Università di Trento

DIPARTIMENTO DI MATEMATICA
Doctoral Programme in Mathematics - Cycle XXXVII

Thesis for the Degree of Philosophiae Doctor

Regularization Techniques for Reduced Order Models in Convection-Dominated Flows

Candidate:
Anna Sanfilippo

Supervisor:
Prof. Francesco Ballarin

Co-Supervisor:
Prof. Ana Alonso Rodríguez

Academic Year 2024-2025

Alle Morgane nella mia vita

"[...] un intrepido esercito di Morgane, fatto di donne - ma anche di uomini, persone in transizione e queer -, che in quelle storie di libertà e coraggio si sono viste e riconosciute."
Morgana. Il corpo della madre
Michela Murgia, Chiara Tagliaferri

Abstract

This thesis aims to provide an overview of our novel stabilization strategy for Reduced Order Models (ROMs) applied to convection-dominated flows in an under-resolved regime: the Approximate Deconvolution Leray ROM (ADL-ROM).

In the last decades, Galerkin Reduced Order Models (G-ROMs) have gained significant attention within the scientific community due to the ability to drastically reduce the computational cost of numerical simulations, an essential goal for real-world applications. Despite their success, the use of ROMs in convection-dominated under-resolved regime has been hindered by numerical instability. Specifically, while the total energy remains bounded, the standard G-ROM leads to inaccurate and unreliable solutions. This issue highlights the necessity of ROM stabilization techniques. One widely adopted approach to address this challenge is the Regularized ROMs (Reg-ROMs), which apply explicit spatial filtering to stabilize G-ROMs. A well-known example is the Leray ROM (L-ROM). However, existing Reg-ROMs often introduce excessive numerical diffusion in their attempt to mitigate G-ROM instabilities, potentially compromising solution accuracy.

To overcome these limitations, the new ADL-ROM integrates Approximate Deconvolution (AD) operators into the classical Leray ROM framework, enhancing accuracy without sacrificing numerical stability. Specifically, we propose two ADL-ROM strategies based on Lavrentiev and van Cittert deconvolution methods. Our numerical investigations demonstrate that, for relatively large filter radii, ADL-ROM is significantly more stable than the standard G-ROM, and more accurate than the L-ROM. Additionally, ADL-ROM exhibits reduced sensitivity to model parameters in comparison to L-ROM. Furthermore, we prove *a priori* error bounds for both the AD operator and the ADL-ROM. To our knowledge, these results represent the first numerical analysis of approximate deconvolution in a ROM context. Then, we discuss ongoing research meant to extend the ADL-ROM methodology to parameter spaces. Numerical and theoretical results are illustrated through several computational simulations of convection-dominated flows.

Acknowledgments

First, I want to thank my supervisor, Prof. Francesco Ballarin, and my internal supervisor, Prof. Ana Alonso Rodríguez, for always supporting me during this long journey, not only at an academic and professional level but also at a personal level.

I thank the referees of this thesis, Prof. Gianluigi Rozza and Prof. Michele Girfoglio, for reviewing this work.

I also have to thank Prof. Traian Iliescu and Mr. Ian Moore for being very helpful and professional during our collaboration.

I acknowledge the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie Actions, grant agreement 872442 (ARIA).

I thank my parents for everything they taught me and for allowing me to move to Trento several years ago.

I also want to thank all the people who have passed through and, in some cases, remained in my life over these years, making it richer and more enjoyable. In particular, I owe a huge thank you to all my friends in Rome who still put up with me at the third thesis.

I thank the two people that during this PhD I have chosen, and would choose a thousand more times, knowingly as friends, Nicola and Sara. Thanks for understanding that I don't like lightheartedness deep down, for having accepted and shared it, but for always making me laugh despite this.

I can't miss to thank Valentino. Not only was he a precious colleague who helped me, advised me and solved hundreds of problems and work-related doubts, but he was also an extraordinary (and initially unexpected) friend with whom I could take a million breaks and talk about everything.

Last but certainly not least, I want to thank Carlo. I don't know how to express in words, and those who know me know that this never happens to me, how grateful I am to him. During my entire PhD, no matter how the day went, I knew that in the evening, I would open the door of my house, and everything would be fine because he would be there. Maybe I can sum up my gratitude with thanks for having consistently been my new concept of home.

Contents

Introduction **1**

1 Preliminaries **9**
1.1 Navier-Stokes Equations 9
1.1.1 Boundary and Initial Conditions 10
1.1.2 Nonlinearity and Dimensional Parameters 11
1.1.3 Applications and Challenges 11
1.2 Finite Element Preliminaries 11
1.2.1 Notation and Function Spaces 12
1.2.2 Weak Formulation of the Navier-Stokes Equations . . . 12
1.2.3 Time Discretization: Backward Differentiation Formula 12
1.2.4 Spatial Discretization: Taylor-Hood Finite Elements . . 14
1.3 Reduced Order Model Preliminaries 16
1.3.1 Proper Orthogonal Decomposition 17
1.3.2 ROM Projection 18
1.3.3 Galerkin ROM 20

2 Leray Model **25**
2.1 Differential Filters 26
2.1.1 Explicit ROM Spatial Filtering 26
2.1.2 ROM-DF Bounds 31
2.2 Leray ROM 32
2.2.1 L-ROM Time Discretization 33
2.2.2 Discussion and Limitations 34

3 Approximate Deconvolution Leray Model **37**
3.1 Approximate Deconvolution Operators 38
3.1.1 The Van Cittert AD 39
3.1.2 The Tikhonov AD 40
3.1.3 The Lavrentiev AD 42

3.2 Approximate Deconvolution Leray Reduced Order Model (ADL-ROM) 44
3.2.1 ADL-ROM Time Discretization 45
3.2.2 Definition of AD Convective Term 45
3.2.3 ADL-ROM Implementation 46
3.3 Numerical Results 46
3.3.1 RBniCSx 47
3.3.2 Flow Past a Cylinder 49
3.3.3 Backward Facing Step 56

4 Numerical Analysis **69**
4.1 Numerical Analysis of AD Operators 70
4.2 Numerical Analysis of ADL-ROM 73
4.3 Numerical Results 86
4.3.1 ADL-ROM Solutions 90
4.3.2 AD Rates of Convergence 93
4.3.3 ADL-ROM Rates of Convergence 96

5 Parametric Approximate Deconvolution Leray Model **101**
5.1 Model Reduction: With Respect to Time and Viscosity 102
5.2 Parametric ADL-ROM 104
5.3 Numerical Results 104

Conclusions **119**

List of Figures **125**

List of Tables **129**

Bibliography **131**

Introduction

Reduced Order Models (ROMs) have emerged as powerful surrogate models capable of drastically reducing the dimensional complexity of Full Order Models (FOMs), which are obtained through classical numerical discretizations such as the Finite Element Method (FEM). By leveraging data-driven approaches, ROMs provide computationally efficient alternatives that retain the essential dynamics of high-fidelity simulations while significantly lowering their computational cost. Among the various ROM techniques, Galerkin Reduced Order Models (G-ROMs) have gained significant attention due to their systematic construction within a Galerkin framework.

The foundation of G-ROMs lies in the data-driven construction of a low-dimensional basis. Specifically, a set of optimal basis functions $\{\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_R\}$ is obtained from training data (i.e., problem and parameters) that encapsulates the dominant structures of the underlying physical system. The leading r basis functions, where $r \ll R$, are then employed in a Galerkin formulation of the governing Partial Differential Equations (PDEs), yielding a nonlinear r-dimensional system of Ordinary Differential Equations (ODEs). This dimensional reduction results in computational models that are orders of magnitude smaller than their full order counterparts.

Galerkin ROMs have demonstrated significant success in the numerical simulation of diffusion-dominated, or laminar, flows characterized by low Reynolds numbers in the context of the Navier-Stokes equations [10, 18, 44, 71, 77]. The ability of G-ROMs to effectively capture and reduce the computational complexity of such flows has made them a widely used tool in model reduction techniques. However, standard G-ROMs exhibit substantial limitations when applied to convection-dominated flows, such as turbulent flows, which are prevalent at high Reynolds numbers. These challenges become particularly pronounced in the under-resolved regime, where the number of degrees of freedom in the reduced order model is insufficient to accurately capture the full complexity of the flow dynamics. This limitation poses a significant drawback since the vast majority of real-world flows encountered in engineering, e.g., aerospace and automotive industry, and geophysical, e.g., weather prediction and climate modeling, applications are convection-dominated. Consequently, the inability of standard G-ROMs to handle such convection-dominated scenarios presents a major obstacle to their broader applicability.

The main reason behind the poor performance of standard G-ROMs in convection-dominated flows is the inherent complexity of such flows, which exhibit a wide range of spatial scales. To faithfully represent all these spatial scales, a quite large number of basis functions would be required, leading to a high-dimensional reduced order model. However, because G-ROMs are designed to be computationally efficient and relatively low-dimensional, practitioners typically truncate the ROM basis to a smaller set of modes. While this truncation reduces computational cost, it also introduces significant inaccuracies in the numerical results. The resulting under-resolved G-ROM often exhibits numerical oscillations, which degrade solution accuracy and stability. The origin of these inaccuracies lies in the omission of higher index basis functions $\{\boldsymbol{\varphi}_{r+1}, \ldots, \boldsymbol{\varphi}_R\}$, which are essential for capturing the fine-scale dynamics of the flow. These higher index modes play a crucial role in the dissipation of energy within the system, similar to the classical Fourier representation of turbulent flows [21, 84]. When these dissipative effects are not adequately accounted for in the reduced order model, the energy that should be dissipated is improperly retained within the lower-dimensional space, leading to the emergence of unphysical oscillations and numerical instabilities.

To mitigate these stability issues and improve the accuracy of under-resolved G-ROMs in convection-dominated flows, there are essentially two strategies:

1. *ROM Closures:* These methods introduce additional terms, often referred to as closure models, that aim to reconstruct the effects of the discarded basis functions. By incorporating corrections that approximate the impact of the omitted modes $\{\boldsymbol{\varphi}_{r+1}, \ldots, \boldsymbol{\varphi}_R\}$ on the resolved dynamics, ROM closures enhance the overall fidelity of the model.

2. *ROM Stabilizations:* Unlike closure models, ROM stabilization techniques modify the standard G-ROM formulation itself by introducing additional stabilization terms. These modifications improve numerical stability and mitigate the occurrence of unphysical oscillations.

In this thesis, we focus only on ROM stabilizations since our original idea was to construct a method that could be easily applied to several problems without additional considerations and estimates. Several ROM stabilization techniques have been explored in the literature [2, 1, 7, 12, 14, 19, 26, 39, 46, 51, 74, 75, 76, 93, 100]. Among these approaches, Regularized ROMs

(Reg-ROMs) have gained considerable attention due to their effectiveness in improving stability through the application of spatial filtering mechanisms.

The fundamental principle underlying Reg-ROMs is the use of a ROM filter to smooth out various terms in the underlying Navier-Stokes equations. The objective of this filtering process is to remove high-frequency numerical artifacts while preserving the key physical features of the flow. There are two primary types of ROM filters currently employed in the construction of Reg-ROMs:

1. *ROM Projection Filter:* In this approach, the filtering process takes place entirely within the ROM space.
2. *ROM Differential Filter:* A technique where filtering is applied within the physical space rather than directly in the ROM space. Higher-order differential filters (DFs) have also been proposed to improve accuracy and stability [40].

Both the ROM projection filter and the ROM differential filter have been effectively utilized in the construction of Reg-ROMs, demonstrating improved stability and accuracy in reduced order modeling of convection-dominated flows. By leveraging these filtering techniques, Reg-ROMs provide a promising avenue for enhancing the performance of G-ROMs in complex flow scenarios, ensuring better agreement with high-fidelity simulations while maintaining computational efficiency.

A particularly well-studied Reg-ROM is the Leray ROM (L-ROM) [33, 37, 52, 85, 96], which is a ROM stabilization for under-resolved convection-dominated flows. Leray model, which draws inspiration from Jean Leray's pioneering work in mathematical fluid dynamics [63], is base on the following core idea: the introduction of a filter that smooths the velocity field. The Leray model was extended to reduced order modeling of the NSE in [96] (see [85] for earlier work on the Kuramoto-Sivashinsky equations). Specifically, the nonlinear convective term $\boldsymbol{u} \cdot \nabla \boldsymbol{u}$ in the Navier-Stokes equations (NSE) is replaced with $\overline{\boldsymbol{u}} \cdot \nabla \boldsymbol{u}$, where $\overline{\boldsymbol{u}}$ denotes the filtered ROM velocity. This filtering operation mitigates the unphysical oscillations that often arise in the under-resolved G-ROM simulations by introducing a controlled amount of numerical dissipation. The Leray model has been successfully applied to various convection-dominated flows, e.g., further NSE applications [33, 37, 38, 34], the stochastic NSE [40, 42], and the quasi-geostrophic equations [35, 36].

