springer-Verlag, Berlin, 2006.

[28] M. Bertero and P. Boccacci. *Introduction to Inverse Problems in Imaging*. Institute of Physics Publishing, Bristol, 1998.

[29] B. Birnir. *The Kolmogorov-Obukhov Theory of Turbulence: A mathematical theory of turbulence*. Springer Briefs in Mathematics. Springer, New York, 2013.

[30] D. Blömker. *Amplitude Equations for Stochastic Partial Differential Equations*, volume 3 of *Interdisciplinary Mathematical Sciences*. World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2007.

[31] D. Blömker and A. Jentzen. Galerkin approximations for the stochastic Burgers equation. *SIAM J. Numer. Anal.*, 51(1):694–715, 2013.

[32] J. Borggaard, T. Iliescu, and Z. Wang. Artificial viscosity proper orthogonal decomposition. *Math. Comput. Modelling*, 53(1-2):269–279, 2011.

[33] S. Boyaval, C. Le Bris, T. Lelièvre, Y. Maday, N. C. Nguyen, and A. T. Patera. Reduced basis techniques for stochastic problems. *Arch. Comput. methods Eng.*, 17:435–454, 2010.

[34] S. Boyaval, C. Le Bris, Y. Maday, N. C. Nguyen, and A. T. Patera. A reduced basis approach for variational problems with stochastic parameters: Application to heat conduction with variable Robin coefficient. *Comput. Methods Appl. Mech. Engrg.*, 198:3187–3206, 2009.

[35] Kathryn Eleda Brenan, Stephen L Campbell, and Linda Ruth Petzold. *Numerical solution of initial-value problems in differential-algebraic equations*. SIAM, 1995.

[36] S. C. Brenner and L. R. Scott. *The mathematical theory of finite element methods*, volume 15 of *Texts in Applied Mathematics*. Springer-Verlag, New York, 1994.

[37] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Compressive sampling and dynamic mode decomposition. *arXiv preprint arXiv:1312.5186*, 2013.

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

[39] M. Buffoni, S. Camarri, A. Iollo, and M. V. Salvetti. Low-dimensional modelling of a confined three-dimensional wake flow. *J. Fluid Mech.*, 569:141–150, 2006.

[40] J. Burkardt, M. Gunzburger, and C. Webster. Reduced order modeling of some nonlinear stochastic partial differential equations. *Inter. J. of Num. Anal. and Modeling*, 4(3-4):368–391, 2007.

[41] John Charles Butcher. *Numerical methods for ordinary differential equations*. John Wiley & Sons, 2016.

[42] A. Caiazzo, T. Iliescu, V. John, and S. Schyschlowa. A numerical investigation of velocity-pressure reduced order models for incompressible flows. *J. Comput. Phys.*, 259:598–616, 2014.

[43] K. Carlberg, M. Barone, and H. Antil. Galerkin v. least-squares Petrov–Galerkin projection in nonlinear model reduction. *J. Comput. Phys.*, 330:693–734, 2017.

[44] K. Carlberg, C. Farhat, J. Cortial, and D. Amsallem. The GNAT method for nonlinear model reduction: effective implementation and application to computational fluid dynamics and turbulent flows. *J. Comput. Phys.*, 2013.

[45] W. Cazemier, R. Verstappen, and AEP Veldman. Proper orthogonal decomposition and low-dimensional models for driven cavity flows. *Phys. Fluids*, 10:1685, 1998.

[46] D. Chapelle, A. Gariah, and J. Sainte-Marie. Galerkin approximation with proper orthogonal decomposition: new error estimates and illustrative examples. *ESAIM: Math. Model. Numer. Anal.*, 46:731–757, 2012.

[47] M. D. Chekroun and H. Liu. Finite-horizon parameterizing manifolds, and applications to suboptimal control of nonlinear parabolic PDEs. *Acta Applicandae Mathematicae*, 135(1):81–144, 2015.

[48] M. D. Chekroun, H. Liu, and S. Wang. *Approximation of Invariant Manifolds: Stochastic Manifolds for Nonlinear SPDEs I*. SpringerBriefs in Mathematics. Springer, New York, 2015.