Despite the significant improvement over the standard Galerkin ROM, the L-ROM is not without drawbacks. One notable issue is its tendency to be overdiffusive when the ROM filter radius, δ, is excessively large, leading to the suppression of physically relevant flow structures, e.g., oscillations and vortices. Additionally, the accuracy of L-ROM is highly sensitive to the choice of δ: small perturbations in the optimal filter radius adversely affecting the model's predictive capabilities. These limitations motivate the development of improved ROM stabilization techniques that can achieve a balance between numerical stability and accuracy.

To this end, we conceived, constructed and now present the new Approximate Deconvolution Leray ROM (ADL-ROM), which is a significant improvement of the L-ROM. ADL-ROM builds upon the Leray ROM framework by incorporating Approximate Deconvolution (AD) operator techniques. Approximate deconvolution, a well-established methodology in image processing and inverse problems, enhances the accuracy of filtered quantities by iteratively reconstructing higher-order corrections. To construct the ADL-ROM model, we introduce three ROM AD operators: the ROM Lavrentiev, the ROM Tikhonov, and the ROM Van Cittert operators. Another novel contribution of the ADL-ROM is that two of these AD operators were never used at ROM level: the Tikhonov and the van Cittert ones.

Thus, the ADL-ROM innovation is that the filtered velocity field $\overline{\boldsymbol{u}}$ is replaced with its approximate deconvolution counterpart $D(\overline{\boldsymbol{u}})$, leading to a refined nonlinear term $D(\overline{\boldsymbol{u}}) \cdot \nabla \boldsymbol{u}$. By leveraging the stability of filtering while recovering crucial fine-scale structures, ADL-ROM effectively mitigates the overdiffusive behavior of L-ROM while preserving its stabilizing properties.

The introduction of ADL-ROM marks a significant advancement in ROM stabilization methodologies, as it represents the first application of approximate deconvolution in the context of Reg-ROMs. Moreover, our work contributes to the rigorous numerical analysis of ROM stabilizations by deriving *a priori* error bounds for both the novel ROM van Cittert approximate deconvolution operator and the ADL-ROM itself. These theoretical results provide fundamental insights into the error dependence of ADL-ROM on both ROM and FOM parameters, bridging the gap between heuristic numerical assessments and rigorous mathematical analysis.

Overall, the development of ADL-ROM offers a promising pathway for improving the robustness and accuracy of ROMs in the numerical simulation of convection-dominated flows. By addressing the key limitations of existing Reg-ROMs, ADL-ROM provides a more reliable, theoretically grounded approach to reduced order modeling in fluid dynamics.

This contribution will address all these topics and provide validations and proofs for the aforementioned statements regarding the AD operators and the ADL-ROM. The work presented here is original and builds upon findings and results from a published paper [86], a submitted preprint [68], and ongoing projects. Notably, some of the results in Chapter 3 and Chapter

4 are obtained from a collaboration with Prof. T. Iliescu and his PhD student Mr. I. Moore at Virginia Tech. Regarding the application of ADL-ROM, I conducted all numerical investigations for the Navier-Stokes equations, while Ian Moore focused on the Burgers' equation. His own results are not included in this thesis. A detailed outline of the thesis follows.

- *Part I.* This part focuses on establishing the notation and revisiting fundamental definitions. It does not present any novel findings but rather consolidates existing material in a structured and consistent manner, ensuring uniform notation throughout this thesis. Furthermore, theoretical results are accompanied by citations referring to their original formulations and proofs. This part is structured into two chapters.

  * *Chapter 1.* This chapter begins by introducing the Partial Differential Equations (PDEs) under consideration, i.e., the Navier-Stokes equations. It then explores key aspects of the Finite Element Method (FEM). Following this, an overview of Reduced Order Methods (ROMs) is provided, with particular attention to the Galerkin ROM formulation. The Proper Orthogonal Decomposition (POD) technique is detailed as a method for constructing the reduced space and performing ROM projection. Additionally, essential lemmas and assumptions required for the development and numerical analysis of the central topic of this thesis, the Approximate Deconvolution Leray ROM (ADL-ROM), are presented.
  * *Chapter 2.* This chapter is divided into two main sections. The first section focuses on the ROM Differential Filter (ROM-DF), which, as previously mentioned, is a fundamental tool for ROM stabilization. Theoretical bounds for ROM-DF are provided, along with an analysis of its stability and error characteristics, particularly concerning approximation errors arising from finite element discretization, Proper Orthogonal Decomposition (POD), and filtering operations. ROM-DF plays a crucial role in developing the Approximate Deconvolution Leray ROM (ADL-ROM), introduced in Chapter 3.

    In the second section, the Leray Reduced Order Model (L-ROM) is introduced. As previously stated, this approach is effective in controlling high-frequency errors, making it particularly useful for turbulent flow simulations. While L-ROM enhances accuracy compared to the standard Galerkin ROM (G-ROM), it has a tendency to produce overly diffusive solutions. To address this limitation, L-ROM serves as a foundation for constructing

ADL-ROM, which seeks to preserve the benefits of filtering while reducing excessive diffusion.

- *Part II.* The second part of this work is dedicated to our new Reg-ROM, the ADL-ROM. On one hand, we introduce the techniques and algorithms underlying ADL-ROM and validate them through several numerical results. On the other hand, we establish and prove several theoretical results concerning the ROM AD operators and the ADL-ROM, further verifying them with numerical experiments.

  This section is organized into three chapters. Chapter 3 is based on a published paper co-authored with I. Moore and T. Iliescu from Virginia Tech [86]. Chapter 4 comprehensively details a submitted preprint, also in collaboration with I. Moore and T. Iliescu [68]. Lastly, Chapter 5 discusses part of an ongoing project.

  * *Chapter 3.* This chapter introduces the Approximate Deconvolution Leray ROM (ADL-ROM), our new Regularized ROM (Reg-ROM) designed to enhance the performance of the Leray ROM (L-ROM). ADL-ROM is the first ROM stabilization approach integrating the approximate deconvolution (AD) framework within Reg-ROMs.

    The chapter begins by presenting three distinct ROM AD operators (van Cittert, Tikhonov, and Lavrentiev). It then describes the development of ADL-ROM itself. The final section showcases several numerical experiments comparing ADL-ROM against L-ROM and G-ROM for convection-dominated flows in under-resolved regimes.

  * *Chapter 4.* This chapter provides a rigorous numerical analysis of the van Cittert approximate deconvolution operator and the Approximate Deconvolution Leray ROM (ADL-ROM).

    The structure of the chapter is as follows: first, we establish the first error bounds for the van Cittert AD operator at the ROM level. Then, we present stability and convergence results for ADL-ROM, specifically deriving the first a priori error bounds for ADL-ROM, offering insights into how ADL-ROM errors depend on ROM and FEM parameters. Finally, numerical experiments verify the theoretical results. These findings represent a significant step forward in the numerical analysis of ROM stabilization techniques.

* *Chapter 5.* This chapter investigates an ongoing research project aimed at extending the ADL-ROM framework to a parametric setting. While previous chapters concentrated on time-domain model reduction, this study explores the impact of incorporating parametric dependencies, particularly variations in viscosity.

  Additionally, the chapter introduces the nested-POD (n-POD) method for simultaneous reduction in time and parameter spaces. It then formulates the parametric ADL-ROM and presents preliminary numerical results evaluating its accuracy. As this research is still evolving, further refinements are required to fully assess the potential of the parametric ADL-ROM.

At the end of the thesis, some conclusions and perspectives are collected.

Furthermore, we highlight that all numerical results presented in this thesis have been implemented using the RBniCSx library. For additional details regarding its implementation and usage, refer to Section 3.3.1.

Chapter 1

Preliminaries

In this chapter, we want to present the theoretical foundations needed for the rest of this thesis. In particular, we introduce the Partial Differential Equations (PDEs) we worked mainly on, i.e., the Navier-Stokes equations. Then we describe and analyze the most valuable concepts for the Finite Element Method (FEM). In the end, we recall the theoretical background for the class of methods known as Reduced Order Methods (ROMs) since we are interested in using them in all the following chapters. In particular, we focus on the Galerkin ROM formulation by explaining the use of the Proper Orthogonal Decomposition to construct the reduced space and the ROM projection. A collection of lemmas and assumptions are given in this chapter since they are fundamental to the construction and the numerical analysis of our new Regularized ROM (Reg-ROM), i.e., Approximate Deconvolution Leray ROM (ADL-ROM).

1.1 Navier-Stokes Equations

As a mathematical model, we consider the incompressible Navier-Stokes equations (NSE), which describe the motion of a viscous, incompressible fluid. These equations form the foundation of fluid dynamics and are expressed as follows

$$\frac{\partial \boldsymbol{u}}{\partial t} - \nu \Delta \boldsymbol{u} + \big(\boldsymbol{u} \cdot \nabla\big)\boldsymbol{u} + \nabla p = \boldsymbol{f}, \quad \text{in } \Omega \times (0, T], \tag{1.1.1}$$

$$\nabla \cdot \boldsymbol{u} = 0, \quad \text{in } \Omega \times (0, T]. \tag{1.1.2}$$

Here, the various components are defined as follows

- $\Omega \subset \mathbb{R}^d$ is the spatial domain of interest, where $d = 2$ or 3 represents the spatial dimension (2D or 3D flow);
- T denotes the final time of the simulation, such that the equations are solved for $t \in (0, T]$;
- $\boldsymbol{u} = [u_1, \ldots, u_d]^\top : \Omega \times [0, T] \to \mathbb{R}^d$ is the velocity vector field of the fluid. The components u_i represent the velocity in the i-th spatial direction;
- $p : \Omega \times [0, T] \to \mathbb{R}$ is the pressure scalar field associated with the fluid;
- ν is the kinematic viscosity, which quantifies the fluid resistance to shear and is inversely proportional to the Reynolds number ($\nu \propto \mathrm{Re}^{-1}$). Naturally, it is a positive quantity;
- $\boldsymbol{f} : \Omega \times [0, T] \to \mathbb{R}^d$ represents the external forcing term (e.g., gravity, applied forces, etc.).

The NSE consist of two coupled equations, which represent

1. *Momentum Equation* (1.1.1): This equation governs the conservation of linear momentum in the fluid, balancing inertial forces, viscous diffusion, convective transport, pressure gradients, and external forces. The term $\frac{\partial \boldsymbol{u}}{\partial t}$ represents the temporal acceleration, $\nu \Delta \boldsymbol{u}$ corresponds to viscous diffusion, $(\boldsymbol{u} \cdot \nabla)\boldsymbol{u}$ models nonlinear convective acceleration, and ∇p accounts for pressure gradients.
2. *Continuity Equation* (1.1.2): This incompressibility condition enforces the conservation of mass in the fluid, ensuring that the fluid density remains constant and the velocity field is divergence-free.

1.1.1 Boundary and Initial Conditions

The NSE are equipped with appropriate boundary and initial conditions to fully specify the problem

- **Initial Conditions**: The initial velocity field is prescribed as

$$\boldsymbol{u}(\boldsymbol{x}, 0) = \boldsymbol{u}_0(\boldsymbol{x}), \quad \text{for all } \boldsymbol{x} \in \Omega, \tag{1.1.3}$$

where $\boldsymbol{u}_0$ is a divergence-free function satisfying $\nabla \cdot \boldsymbol{u}_0 = 0$.

- **Boundary Conditions**: For simplicity, we often consider homogeneous Dirichlet boundary conditions, which impose a no-slip condition at the domain boundaries, i.e.,

$$\boldsymbol{u}(\boldsymbol{x}, t) = 0, \quad \text{for all } \boldsymbol{x} \in \partial\Omega \text{ and } t \in [0, T]. \tag{1.1.4}$$

  Alternatively, other types of boundary conditions, such as free-slip, inflow/outflow, or periodic conditions, may be used depending on the application.

1.1.2 Nonlinearity and Dimensional Parameters

The NSE exhibit significant nonlinearity due to the convective term $(\boldsymbol{u} \cdot \nabla)\boldsymbol{u}$, making them challenging to solve analytically and numerically. The relative importance of inertia versus viscous forces is quantified by the Reynolds number

$$\mathrm{Re} = \frac{\mathrm{UL}}{\nu}, \tag{1.1.5}$$

where U is a characteristic velocity and L is a characteristic length scale of the flow. Large Reynolds numbers correspond to turbulence-dominated regimes, while small Reynolds numbers represent laminar flow.

1.1.3 Applications and Challenges

The Navier-Stokes equations are used in a wide range of applications, including aerodynamics, weather prediction, oceanography, and industrial flows. However, their inherent nonlinearity and coupling between velocity and pressure pose significant computational challenges, particularly for high Reynolds number flows, where turbulence modeling becomes essential.

1.2 Finite Element Preliminaries

In this section, we present the notation and fundamental numerical analysis concepts for the finite element (FE) discretization of the incompressible Navier-Stokes equations (NSE). These preliminaries serve as the foundation for the development and analysis of the FE methods used in subsequent sections.

1.2.1 Notation and Function Spaces

Throughout this section, $\|\cdot\|$ and $(\cdot,\cdot)$ denote the $L^2(\Omega)$ norm and inner product, respectively. We begin by defining the function spaces used for the weak formulation of the NSE. Let

$$\begin{aligned} Q = L_0^2(\Omega) &\doteq \left\{ q \in L^2(\Omega) \mid \int_\Omega q \; d\boldsymbol{x} = 0 \right\}, \\ \boldsymbol{X} = H_0^1(\Omega;\ \mathbb{R}^d) &\doteq \left\{ \boldsymbol{u} \in H^1(\Omega;\ \mathbb{R}^d) \mid \boldsymbol{u}|_{\partial\Omega} = \boldsymbol{0} \right\}, \end{aligned}$$

where $L_0^2(\Omega)$ is the space of square-integrable functions with zero mean, and $H_0^1(\Omega;\ \mathbb{R}^d)$ is the Sobolev space of vector-valued functions that are square-integrable, with square-integrable first derivatives and vanishing trace on $\partial\Omega$.

1.2.2 Weak Formulation of the Navier-Stokes Equations

The weak formulation of (1.1.1)–(1.1.2) is as follows: Find $\boldsymbol{u} \in \boldsymbol{X}$ and $p \in Q$ such that

$$\begin{cases} \displaystyle\int_\Omega \frac{\partial \boldsymbol{u}}{\partial t} \cdot \boldsymbol{v} \; d\boldsymbol{x} + \nu(\nabla \boldsymbol{u}, \nabla \boldsymbol{v}) + b(\boldsymbol{u}, \boldsymbol{u}, \boldsymbol{v}) - (\nabla \cdot \boldsymbol{v}, p) = (\boldsymbol{f}, \boldsymbol{v}), & \forall \boldsymbol{v} \in \boldsymbol{X}, \\ (\nabla \cdot \boldsymbol{u}, q) = 0, & \forall q \in Q, \end{cases} \tag{1.2.1}$$

where the trilinear form $b : \boldsymbol{X} \times \boldsymbol{X} \times \boldsymbol{X} \to \mathbb{R}$ is defined by

$$b(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}) = ((\boldsymbol{u} \cdot \nabla)\boldsymbol{v}, \boldsymbol{w}). \tag{1.2.2}$$

To ensure the well-posedness of the weak formulation, we make the following regularity assumptions (see Definition 29, Proposition 15, and Remark 10 in [60]):

Assumption 1.2.1. *In* (1.1.1)–(1.1.2), *we assume* $\boldsymbol{f} \in L^2(0,T; L^2(\Omega;\ \mathbb{R}^d))$, $\boldsymbol{u}_0 \in \boldsymbol{X}$, $\boldsymbol{u} \in L^2(0,T;\boldsymbol{X}) \cap L^\infty(0,T;L^2(\Omega;\ \mathbb{R}^d))$, $\nabla \boldsymbol{u} \in L^\infty(0,T;L^2(\Omega;\ \mathbb{R}^{d\times d}))$, $\boldsymbol{u}_t \in L^2(0,T;\boldsymbol{X}^*)$, $p \in L^\infty(0,T;Q)$.

1.2.3 Time Discretization: Backward Differentiation Formula

We next proceed to a time discretization by means of the backward differentiation formula of order 2 (BDF2)[1]. Let $k = 1, \ldots, K-1$ denote the index of

[1] Note that other time discretizations are possible, e.g., Crank-Nicolson.

the current time step t_k. For simplicity, we will assume that $t_{k+1} - t_k = \Delta t$, i.e., equispaced timesteps, and $K\Delta t = T$.

We introduce now two notations with subscript $_k$: the notation $\boldsymbol{u}_k$ is reserved for the evaluation of the (strong) solution to (1.1.1)-(1.1.2) at time t_k, i.e., $\boldsymbol{u}_k = \boldsymbol{u}(t_k)$, while $\boldsymbol{u}_k^{\Delta t}$ is employed to denote the unknown resulting after time discretization. Similar notations are adopted for the strong pressure $p_k = p(t_k)$ and its time-discrete approximation $p_k^{\Delta t}$.

BDF2 Time Discretization The BDF2 scheme is an implicit time-stepping method that uses a weighted combination of the current, previous, and second-to-last time steps to approximate the time derivative. This scheme is known for its second-order accuracy and stability properties, particularly in the context of stiff problems.

After semi-discretization with the BDF2 scheme, (1.2.1) becomes

$$\begin{cases} \displaystyle\int_\Omega \frac{3\boldsymbol{u}_{k+1}^{\Delta t} - 4\boldsymbol{u}_k^{\Delta t} + \boldsymbol{u}_{k-1}^{\Delta t}}{2\Delta t} \cdot \boldsymbol{v}\, dx + \nu(\nabla \boldsymbol{u}_{k+1}^{\Delta t}, \nabla \boldsymbol{v}) + b^*(\boldsymbol{u}_{k+1}^{\Delta t}, \boldsymbol{u}_{k+1}^{\Delta t}, \boldsymbol{v}) & \\ \qquad\qquad -(\nabla \cdot \boldsymbol{v}, p_{k+1}^{\Delta t}) = (\boldsymbol{f}_{k+1}, \boldsymbol{v}), & \forall \boldsymbol{v} \in \boldsymbol{X}, \\ (\nabla \cdot \boldsymbol{u}_{k+1}^{\Delta t}, q) = 0, & \forall q \in Q. \end{cases} \tag{1.2.3}$$

Here, the first equation represents the momentum equation, and the second ensures incompressibility.

Skew-Symmetric Trilinear Form Following, e.g., [60, 92], the skew-symmetric trilinear form b^* which appears in (1.2.3) is defined as follows:

$b^* : \boldsymbol{X} \times \boldsymbol{X} \times \boldsymbol{X} \to \mathbb{R}$, such that,

$$b^*(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}) = \frac{1}{2}\left[((\boldsymbol{u} \cdot \nabla)\boldsymbol{v}, \boldsymbol{w}) - ((\boldsymbol{u} \cdot \nabla)\boldsymbol{w}, \boldsymbol{v})\right]. \tag{1.2.4}$$

This choice of trilinear form is motivated by stability considerations. As explained in [60, Section 7.1], we use the skew-symmetric trilinear form b^* in (1.2.3) instead of the trilinear form that we used in (1.2.1) to ensure the stability of the velocity field.

We also note that the trilinear forms b and b^* are equivalent in the continuous case (i.e., $b(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}) = b^*(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}), \forall \boldsymbol{u} \in \boldsymbol{X}, \boldsymbol{v} \in \boldsymbol{X}, \boldsymbol{w} \in \boldsymbol{X}$, see [60, Lemma 18]), but not in the discrete case. Furthermore, we recall the following result from [60, Lemmas 13 and 18]

Lemma 1.2.1. *For all* $\boldsymbol{u}, \boldsymbol{w}, \boldsymbol{v} \in \boldsymbol{X}$*, the skew-symmetric trilinear form* $b^*(\cdot,\cdot,\cdot)$ *satisfies*

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

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

and a sharper bound

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

These properties ensure the stability and boundedness of the nonlinear convective term in the Navier-Stokes equations.

Discrete Gronwall Lemma For later use, we also recall the following discrete version of the Gronwall lemma, as stated in [59, Lemma 2.18]:

Lemma 1.2.2 (Discrete Gronwall Lemma)**.** *Let* $\Delta t > 0$*,* $E > 0$*. Furthermore, let* $\{a_k\}_{k=0}^K$ *,* $\{b_k\}_{k=0}^K$ *,* $\{c_k\}_{k=0}^K$ *,* $\{d_k\}_{k=0}^K$ *be nonnegative sequences such that*

$$a_K + \Delta t \sum_{k=0}^{K} b_k \leq \Delta t \sum_{k=0}^{K} d_k a_k + \Delta t \sum_{k=0}^{K} c_k + E \tag{1.2.5}$$

and

$$d_k < \frac{1}{\Delta t} \quad \textit{for } k = 0, \ldots, K. \tag{1.2.6}$$

Then,

$$a_K + \Delta t \sum_{k=0}^{K} b_k \leq \exp\left(\Delta t \sum_{k=0}^{K} \frac{d_k}{1 - \Delta t d_k} \right) \left(\Delta t \sum_{k=0}^{K} c_k + E \right). \tag{1.2.7}$$

This lemma is instrumental in establishing stability estimates for time-discrete systems, particularly in the context of numerical schemes for partial differential equations. It provides a mechanism to bound solution growth over time in terms of initial data and forcing terms, a critical step in the convergence analysis of numerical methods.

1.2.4 Spatial Discretization: Taylor-Hood Finite Elements

The fully discrete scheme is finally obtained by applying a further discretization in space by means of the finite element method. In particular we use the Taylor-Hood finite element spaces.

Taylor-Hood Finite Element Spaces The Taylor-Hood finite element spaces are widely used in computational fluid dynamics for the numerical solution of incompressible flows due to their well-established stability and convergence properties. The spaces are defined as follows:

- $\boldsymbol{X}^h$: The space of vector-valued piecewise polynomials of degree $m+1$ (e.g., $\mathbb{P}^{m+1}$ elements for velocity), with m a positive integer, which are continuous across element boundaries;

- Q^h: The space of scalar piecewise polynomials of degree m (e.g., $\mathbb{P}^m$ elements for pressure), also continuous across element boundaries.

The higher-order polynomial degree for the velocity field compared to the pressure field ensures the stability of the mixed formulation while maintaining computational efficiency. The parameter h represents the characteristic mesh size, with smaller h leading to finer spatial resolution.

These spaces are specifically chosen to satisfy the Ladyzhenskaya-Babuška-Brezzi (LBB) condition, ensuring the stability of the velocity-pressure pair in the mixed finite element formulation.

The Taylor-Hood finite element spaces yield the following finite element discretization. Given $\boldsymbol{u}_k^h \in \boldsymbol{X}^h$, find $\boldsymbol{u}_{k+1}^h \in \boldsymbol{X}^h$ and $p_{k+1}^h \in Q^h$, such that

$$\begin{cases} \displaystyle\int_\Omega \frac{3\boldsymbol{u}_{k+1}^h - 4\boldsymbol{u}_k^h + \boldsymbol{u}_{k-1}^h}{2\Delta t} \cdot \boldsymbol{v}^h; dx + \nu(\nabla \boldsymbol{u}_{k+1}^h, \nabla \boldsymbol{v}^h) + b^*(\boldsymbol{u}_{k+1}^h, \boldsymbol{u}_{k+1}^h, \boldsymbol{v}^h) & \\ \qquad\qquad -(\nabla \cdot \boldsymbol{v}^h, p_{k+1}^h) = (\boldsymbol{f}_{k+1}^h, \boldsymbol{v}^h), & \forall \boldsymbol{v}^h \in X^h, \\ (\nabla \cdot \boldsymbol{u}_{k+1}^h, q^h) = 0, & \forall q^h \in Q^h. \end{cases} \tag{1.2.8}$$

Here, the first equation corresponds to the momentum balance, discretized in both time and space, and the second one enforces the incompressibility condition for the velocity field at each time step.

To simplify the notation, the fully discrete solutions are denoted as $\boldsymbol{u}_k^h$ and p_k^h. Strictly speaking, these should be written as $\boldsymbol{u}_k^{\Delta t,h}$ and $p_k^{\Delta t,h}$ to indicate their dependence on both the time step Δt and the mesh size h. However, since the semi-discrete scheme in (1.2.3) serves only as an intermediate step, we opt for the simpler notation $\boldsymbol{u}_k^h$ and p_k^h.

Asymptotic Error Bound We assume the following asymptotic error bound for the finite element approximation, similar to the assumptions in [99, Assumption 2.2]:

Assumption 1.2.2. *The finite element approximation* $\boldsymbol{u}^h$ *satisfies*

$$||\boldsymbol{u}_K - \boldsymbol{u}_K^h||^2 + \Delta t \sum_{k=1}^{K} ||\nabla(\boldsymbol{u}_k - \boldsymbol{u}_k^h)||^2 \leq C\left(h^{2m+2} + \Delta t^4\right). \tag{1.2.9}$$

Moreover, for the pressure field, we assume the standard approximation property

$$\inf_{q^h \in Q^h} ||p - q^h|| \leq C h^{m+1}, \quad \forall p \in Q. \tag{1.2.10}$$

From now on, C is a generic constant that can depend on the problem data, but not on the finite element parameters (e.g., $h, m, \Delta t$) or, in the context of the next section and chapters, the ROM parameters (e.g., r, λ_j). The bound in (1.2.9) reflects the combined spatial and temporal discretization errors. The term h^{2m+2} corresponds to the spatial error, while Δt^4 accounts for the second-order accuracy of the BDF2 time-stepping scheme.

Discussion of Stability and Convergence The stability of the fully discrete scheme follows from the properties of the Taylor-Hood elements and the choice of the skew-symmetric trilinear form b^*. The latter ensures that the nonlinear convective term does not introduce numerical instabilities. Convergence is guaranteed by the consistency of the finite element spaces and the boundedness of the discrete operators.