[49] M. D. Chekroun, H. Liu, and S. Wang. *Parameterizing Manifolds and Non-Markovian Reduced Equations: Stochastic Manifolds for Nonlinear SPDEs II*. SpringerBriefs in Mathematics. Springer, New York, 2015.

[50] M. D. Chekroun and L. J. Roques. Models of population dynamics under the influence of external perturbations: Mathematical results. *C. R. Math. Acad. Sci. Paris*, 343(5):307–310, 2006.

[51] P. Chen and A. Quarteroni. A new algorithm for high-dimensional uncertainty quantification based on dimension-adaptive sparse grid approximation and reduced basis methods. *Journal of Computational Physics*, 298:176–193, 2015.

[52] P. Chen, A. Quarteroni, and G. Rozza. A weighted reduced basis method for elliptic partial differential equations with random input data. *SIAM Journal on Numerical Analysis*, 51(6):3163–3185, 2013.

[53] P. Chen and C. Schwab. Sparse-grid, reduced-basis bayesian inversion. *Journal of Computational Physics*, 316:470–503, 2016.

[54] L. Cordier, B. Abou El Majd, and J. Favier. Calibration of POD reduced-order models using Tikhonov regularization. *Int. J. Num. Meth. Fluids*, 63(2):269–296, 2010.

[55] M. Couplet, C. Basdevant, and P. Sagaut. Calibrated reduced-order POD-Galerkin system for fluid flow modelling. *J. Comput. Phys.*, 207(1):192–220, 2005.

[56] M. Couplet, C. Basdevant, and P. Sagaut. Calibrated reduced-order POD-Galerkin system for fluid flow modelling. *J. Comput. Phys.*, 207(1):192–220, 2005.

[57] M. Couplet, P. Sagaut, and C. Basdevant. Intermodal energy transfers in a proper orthogonal decomposition–Galerkin representation of a turbulent separated flow. *J. Fluid Mech.*, 491:275–284, 2003.

[58] M. C. Cross and P. C. Hohenberg. Pattern formation outside of equilibrium. *Rev. Mod. Phys.*, 65(3):851–1112, 1993.

[59] D. N. Daescu and I. M. Navon. A dual-weighted approach to order reduction in 4DVAR data assimilation. *Mon. Weather Rev.*, 136(3):1026–1041, 2008.

[60] G. DE STEFANO and OV VASILYEV. Stochastic coherent adaptive large eddy simulation of forced isotropic turbulence. *Journal of Fluid Mechanics*, 646(1):453–470, 2010.

[61] A. A. Dunca and M. Neda. On the Vreman filter based stabilization for the advection equation. *Appl. Math. Comput.*, 269:379–388, 2015.

[62] A. A. Dunca, M. Neda, and L. G. Rebholz. A mathematical and numerical study of a filtering-based multiscale fluid model with nonlinear eddy viscosity. *Comput. Math. Appl.*, 66(6):917–933, 2013.

[63] A. Ern and J.-L. Guermond. *Theory and Practice of Finite Elements*, volume 159 of *Applied Mathematical Sciences*. Springer-Verlag, New York, 2004.

[64] C. Foiaş, O. Manley, R. Rosa, and R. Temam. *Navier Stokes Equations and Turbulence*. Cambridge University Press, 2001.

[65] D. Galbally, K. Fidkowski, K. Willcox, and O. Ghattas. Non-linear model reduction for uncertainty quantification in large-scale inverse problems. *Int. J. Numer. Meth. Engng.*, 81:1581–1608, 2010.

[66] B.G. Galerkin. Rods and plates. Series on some problems of elastic equilibrium of rods and plates (in Russian). *Vest. Inzh. Tech.*, 19:897–908, 1915.

[67] B. Galletti, A. Bottaro, C.-H. Bruneau, and A. Iollo. Accurate model reduction of transient and forced wakes. *Eur. J. Mech. B-Fluid.*, 26(3):354–366, 2007.