The Taylor-Hood element pair is particularly advantageous for incompressible flows because it

- Satisfies the inf-sup condition, ensuring pressure stability;
- Provides higher accuracy for velocity due to the use of $\mathbb{P}^{m+1}$ elements;
- Is computationally efficient for a range of practical problems.

In the context of the Navier-Stokes equations, these properties make the fully discrete scheme robust for various flow regimes, including low Reynolds number flows and mildly turbulent cases. Refer to [60] for the $\mathbb{P}^{m+1}/\mathbb{P}^m$ Taylor-Hood pair for the FE space discretization, and a second-order time stepping scheme; see also error bound (2.7) in [66].

1.3 Reduced Order Model Preliminaries

In this section, we briefly describe the construction of the standard G-ROM.

1.3.1 Proper Orthogonal Decomposition

We will briefly discuss proper orthogonal decomposition, referred to as POD, which is one of the primary techniques used in Galerkin reduced order modeling to obtain a low-dimensional basis. A more thorough presentation of the topic is given in, e.g., [58]. For a specific time t_k, we call $\boldsymbol{u}(\boldsymbol{x}, t_k)$ a velocity snapshot. This snapshot contains data for all $\boldsymbol{x} \in \Omega$ at that specific time. If the exact solution $\boldsymbol{u}$ were available, then $\boldsymbol{u}_k(\boldsymbol{x})$ could be used as snapshot at time t_k; in practice, since the exact solution is not available, $\boldsymbol{u}_k^h(\boldsymbol{x})$ is used instead.

A further step could be to compute the centered snapshots, i.e. $\{\boldsymbol{u}_0^h - \boldsymbol{U}^h, \boldsymbol{u}_1^h - \boldsymbol{U}^h, \ldots, \boldsymbol{u}_K^h - \boldsymbol{U}^h\}$, where the centering trajectory $\boldsymbol{U}^h$ of the flow is defined as $\boldsymbol{U}^h(\boldsymbol{x}) = \frac{1}{K+1}\sum_{k=0}^{K} \boldsymbol{u}_k^h(\boldsymbol{x})$. This additional stage is needed if the used BCs are not Dirichlet homogeneous BCs.

Remark 1.3.1. To simplify the presentation, we assume that $\boldsymbol{U}^h = \boldsymbol{0}$ and that the NSE (1.1.1)-(1.1.2) are equipped with homogeneous Dirichlet boundary conditions. ♠

Given an ensemble of snapshots $\{\boldsymbol{u}(\boldsymbol{x}, t_k)\}_{k=0}^{K}$, the POD method obtains a low-dimensional basis, $\{\boldsymbol{\varphi}_1, \ldots \boldsymbol{\varphi}_r\}$, that approximates the given data by solving the following minimization problem [58]

$$\min_{\{\boldsymbol{\varphi}_j\}_{j=1}^r} \frac{1}{K+1} \sum_{k=1}^{K} \left\| \boldsymbol{u}(\boldsymbol{x}, t_k) - \sum_{j=1}^{r} \Big(\boldsymbol{u}(\boldsymbol{x}, t_k), \boldsymbol{\varphi}_j(\boldsymbol{x})\Big) \boldsymbol{\varphi}_j(\boldsymbol{x}) \right\|^2, \tag{1.3.1}$$

subject to $(\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_j) = \delta_{ij}$, the Kronecker delta. We introduce the correlation matrix $\mathcal{C} \in \mathbb{R}^{(K+1)\times(K+1)}$, where $\mathcal{C}_{k\ell} \doteq \frac{1}{K+1}\big(\boldsymbol{u}(\boldsymbol{x}, t_k), \boldsymbol{u}(\boldsymbol{x}, t_\ell)\big)$ for $k, \ell = 0, \ldots, K$. The solution to (1.3.1) may be found by solving the eigenvalue problem $\mathcal{C}\boldsymbol{v}_j = \lambda_j \boldsymbol{v}_j$. These eigenvalues λ_j are positive and sorted in descending order $\lambda_1 \geq \cdots \geq \lambda^r \geq \cdots \geq \lambda_R \geq 0$, where $R = \mathrm{rank}(\mathcal{C})$. Then, a set of basis functions of dimension $r \leq R$ satisfying (1.3.1) can be shown to be given by $\boldsymbol{\varphi}_j(\boldsymbol{x}) = \frac{1}{\lambda_j}\sum_{k=1}^{K} (\boldsymbol{v}_j)_k \boldsymbol{u}(\boldsymbol{x}, t_k),\ j = 1, \ldots, r$. Here, $(\boldsymbol{v}_j)_k$ is the k^{th} component of the j^{th} eigenvector $\boldsymbol{v}_j$. Importantly, choosing a basis of dimension $r < R$ leads to an approximation error equal to the sum of the discarded eigenvalues [58]:

$$\frac{1}{K+1} \sum_{k=1}^{K} \left\| \boldsymbol{u}(\boldsymbol{x}, t_k) - \sum_{j=1}^{r} \Big(\boldsymbol{u}(\boldsymbol{x}, t_k), \boldsymbol{\varphi}_j(\boldsymbol{x})\Big) \boldsymbol{\varphi}_j(\boldsymbol{x}) \right\|^2 = \sum_{j=r+1}^{R} \lambda_j. \tag{1.3.2}$$

The POD algorithm introduced above is now employed to define a reduced velocity space $\text{span}\{\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}^r\} = \boldsymbol{X}^r \subset \boldsymbol{X}^h$. Finally, we introduce S^r, the ROM stiffness matrix corresponding to the diffusive term, with components $S^r_{ij} = \big(\nabla\boldsymbol{\varphi}_i, \nabla\boldsymbol{\varphi}_j\big)$. The spectral norm $\|S^r\|_2 = \sigma_{\max}(S^r)$ of this matrix is important in the numerical analysis of the error introduced by POD, as illustrated next. Furthermore, it also appears in the following POD inverse inequality

$$\|\nabla \boldsymbol{v}^r\| \leq \sqrt{\|S^r\|_2}\, \|\boldsymbol{v}^r\| \qquad \forall \boldsymbol{v}^r \in \boldsymbol{X}^r, \tag{1.3.3}$$

as shown in [58, Lemma 2 and Remark 2].

1.3.2 ROM Projection

In this section, we present the concept of the Reduced Order Model (ROM) projection and discuss several fundamental results that are critical for the numerical analysis of ADL-ROM in Chapter 4. The ROM projection is a crucial mathematical tool for reducing the complexity of high-dimensional systems while preserving their essential features within a reduced subspace.

Definition 1.3.2 (ROM Projection)**.** For any $\boldsymbol{u} \in L^2(\Omega; \mathbb{R}^d)$, define $\boldsymbol{P}^r(\boldsymbol{u})$ as the unique element of the reduced space $\boldsymbol{X}^r$ such that

$$\big(\boldsymbol{P}^r(\boldsymbol{u}), \boldsymbol{v}^r\big) = (\boldsymbol{u}, \boldsymbol{v}^r) \qquad \forall\, \boldsymbol{v}^r \in \boldsymbol{X}^r. \tag{1.3.4}$$

Here, $\boldsymbol{X}^r$ is the reduced subspace spanned by the POD basis functions, and the inner product is defined in the standard L^2 sense. The ROM projection $\boldsymbol{P}^r$ can be interpreted as the L^2-orthogonal projection of $\boldsymbol{u}$ onto $\boldsymbol{X}^r$.

The ROM projection plays a key role in reduced order modeling by providing a mechanism to approximate high-dimensional solutions using a lower-dimensional representation. For practical applications, such as fluid flow simulations, this approach ensures an high computational efficiency.

In the numerical analysis of ADL-ROM (Section 4.2), the ROM projection is applied to the sequence $\{\boldsymbol{u}_k\}_{k=0}^K$, i.e., the sequence of weak velocity solutions. The following lemma, proven in [49, Lemma 3.3], provides error bounds for the ROM projection.

Lemma 1.3.3. *The sequence of ROM projections* $\big\{\boldsymbol{P}^r(\boldsymbol{u}_k)\big\}_{k=0}^K$ *satisfies the* *following inequalities*

$$\frac{1}{K+1}\sum_{k=0}^{K}\|\boldsymbol{u}_k-\boldsymbol{P}^r(\boldsymbol{u}_k)\|^2\leq C\left(h^{2m+4}+\Delta t^4+\sum_{j=r+1}^{R}\lambda_j\right),\tag{1.3.5}$$

$$\begin{aligned}\frac{1}{K+1}\sum_{k=0}^{K}\|\nabla(\boldsymbol{u}_k-\boldsymbol{P}^r(\boldsymbol{u}_k))\|^2\leq C\Big(&h^{2m+2}+\|S^r\|_2h^{2m+4}\\&+(1+\|S^r\|_2)\Delta t^4+\sum_{j=r+1}^{R}\|\nabla\boldsymbol{\varphi}_j\|^2\lambda_j\Big).\end{aligned}\tag{1.3.6}$$

This lemma can be interpreted as follows

- **First inequality** (1.3.5)**:** This provides the L^2 error bound for the ROM projection. The error depends on the spatial discretization parameter h, the time step Δt, and the neglected eigenvalues $\{\lambda_j\}_{j=r+1}^{R}$. These terms quantify the contributions of spatial resolution, temporal resolution, and the truncation of POD modes, respectively.

- **Second inequality** (1.3.6)**:** This captures the H^1 error bound (gradient error), which is influenced by similar factors, but includes the spectral norm $\|S^r\|_2$ of the reduced order stiffness matrix, highlighting its impact on the gradient's accuracy.

As in [99, Assumption 4.1], we have to introduce a stronger assumption for later use in Section 2.1 and in the proof of Theorem 4.2.3.

Assumption 1.3.1. *The sequence of ROM projections $\left\{\boldsymbol{P}^r(\boldsymbol{u}_k)\right\}_{k=0}^{K}$ satisfies the pointwise-in-time error bounds*

$$\|\boldsymbol{u}_k-\boldsymbol{P}^r(\boldsymbol{u}_k)\|^2\leq C\left(h^{2m+4}+\Delta t^4+\sum_{j=r+1}^{R}\lambda_j\right),\tag{1.3.7}$$

$$\begin{aligned}\|\nabla(\boldsymbol{u}_k-\boldsymbol{P}^r(\boldsymbol{u}_k))\|^2\leq C\Big(&h^{2m+2}+\|S^r\|_2h^{2m+4}+(1+\|S^r\|_2)\Delta t^4\\&+\sum_{j=r+1}^{R}\|\nabla\boldsymbol{\varphi}_j\|^2\lambda_j\Big).\end{aligned}\tag{1.3.8}$$

Remark 1.3.4. The pointwise-in-time error bounds stated in Assumption 1.3.1 play a critical role in the proof of Theorem 4.2.3. Specifically, the inequality

(1.3.8) is utilized in the derivation of Lemma 4.2.2. Without these bounds, the analysis leading to inequality (4.2.33) would rely on alternative estimates that yield suboptimal results. Indeed, as highlighted in the proof of Lemma 4.2.2, replacing (1.3.8) with (1.3.6) directly would lead to a less favorable bound for the desired inequality.

The necessity of Assumption 1.3.1 has been previously justified in the literature. For example, in [55] (see also Remark 3.2 in [49]), we demonstrated that relying solely on the bounds in Lemma 1.3.3 can result in suboptimal error estimates. This was further substantiated in [55] through two carefully constructed counterexamples, which showed that when Assumption 1.3.1 does not hold, the resulting error bounds fail to achieve optimality.

To address this issue, a more robust approach was proposed in [55], where we proved that incorporating both the snapshots and the snapshot difference quotients in the construction of the reduced order model (ROM) basis enables the derivation of optimal error bounds without relying on Assumption 1.3.1. Subsequent research has refined this methodology. For instance, [24] and [28] showed that optimal error bounds can still be achieved by using only the snapshot difference quotients along with either the snapshot at the initial time or the mean value of the snapshots. These advances were further extended in the recent work [29], which introduced additional simplifications and improvements to the theoretical framework.

Despite these advancements, for the sake of simplicity and to focus on the primary contributions of this thesis, we have chosen not to include snapshot difference quotients in our analysis. Instead, we directly assume the validity of the pointwise-in-time error bounds specified in Assumption 1.3.1. This assumption allows us to streamline the presentation and focus on the key theoretical insights without delving into the additional complexities associated with alternative ROM basis construction techniques. ♠

In the end, the ROM projection is a cornerstone of reduced order modeling. Its theoretical properties, particularly the error bounds presented here, provide the foundation for developing efficient and accurate ROMs for complex dynamical systems.