[68] B. Galletti, C. H. Bruneau, L. Zannetti, and A. Iollo. Low-order modelling of laminar flow regimes past a confined square cylinder. *J. Fluid Mech.*, 503:161–170, 2004.

[69] B. Galletti, C. H. Bruneau, L. Zannetti, and A. Iollo. Low-order modelling of laminar flow regimes past a confined square cylinder. *J. Fluid Mech.*, 503:161–170, 2004.

[70] K. J. Galvin, L. G. Rebholz, and C. Trenchea. Efficient, unconditionally stable, and optimally accurate FE algorithms for approximate deconvolution models. *SIAM J. Numer. Anal.*, 52(2):678–707, 2014.

[71] M. Germano. Differential filters for the large eddy numerical simulation of turbulent flows. *Phys. Fluids*, 29(6):1755–1757, 1986.

[72] M. Germano. Differential filters of elliptic type. *Phys. Fluids*, 29(6):1757–1758, 1986.

[73] M. Germano, U. Piomelli, P. Moin, and W.H. Cabot. A dynamic subgrid-scale eddy viscosity model. *Phys. Fluids A*, 3:1760–1765, 1991.

[74] B. J. Geurts and D. D. Holm. Regularization modeling for large-eddy simulation. *Phys. Fluids*, 15(1):L13–L16, 2003.

[75] B.J. Geurts. *Elements of Direct and Large-Edddy Simulation*. R.T. Edwards, Inc, 2003.

[76] S. Giere, T. Iliescu, V. John, and D. Wells. SUPG reduced order models for convection-dominated convection-diffusion-reaction equations. *Comput. Methods Appl. Mech. Engrg.*, 289:454–474, 2015.

[77] A. Gouasmi, E. Parish, and K. Duraisamy. Characterizing memory effects in coarse-grained nonlinear systems using the Mori-Zwanzig formalism. *arXiv preprint, http://arxiv.org/abs/1611.06277*, 2016.

[78] W. R. Graham, J. Peraire, and K. Y. Tang. Optimal control of vortex shedding using low-order models. Part I – open-loop model development. *Int. J. Numer. Meth. Eng.*, 44(7):945–972, 1999.

[79] P.M. Gresho and R.L. Sani. *Incompressible Flow and the Finite Element Method, Isothermal Laminar Flow*, volume 2. John Wiley & Sons, 2000. Paperback.

[80] B. Haasdonk, K. Urban, and B. Wieland. Reduced basis methods for parameterized partial differential equations with stochastic influences using the Karhunen–Loéve expansion. *SIAM/ASA Journal on Uncertainty Quantification*, 1(1):79–105, 2013.

[81] P. C. Hansen. *Discrete Inverse Problems: Insight and Algorithms*, volume 7. Society for Industrial and Applied Mathematics, 2010.

[82] J. S. Hesthaven, G. Rozza, and B. Stamm. *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*. Springer, 2015.

[83] G. Hetzer, W. Shen, and S. Zhu. Asymptotic behavior of positive solutions of random and stochastic parabolic equations of Fisher and Kolmogorov types. *J. Dyn. Diff. Equat.*, 14(1):139–188, 2002.

[84] J. G. Heywood and R. Rannacher. Finite-element approximation of the nonstationary Navier-Stokes problem. Part IV: Error analysis for second-order time discretization. *SIAM J. Numer. Anal.*, 27(2):353–384, 1990.

[85] P. Holmes, J. L. Lumley, and G. Berkooz. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*. Cambridge, 1996.

[86] T. Y. Hou, W. Luo, B. Rozovskii, and H.-M. Zhou. Wiener chaos expansions and numerical solutions of randomly forced equations of fluid mechanics. *J. Comput. Phys.*, 216(2):687–706, 2006.

[87] T. J. R. Hughes, G. R. Feijóo, L. Mazzei, and J.-B. Quincy. The variational multiscale methodâĂŤa paradigm for computational mechanics. *Comput. Methods Appl. Mech. Engrg.*, 166(1):3–24, 1998.

[88] T. Iliescu, H. Liu, and X. Xie. Regularized reduced order models for a stochastic Burgers equation. 2017. submitted.

[89] T. Iliescu and Z. Wang. Variational multiscale proper orthogonal decomposition: Convection-dominated convection-diffusion-reaction equations. *Math. Comput.*, 82(283):1357–1378, 2013.

[90] T. Iliescu and Z. Wang. Are the snapshot difference quotients needed in the proper orthogonal decomposition? *SIAM J. Sci. Comput.*, 36(3):A1221–A1250, 2014.

[91] T. Iliescu and Z. Wang. Variational multiscale proper orthogonal decomposition: Navier-Stokes equations. *Num. Meth. P.D.E.s*, 30(2):641–663, 2014.

[92] A. Jentzen and P. E. Kloeden. *Taylor Approximations for Stochastic Partial Differential Equations*, volume 83 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. SIAM, Philadelphia, PA, 2011.

[93] V. John. *Large Eddy Simulation of Turbulent Incompressible Flows*, volume 34 of *Lecture Notes in Computational Science and Engineering*. Springer-Verlag, Berlin, 2004. Analytical and Numerical Results for a Class of LES Models.

[94] I. Kalashnikova and M. F. Barone. On the stability and convergence of a Galerkin reduced order model (ROM) of compressible flow with solid wall and far-field boundary treatment. *Int. J. Num. Meth. Eng.*, 83(10):1345–1375, 2010.

[95] S. Kaya and C. C. Manica. Convergence analysis of the finite element method for a fundamental model in turbulence. *Math. Models Methods Appl. Sci.*, 22(11):1250033, 2012.

[96] P. E. Kloeden and E. Platen. *Numerical Solution of Stochastic Differential Equations*, volume 23 of *Applications of Mathematics*. Springer-Verlag, Berlin, 1992.

[97] T. Kostova, G. Oxberry, K. Chand, and W. Arrighi. Error bounds and analysis of proper orthogonal decomposition model reduction methods using snapshots from the solution and the time derivatives. *arXiv preprint,http://arxiv.org/abs/1501.02004*, 2015.

[98] A. G. Kravchenko and P. Moin. Numerical studies of flow over a circular cylinder at $Re_D = 3900$. *Phys. Fluids*, 12(2):403–417, 2000.

[99] K. Kunisch and S. Volkwein. Control of the Burgers equation by a reduced-order approach using proper orthogonal decomposition. *J. Optim. Theory Appl.*, 102(2):345–371, 1999.

[100] K. Kunisch and S. Volkwein. Galerkin proper orthogonal decomposition methods for parabolic problems. *Numer. Math.*, 90(1):117–148, 2001.

[101] K. Kunisch and S. Volkwein. Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. *SIAM J. Numer. Anal.*, 40(2):492–515 (electronic), 2002.

[102] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor. *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems*. SIAM, 2016.

[103] T. Lassila, A. Manzoni, A. Quarteroni, and G. Rozza. A reduced computational and geometrical framework for inverse problems in hemodynamics. *International journal for numerical methods in biomedical engineering*, 29(7):741–776, 2013.

[104] T. Lassila, A. Manzoni, A. Quarteroni, and G. Rozza. Model order reduction in fluid dynamics: challenges and perspectives. In *Reduced Order Methods for Modeling and Computational Reduction*, pages 235–273. Springer, 2014.

[105] W. Layton, C. C. Manica, M. Neda, and L. G. Rebholz. Numerical analysis and computational testing of a high accuracy Leray-deconvolution model of turbulence. *Num. Meth. P.D.E.s*, 24(2):555–582, 2008.

[106] W. J. Layton. *Introduction to the numerical analysis of incompressible viscous flows*, volume 6. Society for Industrial and Applied Mathematics (SIAM), 2008.

[107] W. J. Layton and L. G. Rebholz. *Approximate Deconvolution Models of Turbulence: Analysis, Phenomenology and Numerical Analysis*, volume 2042. Springer, 2012.

[108] J. Leray. Sur le mouvement d'un fluide visqueux emplissant l'espace. *Acta Math.*, 63:193–248, 1934.