1.3.3 Galerkin ROM

The Galerkin Reduced Order Model (G-ROM) combines Proper Orthogonal Decomposition (POD) with the Galerkin projection to reduce the computational dimension of the system corresponding to solving the Navier-Stokes equations (NSE) for complex fluid dynamics. By leveraging POD truncation, the G-ROM approximates the velocity field $\boldsymbol{u}(\boldsymbol{x}, t)$ as a linear combination of a reduced set of POD basis functions, which are derived from a set of high-fidelity simulation data (snapshots). Mathematically, this approximation is expressed as

$$\boldsymbol{u}(\boldsymbol{x},t) \approx \boldsymbol{u}^r(\boldsymbol{x},t) \doteq \sum_{j=1}^{r} a_j(t)\boldsymbol{\varphi}_j(\boldsymbol{x}), \tag{1.3.9}$$

where $\boldsymbol{u}(\boldsymbol{x},t)$ represents the full order velocity field, $\boldsymbol{u}^r(\boldsymbol{x},t)$ is its reduced order approximation, $\{\boldsymbol{\varphi}_j\}_{j=1}^r$ are the POD basis functions, and $\{a_j(t)\}_{j=1}^r$ are time-dependent coefficients to be determined.

Derivation of the G-ROM To derive the G-ROM, we substitute the reduced order approximation $\boldsymbol{u}^r$ into the NSE (1.1.1)-(1.1.2). Then, we apply the proper Galerkin method by testing the resulting equations with the POD basis functions $\boldsymbol{\varphi}_j$, $j = 1, \ldots, r$, which form an orthonormal basis of the reduced space $\boldsymbol{X}^r$. In other words, we project the resulting equations onto the ROM space $\boldsymbol{X}^r$. This process leads to the formulation of the G-ROM for the NSE, which can be stated as: Find $\boldsymbol{u}^r \in \boldsymbol{X}^r$ such that

$$\left(\frac{\partial \boldsymbol{u}^r}{\partial t}, \boldsymbol{\varphi}\right) + \nu\left(\nabla \boldsymbol{u}^r, \nabla \boldsymbol{\varphi}\right) + b^*\left(\boldsymbol{u}^r, \boldsymbol{u}^r, \boldsymbol{\varphi}\right) = (\boldsymbol{f}, \boldsymbol{\varphi}), \quad \forall \boldsymbol{\varphi} \in \boldsymbol{X}^r, \tag{1.3.10}$$

with the initial condition $\boldsymbol{u}^r(\cdot, 0) \in \boldsymbol{X}^r$. Here, b^* represents the nonlinear convective term

$$b^*(\boldsymbol{u}^r, \boldsymbol{u}^r, \boldsymbol{\varphi}) = \int_\Omega (\boldsymbol{u}^r \cdot \nabla)\boldsymbol{u}^r \cdot \boldsymbol{\varphi}\, d\boldsymbol{x}. \tag{1.3.11}$$

The ROM velocity fields $\boldsymbol{u}^r \in \boldsymbol{X}^r$ are inherently weakly-divergence free due to the construction of the POD basis from divergence free snapshots. Consequently, the pressure term in the NSE does not appear in (1.3.10). If pressure approximation is desired, alternative ROM formulations, such as those using supremizer enrichment [10] or artificial stabilization [22], must be employed. These approaches address the need for pressure contributions in certain applications, particularly when enforcing boundary conditions or analyzing pressure-driven flows [71, 82, 83, 90].

The accuracy and stability of the G-ROM have been extensively analyzed in the literature. Studies have explored error bounds for both spatial and temporal discretizations, addressing their dependence on parameters such as the viscosity ν and the number of retained modes r. For details, see, e.g., [20, 48, 56, 58, 57, 65, 89]. These works provide a rigorous theoretical foundation for understanding the performance of the G-ROM in various regimes.

Matrix-Vector Form of G-ROM Using the expansion (1.3.9), the G-ROM can be rewritten as a system of differential equations for the vector of time coefficients, $\boldsymbol{a} = \boldsymbol{a}(t)$ as follows:

$$\boldsymbol{a}' = \boldsymbol{b} + A\boldsymbol{a} + \boldsymbol{a}^\top B\boldsymbol{a}, \tag{1.3.12}$$

where $\boldsymbol{a}' = \frac{\partial \boldsymbol{a}}{\partial t}$, and the vector $\boldsymbol{b}$, the matrix A, and the tensor B are defined as follows

$$b_i = (\boldsymbol{\varphi}_i, \boldsymbol{f}), \tag{1.3.13}$$

$$A_{im} = -\nu \left(\nabla\boldsymbol{\varphi}_i, \nabla\boldsymbol{\varphi}_m\right), \tag{1.3.14}$$

$$B_{imn} = -\left(\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_m \cdot \nabla\boldsymbol{\varphi}_n\right), \tag{1.3.15}$$

for $i, m, n = 1, \ldots, r$.

Remark 1.3.5 (Centering Trajectory)**.** In the case of inhomogeneous BCs, $\boldsymbol{U}^h$ will appear in (1.3.9), (2.2.2), (3.1.3), (3.1.9), and (3.1.14). The resulting expansions will be the following:

$$\boldsymbol{u}^r(\boldsymbol{x}, t) = \boldsymbol{U}^h(\boldsymbol{x}) + \sum_{i=1}^{r} a_i(t)\boldsymbol{\varphi}_i(\boldsymbol{x}), \qquad \boldsymbol{a} \doteq [a_1, \ldots, a_r]^\top, \tag{1.3.16}$$

$$\overline{\boldsymbol{u}^r}(\boldsymbol{x}, t) = \boldsymbol{U}^h(\boldsymbol{x}) + \sum_{i=1}^{r} \overline{a}_i(t)\boldsymbol{\varphi}_i(\boldsymbol{x}), \qquad \overline{\boldsymbol{a}} \doteq [\overline{a}_1, \ldots, \overline{a}_r]^\top, \tag{1.3.17}$$

$$\boldsymbol{u}^r_{AD}(\boldsymbol{x}, t) = \boldsymbol{U}^h(\boldsymbol{x}) + \sum_{i=1}^{r} a_{AD,i}(t)\boldsymbol{\varphi}_i(\boldsymbol{x}), \quad \boldsymbol{a}_{AD} \doteq [a_{AD,1}, \ldots, a_{AD,r}]^\top. \tag{1.3.18}$$

Moreover, the vector $\boldsymbol{b}$, the matrix A, and the tensor B will be defined as follows:

$$b_i = \big(\boldsymbol{\varphi}_i, \boldsymbol{f}\big) - \big(\boldsymbol{\varphi}_i, \boldsymbol{U}^h \cdot \nabla\boldsymbol{U}^h\big) - Re^{-1}\big(\nabla\boldsymbol{\varphi}_i, \nabla\boldsymbol{U}^h\big), \tag{1.3.19}$$

$$A_{im} = -\big(\boldsymbol{\varphi}_i, \boldsymbol{U}^h \cdot \nabla\boldsymbol{\varphi}_m\big) - \big(\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_m \cdot \nabla\boldsymbol{U}^h\big) - Re^{-1}\big(\nabla\boldsymbol{\varphi}_i, \nabla\boldsymbol{\varphi}_m\big), \tag{1.3.20}$$

$$B_{imn} = -\big(\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_m \cdot \nabla\boldsymbol{\varphi}_n\big), \tag{1.3.21}$$

for $i, n, m = 1, \ldots, r$.

We note that a nonzero centering trajectory will actually be required in Section 3.3 to handle the nonzero inlet conditions. The methodology introduced in this chapter readily extends to that case, which yields additional terms on the right-hand side of (2.1.10), (3.1.6), (3.1.11), and (3.1.15). For instance, following the same approach as that used in (1.3.19), we obtain the following modification of (2.1.10)

$$\left(M^r + \delta^2 S^r\right) \bar{\boldsymbol{c}} = M^r \boldsymbol{c} + \boldsymbol{g},$$

where

$$\boldsymbol{g}_i \;=\; -\big(\boldsymbol{\varphi}_i, \boldsymbol{U}^h\big) - \delta^2 \big(\nabla \boldsymbol{\varphi}_i, \nabla \boldsymbol{U}^h\big)\,.$$

Time Integration Scheme for the G-ROM In practice, solving the G-ROM requires temporal discretization. A useful choice is the second order backward differentiation formula (BDF2). As said before, our time-scheme choice is not mandatory and all the theoretical and computational results shown in this thesis are valid for other time schemes.

The BDF2 scheme for the G-ROM yields: For $\Delta t > 0$ and for $k = 1, \ldots, K-1$, given $\boldsymbol{u}_{k-1}^r, \boldsymbol{u}_k^r \in \boldsymbol{X}^r$, find $\boldsymbol{u}_{k+1}^r \in \boldsymbol{X}^r$ such that

$$\frac{\left(3\boldsymbol{u}_{k+1}^r - 4\boldsymbol{u}_k^r + \boldsymbol{u}_{k-1}^r, \boldsymbol{v}^r\right)}{2\Delta t} + \nu \left(\nabla \boldsymbol{u}_{k+1}^r, \nabla \boldsymbol{v}^r\right) + b^* \left(\boldsymbol{u}_{k+1}^r, \boldsymbol{u}_{k+1}^r, \boldsymbol{v}^r\right) = \left(\boldsymbol{f}_{k+1}, \boldsymbol{v}^r\right), \quad \forall \boldsymbol{v}^r \in \boldsymbol{X}^r. \tag{1.3.22}$$

In Algorithm 1, we outline the main steps of the G-ROM discretization.

Algorithm 1 G-ROM Pseudocode

$\boldsymbol{u}_{-1}$, $\boldsymbol{u}_0$, $\boldsymbol{u}_{in}$, r ▷ Inputs needed
for $k \in \{1, \ldots, K-1\}$ **do** ▷ Time loop
    FOM simulation to compute $\boldsymbol{u}_{k+1}^h$ ▷ Snapshot collection
end for
$\boldsymbol{X}^r \doteq \mathrm{POD}\left(\{\boldsymbol{u}_k^h\}_{k=1}^K\,; r\right)$ ▷ POD for velocity space
for $k \in \{1, \ldots, K-1\}$ **do** ▷ Time loop
    Solve (1.3.22) to compute $\boldsymbol{u}_{k+1}^r$ ▷ G-ROM
end for

While the G-ROM offers significant computational savings due to POD truncation, its performance is often limited to laminar flows or mildly turbulent regimes. This limitation arises from the model's inability to fully capture the complex dynamics of turbulence using a reduced basis. Extensions such as Regularized ROMs (Reg-ROMs) have been developed to overcome these challenges and enhance the applicability of ROMs to more complex flow scenarios. In particular, the aim of the rest of this thesis is to fully introduce, describe and analyze our novel Reg-ROM, i.e., the Approximate Deconvolution Leray ROM (ADL-ROM).

Chapter 2

Leray Model

The ROM Differential Filter (ROM-DF) represents a fundamental tool in the numerical analysis of reduced order models (ROMs) applied to complex systems such as fluid dynamics. In this chapter, we focus on establishing theoretical bounds for the ROM-DF, providing insights into its stability and error characteristics. These bounds are crucial for understanding how the ROM-DF interacts with various sources of approximation error, including those arising from finite element discretization, the Proper Orthogonal Decomposition (POD) process, and the filtering operations. The ROM-DF is one of the most important ingredients for constructing the Approximate Deconvolution Leray ROM (ADL-ROM), which we propose in Chapter 3.

In addition to analyzing the ROM-DF, this chapter introduces and describes the Leray Reduced Order Model (L-ROM), a modified ROM framework designed to enhance the numerical stability and physical fidelity of reduced order simulations. The Leray ROM achieves this by incorporating the ROM-DF directly into the ROM formulation. This approach effectively mitigates the propagation of high-frequency errors and improves the overall accuracy of the reduced order solution.

The L-ROM is particularly important in applications where turbulent flows are involved, as it provides a more robust representation of the underlying physical phenomena while maintaining computational efficiency. By blending the advantages of ROM-DF and the Leray strategy, this model addresses critical challenges in reduced order modeling, such as numerical instabilities and sensitivity to the choice of basis functions. One of the essential issues with L-ROM is that, in addressing the inaccuracy of G-ROM for turbulent flows, it often produces overdiffusive solutions. Thus, we wanted to give an overview of L-ROM in this thesis since it was our starting point to construct ADL-ROM. Indeed, we wanted to create a new Reg-ROM capable of overcoming the inaccuracies of G-ROM by filtering them as L-ROM, but not too overdiffusive.

2.1 Differential Filters

In this section, we thoroughly present a continuous spatial filter and two discrete spatial filters, i.e., the finite element and the reduced order ones. These filters serve as fundamental components in the construction of advanced ROM frameworks. Specifically, in Chapters 3 and 4, we focus exclusively on the explicit ROM spatial filtering approach to formulate our novel Reg-ROM, known as the ADL-ROM.

2.1.1 Explicit ROM Spatial Filtering

To establish the foundation of our new framework, we adopt the explicit Differential Filter (DF). This section delineates the continuous formulation of the DF in Section 2.1.1.1, along with two discrete variants detailed in Sections 2.1.1.2 and 2.1.1.3.