[109] G. J. Lord and J. Rougemont. A numerical scheme for stochastic PDEs with Gevrey regularity. *IMA J. Numer. Anal.*, 24(4):587–604, 2004.

[110] S. Lorenzi, A. Cammi, L. Luzzi, and G. Rozza. POD-Galerkin method for finite volume approximation of Navier-Stokes and RANS equations. *Comput. Methods Appl. Mech. Engrg.*, 2016.

[111] Z. Luo, J. Chen, I. M. Navon, and X. Yang. Mixed finite element formulation and error estimates based on proper orthogonal decomposition for the nonstationary Navier-Stokes equations. *SIAM J. Numer. Anal.*, 47(1):1–19, 2008.

[112] X. Xie T. Iliescu M. Mohebujjaman, L. G. Rebholz. Energy Balance and Mass Conservation in Reduced Order Models of Fluid Flows, 2017.

[113] C. C. Manica and S. Kaya Merdan. Finite element error analysis of a zeroth order approximate deconvolution model based on a mixed formulation. *J. Math. Anal. Appl.*, 331(1):669–685, 2007.

[114] C. Mueller, L. Mytnik, and J. Quastel. Small noise asymptotics of traveling waves. *Markov Process. Related Fields*, 14(3):333–342, 2008.

[115] C. Mueller, L. Mytnik, and J. Quastel. Effect of noise on front propagation in reaction-diffusion equations of KPP type. *Inventiones Math.*, 184(2):405–453, 2011.

[116] M. A. Muñoz. Multiplicative noise in non-equilibrium phase transitions: A tutorial. In *Advances in Condensed Matter and Statistical Physics*, pages 37–68. Nova Science Publishers, Inc., 2004.

[117] N. Nguyen, G. Rozza, and A. T. Patera. Reduced basis approximation and a posteriori error estimation for the time-dependent viscous BurgersâĂŹ equation. *Calcolo*, 46(3):157–185, 2009.

[118] B. R. Noack, M. Morzynski, and G. Tadmor. *Reduced-Order Modelling for Flow Control*, volume 528. Springer Verlag, 2011.

[119] B. R. Noack, P. Papas, and P. A. Monkewitz. Low-dimensional Galerkin model of a laminar shear-layer. Technical Report 2002-01, École Polytechnique Fédérale de Lausanne, 2002.

[120] B. R. Noack, P. Papas, and P. A. Monkewitz. The need for a pressure-term representation in empirical Galerkin models of incompressible shear flows. *J. Fluid Mech.*, 523:339–365, 2005.

[121] B. R. Noack, M. Schlegel, B. Ahlborn, G. Mutschke, M. Morzynski, P. Comte, and G. Tadmor. A finite-time thermodynamics of unsteady fluid flows. *J. Non-Equil. Thermody.*, 33(2):103–148, 2008.

[122] J. Nocedal and S. Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[123] B. Øksendal. *Stochastic Differential Equations: An Introduction with Applications*. Springer-Verlag, Berlin, 6th edition, 2003.

[124] J. Östh. *Unsteady Numerical Simulations and Reduced-Order Modelling of Flows around Vehicles*. PhD thesis, Chalmers University of Technology, 2014.

[125] J. Östh, B. R. Noack, S. Krajnović, D. Barros, and J. Borée. On the need for a nonlinear subscale turbulence term in POD models as exemplified for a high-Reynolds-number flow over an Ahmed body. *J. Fluid Mech.*, 747:518–544, 2014.

[126] P. Pacciarini and G. Rozza. Stabilized reduced basis method for parametrized advection–diffusion PDEs. *Comput. Meth. Appl. Mech. Eng.*, 274:1–18, 2014.

[127] Eric J Parish and Karthik Duraisamy. A paradigm for data-driven predictive modeling using field inversion and machine learning. *Journal of Computational Physics*, 305:758–774, 2016.

[128] B. Peherstorfer and K. Willcox. Dynamic data-driven reduced-order models. *Comput. Methods Appl. Mech. Engrg.*, 291:21–41, 2015.

[129] Benjamin Peherstorfer and Karen Willcox. Data-driven operator inference for nonintrusive projection-based model reduction. *Computer Methods in Applied Mechanics and Engineering*, 306:196–215, 2016.

[130] O. Pironneau. *Finite element methods for fluids*. John Wiley & Sons Ltd., Chichester, 1989. Translated from the French.

[131] B. Protas, B. R. Noack, and J. Östh. Optimal nonlinear eddy viscosity in Galerkin models of turbulent flows. *J. Fluid Mech.*, 766:337–367, 2015.

[132] A. Quarteroni, A. Manzoni, and F. Negri. *Reduced Basis Methods for Partial Differential Equations: An Introduction*, volume 92. Springer, 2015.

[133] A. Quarteroni, G. Rozza, and A. Manzoni. Certified reduced basis approximation for parametrized partial differential equations and applications. *J. Math. Ind.*, 1(1):1–49, 2011.

[134] T. Chacón Rebollo and R. Lewandowski. *Mathematical and Numerical Foundations of Turbulence Models and Applications*. Springer, 2014.

[135] L. Roques and M. D. Chekroun. On population resilience to external perturbations. *SIAM J. Appl. Math.*, 68(1):133–153, 2007.

[136] L. Roques and M. D. Chekroun. Does reaction-diffusion support the duality of fragmentation effect? *Ecological Complexity*, 7(1):100–106, 2010.

[137] C. W. Rowley and S. T. M. Dawson. Model reduction for flow analysis and control. *Ann. Rev. Fluid Mech.*, 49:387–417, 2017.

[138] F. Sabetghadam and A. Jafarpour. α regularization of the POD-Galerkin dynamical systems of the Kuramoto–Sivashinsky equation. *Appl. Math. Comput.*, 218(10):6012–6026, 2012.

[139] P. Sagaut. *Large Eddy Simulation for Incompressible Flows*. Scientific Computation. Springer-Verlag, Berlin, third edition, 2006.

[140] O. San and T. Iliescu. Proper orthogonal decomposition closure models for fluid flows: Burgers equation. *Int. J. Numer. Anal. Mod., Series B*, 5(3):285–305, 2014.

[141] O. San and T. Iliescu. A stabilized proper orthogonal decomposition reduced-order model for large scale quasigeostrophic ocean circulation. *Adv. Comput. Math.*, pages 1289–1319, 2015.

[142] P. J. Schmid. Dynamic mode decomposition of numerical and experimental data. *J. Fluid Mech.*, 656:5–28, 2010.

[143] J. R. Singler. New POD error expressions, error bounds, and asymptotic results for reduced order models of parabolic PDEs. *SIAM J. Numer. Anal.*, 52(2):852–876, 2014.

[144] S. Sirisup and G. E. Karniadakis. A spectral viscosity method for correcting the long-term behavior of POD models. *J. Comput. Phys.*, 194(1):92–116, 2004.

[145] L. Sirovich. Turbulence and the dynamics of coherent structures. Parts I–III. *Quart. Appl. Math.*, 45(3):561–590, 1987.

[146] J. S. Smagorinsky. General circulation experiments with the primitive equations. *Mon. Weather Rev.*, 91:99–164, 1963.

[147] R. Ştefănescu, A. Sandu, and I. M. Navon. POD/DEIM reduced-order strategies for efficient four dimensional variational data assimilation. *J. Comput. Phys.*, 295:569–595, 2015.

[148] S. Stolz and N.A Adams. An approximate deconvolution procedure for large-eddy simulation. *Phys. Fluids*, 11(7):1699–1701, 1999.

[149] S. Stolz and N.A Adams. Large-eddy simulation of high-reynolds-number supersonic boundary layers using the approximate deconvolution model and a rescaling and recycling technique. *Phys. Fluids*, 15(8):2398–2412, 2003.

[150] S. Stolz, N.A. Adams, and L. Kleiser. An approximate deconvolution model for large-eddy simulation with application to incompressible wall-bounded flows. *Phys. Fluids*, 13(4):997–1015, 2001.

[151] S. Stolz, N.A. Adams, and L. Kleiser. The approximate deconvolution model for large-eddy simulations of compressible flows and its application to shock-turbulent-boundary-layer interaction. *Phys. Fluids*, 13(10):2985–3001, 2001.