The Differential Filter (DF) was initially introduced in the field of Large Eddy Simulation (LES) by Germano [31, 30]. Its utility was later extended to ROMs, providing a robust mechanism for developing regularized ROM frameworks. Notably, the ROM-DF was first applied in [85] to the 1D Kuramoto-Sivashinsky equation under periodic boundary conditions. Subsequently, the ROM-DF was employed in [96] for the three-dimensional Navier-Stokes equations (3D NSE) within a general, non-periodic domain configuration.

2.1.1.1 Continuous Differential Filter (C-DF)

The Continuous Differential Filter (C-DF) represents a crucial tool for constructing various models in computational science. Its strong formulation is presented as follows:

Definition 2.1.1 (Continuous Differential Filter)**.** Let $\delta \in \mathbb{R}^+$ be a fixed filter radius, determining the scale of filtering. For a given function $\boldsymbol{\Phi} \in C^0(\Omega;\ \mathbb{R}^d)$, the continuous differential filter is a map

$$F : C^0(\Omega;\ \mathbb{R}^d) \to C^2(\Omega;\ \mathbb{R}^d), \quad \boldsymbol{\Phi} \mapsto \overline{\boldsymbol{\Phi}},$$

such that $\overline{\boldsymbol{\Phi}} \in C^2(\Omega;\ \mathbb{R}^d)$ is the unique solution to the elliptic boundary value problem

$$\begin{cases} (I - \delta^2\Delta)\overline{\boldsymbol{\Phi}} = \boldsymbol{\Phi} & \text{in } \Omega, \\ \overline{\boldsymbol{\Phi}} = \mathbf{0} & \text{on } \partial\Omega. \end{cases} \tag{2.1.1}$$

The C-DF formulation in (2.1.1) employs a linear spatial filtering operator to achieve smoothing of input variables by attenuating high-frequency components (small scales). This process ensures that the resulting data retains only the large-scale structures, as discussed in [15].

Despite its theoretical appeal, the direct implementation of the C-DF in combination with a Full Order Model (FOM) or a Reduced Order Model (ROM) poses practical challenges. Specifically, the reliance on a strong formulation of the elliptic operator $(I - \delta^2\Delta)$ necessitates an alternative approach. For practical applications, a weak formulation of $(I-\delta^2\Delta)$ is utilized, which underpins the development of the discrete differential filters introduced in subsequent sections.

One noteworthy property of the continuous differential filter is stated in the following lemma, which was proved in [53].

Lemma 2.1.2. *The Continuous Differential Filter, as defined in 2.1.1, is a self-adjoint operator on $L^2(\Omega;\ \mathbb{R}^d)$.*

Proof. Let $\mathbf{\Phi}, \mathbf{\Psi} \in L^2(\Omega;\ \mathbb{R}^d)$. By the definition of the C-DF, the filtered function $\overline{\mathbf{\Psi}}$ satisfies

$$\overline{\mathbf{\Psi}} - \delta^2\Delta\overline{\mathbf{\Psi}} = \mathbf{\Psi} \qquad \text{in } \Omega, \tag{2.1.2}$$

$$\overline{\mathbf{\Psi}} = \mathbf{0} \qquad \text{on } \partial\Omega. \tag{2.1.3}$$

Multiplying (2.1.2) by $\overline{\mathbf{\Phi}}$, integrating over Ω, and applying integration by parts (noting that $\overline{\mathbf{\Phi}} = \mathbf{0}$ on $\partial\Omega$), we obtain

$$\delta^2(\nabla\overline{\mathbf{\Psi}}, \nabla\overline{\mathbf{\Phi}}) + (\overline{\mathbf{\Psi}}, \mathbf{\Phi}) = (\mathbf{\Psi}, \overline{\mathbf{\Phi}}) \quad \text{in } \Omega. \tag{2.1.4}$$

Similarly, multiplying the first equation in (2.1.1) by $\overline{\mathbf{\Psi}}$, integrating by parts, and using (2.1.3), we arrive at

$$\delta^2(\nabla\overline{\mathbf{\Phi}}, \nabla\overline{\mathbf{\Psi}}) + (\overline{\mathbf{\Phi}}, \overline{\mathbf{\Psi}}) = (\mathbf{\Phi}, \overline{\mathbf{\Psi}}) \quad \text{in } \Omega. \tag{2.1.5}$$

Equations (2.1.4) and (2.1.5) prove the lemma. □

2.1.1.2 Finite Element Differential Filter (FE-DF)

The Finite Element Differential Filter (FE-DF) represents a discrete implementation of the differential filter, tailored for finite element (FE) spaces. This method is essential in numerical simulations, as it effectively enables spatial filtering while conforming to the discretized finite element framework. The formal definition of the FE-DF is

Definition 2.1.3 (Finite Element Differential Filter)**.** Let $\delta \in \mathbb{R}^+$ be a fixed filter radius that dictates the scale of the filtering operation. Given an input function $\mathbf{\Phi} \in L^2(\Omega;\ \mathbb{R}^d)$, the FE differential filter is a map

$$F^h : L^2(\Omega;\ \mathbb{R}^d) \to \boldsymbol{X}^h, \quad \mathbf{\Phi} \mapsto \overline{\mathbf{\Phi}},$$

where $\boldsymbol{X}^h$ denotes the finite-dimensional FE space. The filtered output $\overline{\mathbf{\Phi}} \in \boldsymbol{X}^h$ is determined as the unique solution to the following boundary value problem

$$\begin{cases} (I - \delta^2 \Delta)\overline{\mathbf{\Phi}} = \mathbf{\Phi} & \text{in } \Omega, \\ \overline{\mathbf{\Phi}} = \mathbf{0} & \text{on } \partial\Omega. \end{cases} \tag{2.1.6}$$

While (2.1.6) is formulated in strong form and is applicable to any function $\boldsymbol{u} \in \boldsymbol{X} \subset L^2(\Omega;\ \mathbb{R}^d)$, its practical implementation is limited to finite element (FE) velocity fields $\boldsymbol{u}^h$. Consequently, solving the boundary value problem described in (2.1.6) translates, in practice, to solving a linear system obtained through the finite element discretization. This linear system is given by

$$\left(M^h + \delta^2 S^h\right) \overline{\boldsymbol{a}^h} = M^h \boldsymbol{a}^h, \tag{2.1.7}$$

where $M^h \in \mathbb{R}^{D \times D}$ is the finite element mass matrix. This matrix arises from the inner product of basis functions and accounts for the discretization of the L^2 inner product. $S^h \in \mathbb{R}^{D \times D}$ is the finite element stiffness matrix. This matrix encapsulates the discretization of the Laplacian operator Δ in the weak form. $\overline{\boldsymbol{a}^h} \in \mathbb{R}^D$ is the vector of finite element coefficients corresponding to the filtered output variable $\overline{\boldsymbol{u}^h}$ and $\boldsymbol{a}^h \in \mathbb{R}^D$ is the vector of finite element coefficients associated with the input variable $\boldsymbol{u}^h$.

From now on, D denotes the dimension of the finite element velocity space $\boldsymbol{X}^h$, i.e., the number of degrees of freedom in the discretized finite element representation[1].

Properties and Practical Considerations The FE-DF framework provides a computationally efficient way to perform spatial filtering in a discretized setting, leveraging the structure of finite element spaces. The combination of the mass matrix M^h and stiffness matrix S^h ensures that the

[1]The dimension D is determined by the choice of the finite element basis and the mesh resolution used in the discretization.

filter preserves essential numerical properties, such as symmetry and positive definiteness, under appropriate conditions.

The parameter δ, representing the filter radius, plays a critical role in determining the extent of filtering. Larger values of δ correspond to stronger smoothing effects, as the filter attenuates higher frequencies more aggressively. Conversely, smaller values of δ yield less filtering, allowing finer scales to persist in the solution.

By solving the linear system (2.1.7), the FE-DF effectively smooths the input data $\boldsymbol{u}^h$, producing a filtered result $\overline{\boldsymbol{u}^h}$ that retains only the desired large-scale structures. This makes the FE-DF an indispensable tool for regularization and stabilization in numerical simulations, particularly in contexts such as Reduced Order Modeling and Large Eddy Simulation.

2.1.1.3 ROM Differential Filter (ROM-DF)

The Reduced Order Model Differential Filter (ROM-DF) is a discrete spatial filter specifically designed for reduced order models (ROMs). It is particularly well-suited for computationally efficient simulations, as it operates within the low-dimensional subspace spanned by the ROM basis functions. The formal definition of the ROM-DF is as follows:

Definition 2.1.4 (ROM Differential Filter)**.** Let $\delta \in \mathbb{R}^+$ be a constant filter radius, which controls the scale of filtering. Given an input function $\boldsymbol{\Phi} \in L^2(\Omega;\ \mathbb{R}^d)$, the ROM differential filter is a map

$$F^r : L^2(\Omega;\ \mathbb{R}^d) \to \boldsymbol{X}^r, \quad \boldsymbol{\Phi} \mapsto \overline{\boldsymbol{\Phi}},$$

where $\boldsymbol{X}^r$ is the reduced order velocity space. The filtered output $\overline{\boldsymbol{\Phi}} \in \boldsymbol{X}^r$ is defined as the unique solution to the following weak formulation

$$\delta^2(\nabla\overline{\boldsymbol{\Phi}}, \nabla\boldsymbol{v}^r) + (\overline{\boldsymbol{\Phi}}, \boldsymbol{v}^r) = (\boldsymbol{\Phi}, \boldsymbol{v}^r) \quad \forall \boldsymbol{v}^r \in \boldsymbol{X}^r. \tag{2.1.8}$$

In this context, the operator F^r applies the filter to the input $\boldsymbol{\Phi}$ to produce a smoothed output $\overline{\boldsymbol{\Phi}}$. For simplicity, the shorthand notation $\overline{\bullet}$ will be used to represent the filtered result of any input $\bullet$, which will typically be a velocity field $\boldsymbol{u} \in \boldsymbol{X} \subset L^2(\Omega;\ \mathbb{R}^d)$.

We note that the filtering operator $\overline{\bullet}$ depends on the dimension r of the reduced velocity space $\boldsymbol{X}^r$. However, to avoid notational clutter, the parameter r is not explicitly included in the overline notation.

The ROM-DF operator $F^r = (I - \delta^2\Delta)^{-1}$ can equivalently be expressed in terms of a PDE. For a given input function $\boldsymbol{u}$, the ROM-DF involves solving the following boundary value problem

$$\begin{cases} (I - \delta^2 \Delta)\overline{\boldsymbol{u}} = \boldsymbol{u}, & \text{in } \Omega, \\ \overline{\boldsymbol{u}} = \boldsymbol{0}, & \text{on } \partial\Omega, \end{cases} \tag{2.1.9}$$

where $\partial\Omega$ is the boundary of the spatial domain Ω. This formulation highlights the direct relationship between the ROM-DF and the elliptic operator $(I - \delta^2 \Delta)$.

Although (2.1.9) is written in strong form and applies to any function $\boldsymbol{u}$, in practice, the ROM-DF is applied to ROM velocity fields $\boldsymbol{u}^r$. To do so, the Galerkin method is employed to discretize the PDE (2.1.9) in the reduced space spanned by the ROM basis functions $\{\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_r\}$. This leads to the following linear system

$$\left(M^r + \delta^2 S^r\right) \overline{\boldsymbol{a}} = M^r \boldsymbol{a}, \tag{2.1.10}$$

where $M^r \in \mathbb{R}^{r\times r}$ is the POD mass matrix (which is the identity matrix since the POD basis functions are orthonormal), $S^r \in \mathbb{R}^{r\times r}$ is the POD stiffness matrix, $\overline{\boldsymbol{a}} \in \mathbb{R}^r$ is the vector of POD coefficients of the output filtered variable $\overline{\boldsymbol{u}^r}$, and $\boldsymbol{a} \in \mathbb{R}^r$ is the vector of POD coefficients of the input variable $\boldsymbol{u}^r$.

The entries of the mass and stiffness matrices are computed as

$$M^r_{im} = \left(\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_m\right), S^r_{im} = \left(\nabla\boldsymbol{\varphi}_i, \nabla\boldsymbol{\varphi}_m\right), \quad i, m = 1, \ldots, r.$$

Efficiency and Computational Advantage One of the key advantages of the ROM-DF lies in its computational efficiency. The size of the linear system in (2.1.10) is $r \times r$, where r is the dimension of the reduced order space. This is significantly smaller[2] than the $D \times D$ system associated with the FE-DF. As a result, the ROM-DF offers a substantial reduction in computational cost, making it an attractive option for complex numerical simulations, e.g., turbulent flow simulations.

Moreover, the user-defined filter radius δ provides flexibility in controlling the extent of smoothing, allowing the ROM-DF to be tailored to specific problem requirements. By carefully selecting δ, the ROM-DF can effectively balance the trade-off between filtering out small-scale noise and preserving important large-scale features in the solutions.

[2] In order to have a meaningful ROM, both G-ROM and Reg-ROMs, we have to choose $r \ll D$. Otherwise, the ROMs lose their excellent computational efficiency in comparison to FEM.

2.1.2 ROM-DF Bounds

This section provides a brief examination of the theoretical bounds associated with the ROM-DF. These bounds quantify the sources of error that arise from various aspects of the filtering process.

To establish these bounds, we present key lemmas from the literature, including results from [99] and [59]. These lemmas provide insights into the stability and accuracy of the ROM-DF and highlight differences between the ROM and FEM frameworks.

The following lemma, originally proven in [99, Lemma 4.3], provides an important bound that forms the foundation of the ROM-DF error analysis. For clarity and conciseness, we reproduce the lemma below with a slight modification: we omit an additional term on the left-hand side of the inequality, as it is not utilized in the scope of our analysis.

The bound in Lemma 2.1.5 plays a critical role in categorizing the different sources of error that emerge during the filtering process. These error sources can be grouped into four distinct categories: those originating from the FE data, those resulting from the POD process, those associated with the filtering procedure itself, and finally, terms that involve interactions or combinations of these factors.

Lemma 2.1.5. *Let* $\boldsymbol{u}_k = \boldsymbol{u}(\boldsymbol{x}, t_k) \in \boldsymbol{X}$ *denote a solution to* (1.2.1)*, with the additional regularity condition* $\Delta \boldsymbol{u}_k \in L^2(\Omega)$*. Assuming that Assumption 1.3.1 holds, the following inequality holds, for all* k*,*

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

In addition to the above lemma, we recall several useful estimates from [99, Lemma 4.4], which are presented below.

Lemma 2.1.6. *Let* $\boldsymbol{u} \in \boldsymbol{X}$*. Then, the following bounds hold*

$$
\|\overline{\boldsymbol{u}}\| \leq \|\boldsymbol{u}\|, \tag{2.1.11}
$$

$$
\|\nabla \overline{\boldsymbol{u}}\| \leq \sqrt{\|S^r\|_2}\, \|\boldsymbol{u}\|. \tag{2.1.12}
$$

Furthermore, if $\boldsymbol{u} \in \boldsymbol{X}^r$*, then the stronger bound holds*

$$\|\nabla \overline{\boldsymbol{u}}\| \leq \|\nabla \boldsymbol{u}\|. \tag{2.1.13}$$

Remark 2.1.7. The stability bounds outlined in Lemma 2.1.6, originally proven in [99, Lemma 4.4], exhibit notable differences compared to the FE stability bounds presented in [59, Lemma 2.11].

Specifically, in the ROM context, we establish two distinct bounds in Lemma 2.1.6:

- For $\boldsymbol{u} \in \boldsymbol{X}$, we derive the bound given in (2.1.12).
- For $\boldsymbol{u} \in \boldsymbol{X}^r$, we derive the stronger bound stated in (2.1.13).

In contrast, within the FEM framework, the stronger bound (analogous to (2.1.13)) is established for $\boldsymbol{u} \in \boldsymbol{X}$ in inequality (2.22) of [59]. The derivation of this FEM bound relies on the use of the Ritz projection, as communicated by William Layton (personal communication). However, extending this approach to the ROM setting remains challenging, as the approximation properties of the Ritz projection within the context of POD are not fully understood [48, 55]. As a result, proving the ROM analog of (2.1.13) for $\boldsymbol{u} \in \boldsymbol{X}$ is an open problem in this area of research. ♠

Finally, it is worth noting that (2.1.11) suggests the definition of an auxiliary operator $\widetilde{F}^r : \boldsymbol{X} \to \boldsymbol{X}^r$, defined as $\widetilde{F}^r \doteq F^r|_{\boldsymbol{X}}$. Under this definition, (2.1.11) implies that $\widetilde{F}^r$ is a bounded operator on $\boldsymbol{X}$. However, it is important to highlight that no analogous conclusion can be drawn in general for the operator F^r on the broader space $L^2(\Omega; \mathbb{R}^d) \supset \boldsymbol{X}$.

2.2 Leray ROM

The Leray model is rooted in the pioneering work of Jean Leray in 1934, who introduced the Leray model to provide a theoretical framework for the existence of weak solutions for the Navier-Stokes equations (NSE) [63]. His work was instrumental in laying the mathematical foundation for studying turbulence and fluid dynamics, emphasizing regularization techniques to manage the complexity of these equations. Over the decades, Leray's ideas have inspired numerous advancements in fluid mechanics, particularly in stabilization methods for computational models.

More recently, the Leray model has been adopted as a computational tool for simulating convection-dominated flows, such as turbulent flows, using conventional numerical methods [32]. These methods include the finite element method [62], which allows for the practical implementation of Leray-inspired stabilization strategies. When the differential filter is employed, the Leray model closely resembles the NS-α model proposed by Foias, Holm, Titi, and their collaborators [27], demonstrating its versatility and alignment with other modern approaches.

In the past decade, the Leray model has been successfully extended to the realm of reduced order modeling (ROM). Initially applied to the Kuramoto-Sivashinsky equations [85], it was later adapted for the NSE [95, 96] and the quasi-geostrophic equations [35, 36]. These extensions have opened new avenues for efficiently approximating complex, high-dimensional systems.

The L-ROM employs a stabilized approach by incorporating a filtered velocity field, inspired by Leray's regularization techniques, to address the challenges faced by traditional Galerkin ROMs (G-ROMs). These challenges include spurious energy transfer and numerical instabilities in convection-dominated flows. The filtering process reduces the influence of unresolved scales, enhancing the stability and accuracy of the reduced order model.

2.2.1 L-ROM Time Discretization

In our setting, the L-ROM time discretization is formulated as follows: Given $\boldsymbol{u}_k^r$ and $\boldsymbol{u}_{k-1}^r$, find $\boldsymbol{u}_{k+1}^r$ such that

$$\begin{aligned}\left(\frac{\boldsymbol{u}_{k+1}^r - \frac{4}{3}\boldsymbol{u}_k^r + \frac{1}{3}\boldsymbol{u}_{k-1}^r}{\Delta t}, \boldsymbol{\varphi}_i\right) + \frac{2}{3}\nu\left(\nabla \boldsymbol{u}_{k+1}^r, \nabla \boldsymbol{\varphi}_i\right) + \frac{2}{3}\left(\left(\overline{\boldsymbol{u}_{k+1}^r} \cdot \nabla\right) \boldsymbol{u}_{k+1}^r, \boldsymbol{\varphi}_i\right) \\ = \left(\boldsymbol{f}_{k+1}, \boldsymbol{\varphi}_i\right),\end{aligned} \tag{2.2.1}$$

for all $k = 1, \ldots, K-1$ and $i = 1, \ldots, r$, where Δt is the time step.

Utilizing the ROM-DF (2.1.8) outlined in Section 2.1, the filtered convective term in (2.2.1) is defined by the following expansion[3]

$$\overline{\boldsymbol{u}_{k+1}^r}(\boldsymbol{x}, t) \doteq \sum_{j=1}^{r} \overline{a}_{j,k+1}(t) \boldsymbol{\varphi}_j(\boldsymbol{x}). \tag{2.2.2}$$

The coefficients $\overline{\boldsymbol{a}}_{k+1}$ in (2.2.2) are determined by solving the following reduced linear system

$$\left(M^r + \delta^2 S^r\right) \overline{\boldsymbol{a}}_{k+1} = M^r \boldsymbol{a}_{k+1}, \tag{2.2.3}$$

[3] We assume the same kind of expansion in (1.3.9), i.e., we use the same procedure as for G-ROM.

where M^r and S^r are the POD mass and stiffness matrices, respectively, δ is the radius of the ROM-DF, $\overline{\boldsymbol{a}}_{k+1}$ is the vector of reduced coefficients of the output variable, $\overline{\boldsymbol{u}^r}_{k+1}$, and $\boldsymbol{a}_{k+1}$ is the vector of reduced coefficients of the input variable, $\boldsymbol{u}^r_{k+1}$. This linear system introduces minimal computational overhead due to its reduced dimensionality, as it is of size $r \times r$.

A second order backward difference formula (BDF2) time discretization is employed in (2.2.1), although other time discretizations can be used. At each time step t_{k+1}, the L-ROM time discretization yields the following nonlinear system

$$\frac{1}{\Delta t} M^r \left(\boldsymbol{a}_{k+1} - \frac{4}{3}\boldsymbol{a}_k + \frac{1}{3}\boldsymbol{a}_{k-1} \right) + \frac{2}{3}\nu S^r \boldsymbol{a}_{k+1} + \frac{2}{3} C(\overline{\boldsymbol{a}}_{k+1})\boldsymbol{a}_{k+1} = \boldsymbol{b}_{k+1}, \quad (2.2.4)$$

where $\boldsymbol{a}_{k+1}$ is the unknown reduced coefficient vector of the unfiltered solution at t_{k+1}, $\overline{\boldsymbol{a}}_{k+1}$ is the filtered velocity coefficient vector, $\boldsymbol{b}_{k+1}$ is the vector introduced in (1.3.13), and $C(\overline{\boldsymbol{a}}_{k+1})$ is defined as

$$C(\overline{\boldsymbol{a}}_{k+1})_{ij} = \left((\overline{\boldsymbol{u}^r_{k+1}} \cdot \nabla)\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_j \right). \quad (2.2.5)$$

System (2.2.4) is solved using the Newton method. At each iteration, $\overline{\boldsymbol{a}}_{k+1}$ is updated via (2.2.3), and the residual of (2.2.4) is evaluated. Algorithm 2 outlines the main steps of the L-ROM discretization.

Algorithm 2 L-ROM Pseudocode

$\boldsymbol{u}_{-1}$, $\boldsymbol{u}_0$, $\boldsymbol{u}_{in}$, r ▷ Inputs needed
for $k \in \{1, \ldots, K-1\}$ **do** ▷ Time loop
    Perform FOM simulation to compute $\boldsymbol{u}^h_{k+1}$ ▷ Snapshot collection
end for
$\boldsymbol{X}^r \doteq \text{POD}\left(\left\{ \boldsymbol{u}^h_k \right\}_{k=1}^{K}; r \right)$ ▷ POD for velocity space
for $k \in \{1, \ldots, K-1\}$ **do** ▷ Time loop
    Solve (2.2.4) to compute $\boldsymbol{a}_{k+1}$ ▷ L-ROM
    Solve (2.2.3) at each nonlinear iteration ▷ Leray filtering
end for

2.2.2 Discussion and Limitations

The Leray Reduced Order Model (L-ROM) represents a significant advancement over the standard Galerkin Reduced Order Model (G-ROM) (1.3.22), particularly in its ability to mitigate spurious numerical oscillations that commonly arise in convection-dominated flows. These oscillations, which result from the inherent instability of the G-ROM formulation in such settings, can lead to inaccuracies and inefficiencies, making the G-ROM unreliable for many practical applications involving turbulent, or high Reynolds number, flows. By incorporating spatial filtering into its formulation, the L-ROM effectively dampens these oscillations, enhancing both the stability and physical fidelity of the reduced order solution.

The spatial filtering mechanism in the L-ROM is achieved by introducing a filtered velocity field into the nonlinear convective term. This modification reduces the impact of unresolved modes and suppresses the transfer of energy to smaller, unrepresented scales. As a result, the L-ROM provides a more stable and accurate approximation of the underlying flow dynamics, making it particularly suitable for scenarios where the G-ROM might fail to produce meaningful results.

However, the L-ROM is not without limitations. Excessive filtering, characterized by a large filter radius δ, can result in overdiffusion. This phenomenon, in which the flow dynamics are overly smoothed, leads to the suppression of physically relevant small-scale features, ultimately reducing the accuracy of the model. Such overdiffusion can undermine the benefits of stabilization and compromise the model's ability to capture essential characteristics of the flow, such as fine-grained vortices or sharp gradients.

Moreover, the performance of the L-ROM is highly sensitive to the choice of the filter radius δ. Even small variations in δ can lead to significant changes in the accuracy and stability of the solution. For instance, an overly small δ may fail to adequately stabilize the model, reintroducing numerical oscillations, while a slightly larger δ might oversmooth the solution. This sensitivity necessitates careful calibration of δ based on the specific problem and flow characteristics, i.e., the FEM mesh size h, under investigation. In summary, while the L-ROM offers substantial improvements over the G-ROM, its effectiveness depends on judicious parameter selection, particularly the filter radius δ.

These reasons were the key drivers behind the start of our research. Our goal was to develop a new Reg-ROM, ADL-ROM, that retained the stabilizing benefits of spatial filtering while preserving the essential physical features of the flow, such as the accurate representation of physical vortices. In other words, we wanted to combine the better parts of G-ROM, i.e., preserving the physical features of the flow, and L-ROM, i.e., smoothing out the spurious oscillations.

Chapter 3

Approximate Deconvolution Leray Model

Despite the significant improvements achieved by the L-ROM over the standard G-ROM, numerical investigations reveal two key drawbacks of the L-ROM. First, it exhibits an overdiffusive nature when the ROM filter radius, δ, is excessively large. In such cases, the filtering process introduces an undue amount of dissipation, compromising the model accuracy. Second, the L-ROM demonstrates a high sensitivity to variations in the ROM filter radius. Even minor deviations from the optimal filter radius can lead to noticeable reductions in the model accuracy, highlighting a potential limitation in its robustness.