[152] Gilbert Strang and George J Fix. *An analysis of the finite element method*, volume 212. Prentice-hall Englewood Cliffs, NJ, 1973.

[153] K. Taira, S. L. Brunton, S. Dawson, C. W. Rowley, T. Colonius, B. J. McKeon, O. T. Schmidt, S. Gordeyev, V. Theofilis, and L. S. Ukeiley. Modal analysis of fluid flows: An overview. *Available as arXiv preprint at* `http://arxiv.org/abs/1702.01453`, 2017.

[154] R. Temam. *Navier-Stokes equations: Theory and numerical analysis*, volume 2. American Mathematical Society, 2001.

[155] F. Thomasset. Implementation of FEM for Navier-Stokes equations, 1981.

[156] V. Thomée. *Galerkin finite element methods for parabolic problems*. Springer Verlag, 2006.

[157] S. Ullmann and J. Lang. A POD-Galerkin reduced model with updated coefficients for Smagorinsky LES. In J. C. F. Pereira and A. Sequeira, editors, *V European Conference on Computational Fluid Dynamics, ECCOMAS CFD 2010*, Lisbon, Portugal, June 2010.

[158] C. R. Vogel. *Computational Methods for Inverse Problems*, volume 23. Society for Industrial and Applied Mathematics (SIAM), 2002.

[159] S. Volkwein. Model reduction using proper orthogonal decomposition. *Lecture Notes, Faculty of Mathematics and Statistics, University of Konstanz*, 2011. `http://www.math.uni-konstanz.de/numerik/personen/volkwein/teaching/POD-Vorlesung.pdf`.

[160] Y. Wang, I. M. Navon, X. Wang, and Y. Cheng. 2D Burgers equation with large Reynolds number using POD/DEIM and calibration. *Int. J. Num. Meth. Fluids*, 82(12):909–931, 2016.

[161] Z. Wang, I. Akhtar, J. Borggaard, and T. Iliescu. Two-level discretizations of nonlinear closure models for proper orthogonal decomposition. *J. Comput. Phys.*, 230:126–146, 2011.

[162] Z. Wang, I. Akhtar, J. Borggaard, and T. Iliescu. Proper orthogonal decomposition closure models for turbulent flows: A numerical comparison. *Comput. Meth. Appl. Mech. Eng.*, 237-240:10–26, 2012.

[163] J. Weller, E. Lombardi, and A. Iollo. Robust model identification of actuated vortex wakes. *Phys. D*, 238(4):416–427, 2009.

[164] D. Wells. *Stabilization of POD-ROMs*. PhD thesis, Virginia Tech, `http://vtechworks.lib.vt.edu/bitstream/handle/10919/52960/Wells_DR_D_2015.pdf?sequence=1&isAllowed=y`, 2015.

[165] D. Wells, Z. Wang, X. Xie, and T. Iliescu. An evolve-then-filter regularized reduced order model for convection-dominated flows. *Int. J. Num. Meth. Fluids*, 2017. To appear. Available as arXiv preprint at `http://arxiv.org/abs/1506.07555`.

[166] D. Xiao, F. Fang, A. G. Buchan, C. C. Pain, I. M. Navon, J. Du, and G. Hu. Non-linear model reduction for the Navier–Stokes equations using residual DEIM method. *J. Comput. Phys.*, 263:1–18, 2014.

[167] X. Xie. Approximate deconvolution reduced order modeling. Master's thesis, Virginia Tech, http://theses.lib.vt.edu/theses/available/etd-11232015-161835/unrestricted/Xie_X_T_2015.pdf, 2015.

[168] X. Xie, D. Wells, Z. Wang, and T. Iliescu. Approximate deconvolution reduced order modeling. *Comput. Methods Appl. Mech. Engrg.*, 313:512–534, 2017.

[169] Olgierd Cecil Zienkiewicz, Robert Leroy Taylor, and Robert Lee Taylor. *The finite element method*, volume 3. McGraw-hill London, 1977.