To address these drawbacks of the L-ROM, we proposed a novel Regularized ROM (Reg-ROM) in [86], known as the *approximate deconvolution Leray ROM (ADL-ROM)*. This innovative approach represents a significant improvement over the L-ROM. To our knowledge, this was the first instance of utilizing the approximate deconvolution (AD) framework to construct ROM stabilizations such as Reg-ROMs. Moreover, in reduced order modeling, the AD concept had only been employed once before, in [98], for a distinctly different purpose, namely, to develop a ROM closure model.

The core innovation of the ADL-ROM lies in leveraging approximate deconvolution, a classical methodology widely used in the image processing and inverse problems communities, to enhance the accuracy and robustness of the L-ROM. Specifically, in the ADL-ROM, the filtered L-ROM velocity, $\overline{\boldsymbol{u}^r}$, is replaced by the AD velocity, $D^r(\overline{\boldsymbol{u}^r})$. Consequently, the ADL-ROM nonlinear term is defined as $D^r(\overline{\boldsymbol{u}^r}) \cdot \nabla \boldsymbol{u}^r$. Since the AD operator $D^r(\overline{\boldsymbol{u}^r})$ provides a stable and accurate approximation of $\boldsymbol{u}^r$, its inclusion in the ADL-ROM improves the accuracy of the nonlinear term while preserving the enhanced stability introduced by the ROM filtering of $\overline{\boldsymbol{u}^r}$. As a result, the ADL-ROM strikes a balance between the baseline G-ROM and the overdissipative L-ROM, offering a compromise that combines accuracy and stability.

In this chapter, we guide the reader through the development of the ADL-ROM, elucidating its construction and practical application. We begin in Section 3.1 by introducing three distinct AD strategies: the van Cittert, Tikhonov, and Lavrentiev AD operators. Next, in Section 3.2, we detail the formulation of the ADL-ROM itself. Finally, in Section 3.3, we conduct a series of numerical investigations to evaluate the performance of the AD operators and the ADL-ROM. Specifically, we compare the three AD operators and assess the ADL-ROM in relation to the classical L-ROM and the standard G-ROM in the context of convection-dominated flow past a backward-facing step.

Some of the content in this chapter is derived from our foundational work on the ADL-ROM, as published in [86].

3.1 Approximate Deconvolution Operators

To develop the Approximate Deconvolution Leray ROM (ADL-ROM), it is essential to define and understand the ROM Approximate Deconvolution (AD) operators. The AD operators play a crucial role in this context by transforming an overly smoothed solution component into a more refined and accurate approximation of the desired solution. This approach ensures better accuracy and stability in numerical computations.

The concept of approximate deconvolution is not new and has its roots in various applications such as image processing and inverse problems. Over the years, it has been effectively extended to the realm of ROM closures [98] and ROM stabilization [86]. These advancements highlight the versatility and significance of AD in computational modeling.

The primary objective here is to compute $\boldsymbol{u}^r_{AD} \approx (F^r)^{-1}\ \overline{\boldsymbol{u}^r}$ without explicitly relying on the inverse operator $(F^r)^{-1}$. Instead, we employ an AD operator to achieve this task. Direct application of the filter inversion would result in an algorithm with poor numerical conditioning and potential ill-posedness, as discussed in [16, 43, 97]. Moreover, even if numerical instability could be mitigated, applying both F^r and $(F^r)^{-1}$ would nullify the filtering effect, thereby defeating the purpose of the procedure.

Various AD operators are available for use in this context. In this thesis, we focus on three notable options:

- The van Cittert operator, which has been applied at the finite element level in [59];

- Two classical regularization methods, both widely utilized in image processing and large eddy simulation [16, 62], and in modern regularization approaches, such as those discussed in [13]. These two regularization method are
  - the Tikhonov regularization;
  - the Lavrentiev regularization.

3.1.1 The Van Cittert AD

The van Cittert operator is an AD operator that iteratively refines an overdiffusive solution. Below, we define its continuous and ROM versions.

Definition 3.1.1 (Continuous van Cittert operator)**.** The continuous van Cittert operator $D_N : C^0(\Omega;\ \mathbb{R}^d) \to C^0(\Omega;\ \mathbb{R}^d)$ is defined as the map

$$\mathbf{\Phi} \mapsto D_N \mathbf{\Phi} \doteq \sum_{n=0}^{N} (I - F)^n \mathbf{\Phi}, \tag{3.1.1}$$

where $N \in \mathbb{N}$ represents the order of the operator.

Definition 3.1.2 (ROM van Cittert operator)**.** The ROM van Cittert operator $D_N^r : L^2(\Omega;\ \mathbb{R}^d) \to \boldsymbol{X}^r$ is defined as the map

$$\mathbf{\Phi} \mapsto D_N^r \mathbf{\Phi} \doteq \sum_{n=0}^{N} (I - F^r)^n\ \mathbf{\Phi}, \tag{3.1.2}$$

where $N \in \mathbb{N}$ is the order of the operator.

Although the ROM van Cittert operator in (3.1.2) can be applied to any function $\boldsymbol{u} \in L^2(\Omega;\ \mathbb{R}^d)$, in practice, it is specifically used for ROM filtered velocity fields $\overline{\boldsymbol{u}^r}$. In other words, for a positive integer N, the ROM AD velocity is computed as

$$\boldsymbol{u}^r_{AD} = D_N^r \overline{\boldsymbol{u}^r} = \sum_{n=0}^{N} (I - F^r)^n\ \overline{\boldsymbol{u}^r}, \tag{3.1.3}$$

following a similar formulation to the FEM AD velocity in [62, Section 3.1.1].

To implement this operator, we use a Richardson iteration process [62, Section 3.3.4] defined as

$$\boldsymbol{u}_{AD}^{r,(n+1)} = \boldsymbol{u}_{AD}^{r,(n)} + \{\overline{\boldsymbol{u}^r} - F^r\ \boldsymbol{u}_{AD}^{r,(n)}\}, \qquad n = 0, \dots, N-1, \tag{3.1.4}$$

where the initial value $\boldsymbol{u}_{AD}^{r,(0)}$ is set as $\overline{\boldsymbol{u}^r}$. By running the iteration (3.1.4) N times, we obtain $\boldsymbol{u}_{AD}^r \doteq \boldsymbol{u}_{AD}^{r,(N)}$ as the N^{th} order van Cittert operator applied to $\overline{\boldsymbol{u}^r}$. However, to make this iteration practical, we need to recall that each application of the filter F^r requires the inversion of a differential operator. Therefore, denoting by $\widetilde{\boldsymbol{u}^{r,(n+1)}}$ the quantity $\widetilde{\boldsymbol{u}^{r,(n+1)}} \doteq F^r \boldsymbol{u}_{AD}^{r,(n)}$, and substituting in our filter from (2.1.8), we observe that $\widetilde{\boldsymbol{u}^{r,(n+1)}}$ is found by solving for $n = 0, \ldots, N-1$ the following problem

$$(I - \delta^2 \Delta)^{-1} \boldsymbol{u}_{AD}^{r,(n)} = \widetilde{\boldsymbol{u}^{r,(n+1)}} \Longleftrightarrow (I - \delta^2 \Delta) \widetilde{\boldsymbol{u}^{r,(n+1)}} = \boldsymbol{u}_{AD}^{r,(n)}. \tag{3.1.5}$$

To employ the van Cittert AD in a ROM framework, we project this linear system onto the ROM space. By multiplying (3.1.5) by ROM test functions and expanding the solution using ROM basis functions, we derive the system

$$\left(M^r + \delta^2 S^r\right) \widetilde{\boldsymbol{a}^{(n+1)}} = M^r \boldsymbol{a}_{AD}^{(n)}, \tag{3.1.6}$$

where $M^r \in \mathbb{R}^{r \times r}$ is the POD mass matrix (which is the identity matrix due to the orthonormality of POD basis functions), $S^r \in \mathbb{R}^{r \times r}$ is the POD stiffness matrix, $\widetilde{\boldsymbol{a}^{(n+1)}} \in \mathbb{R}^r$ represents the POD coefficients of the output variable $\widetilde{\boldsymbol{u}^{r,(n+1)}}$, and $\boldsymbol{a}_{AD}^{(n)} \in \mathbb{R}^r$ denotes the POD coefficients of the input AD variable $\boldsymbol{u}_{AD}^{r,(n)}$.

The iterative process (3.1.4) is then as follows:

1. Initialize $\boldsymbol{a}_{AD}^{(0)} = \overline{\boldsymbol{a}}$;

2. Update the coefficients iteratively as

$$\boldsymbol{a}_{AD}^{(n+1)} = \boldsymbol{a}_{AD}^{(n)} + \{\overline{\boldsymbol{a}} - \widetilde{\boldsymbol{a}^{(n+1)}}\}, \qquad n = 0, \ldots, N-1;$$

3. In the end, define $\boldsymbol{a}_{AD} \doteq \boldsymbol{a}_{AD}^{(N)}$ at the completion of the N-th iteration.

3.1.2 The Tikhonov AD

The Tikhonov operator is an AD operator that refines an overdiffusive solution by solving of a linear system. Below, we define its continuous and ROM versions.

Definition 3.1.3 (Continuous Tikhonov operator)**.** The continuous Tikhonov operator $D_{T,\mu} : C^0(\Omega;\ \mathbb{R}^d) \to C^0(\Omega;\ \mathbb{R}^d)$ is defined as the map

$$\boldsymbol{\Phi} \mapsto D_{T,\mu} \boldsymbol{\Phi} \doteq (F^* F + \mu I)^{-1} F^* \boldsymbol{\Phi}, \tag{3.1.7}$$

where F^* denotes the adjoint of the operator F, and $\mu \in \mathbb{R}^+$ is a positive constant.

Definition 3.1.4 (ROM Tikhonov operator)**.** The ROM Tikhonov operator $D^r_{T,\mu} : L^2(\Omega;\ \mathbb{R}^d) \to \boldsymbol{X}^r$ is defined as the map

$$\boldsymbol{\Phi} \mapsto D^r_{T,\mu}\boldsymbol{\Phi} \doteq \left[(F^r)^* F^r + \mu I\right]^{-1} (F^r)^*\ \boldsymbol{\Phi}. \tag{3.1.8}$$

where $(F^r)^*$ denotes the adjoint of the operator F^r, and $\mu \in \mathbb{R}^+$ is a positive constant.

Although the ROM Tikhonov operator described in (3.1.8) applies to any function $\boldsymbol{u} \in L^2(\Omega;\ \mathbb{R}^d)$, in practice, it is applied to ROM filtered velocity fields $\overline{\boldsymbol{u}^r}$. In other words, the Tikhonov method is defined as

$$\boldsymbol{u}^r_{AD} = D^r_{,T\mu}\overline{\boldsymbol{u}^r} = \left[(F^r)^* F^r + \mu I\right]^{-1} (F^r)^*\overline{\boldsymbol{u}^r}. \tag{3.1.9}$$

We refer, e.g., to [62, section 3.3.1] for further details on the Tikhonov AD operator. When plugging in the specific filter from (2.1.8) and proceeding formally, we can write

$$\begin{aligned}
\left[(F^r)^* F^r + \mu I\right] \boldsymbol{u}^r_{AD} &= (F^r)^*\overline{\boldsymbol{u}^r}, \\
\left[(I - \delta^2\Delta)^{-*}(I - \delta^2\Delta)^{-1} + \mu I\right] \boldsymbol{u}^r_{AD} &= (I - \delta^2\Delta)^{-*}\overline{\boldsymbol{u}^r}, \\
\left[(I - \delta^2\Delta)^{-1} + \mu(I - \delta^2\Delta)^*\right] \boldsymbol{u}^r_{AD} &= \overline{\boldsymbol{u}^r}, \\
\left[I + \mu(I - \delta^2\Delta)(I - \delta^2\Delta)^*\right] \boldsymbol{u}^r_{AD} &= (I - \delta^2\Delta)\overline{\boldsymbol{u}^r}, \\
\left[I + \mu(I - \delta^2\Delta^* - \delta^2\Delta + \delta^2\Delta\delta^2\Delta^*)\right] \boldsymbol{u}^r_{AD} &= (I - \delta^2\Delta)\overline{\boldsymbol{u}^r}, \\
\left[I + \mu(I - 2\delta^2\Delta + \delta^4\Delta^2)\right] \boldsymbol{u}^r_{AD} &= (I - \delta^2\Delta)\overline{\boldsymbol{u}^r}.
\end{aligned} \tag{3.1.10}$$

In (3.1.10), we have utilized the fact that the Laplacian operator is self-adjoint. This intrinsic property of the Laplacian simplifies many theoretical and computational considerations.

In the numerical comparison of the three Approximate Deconvolution (AD) approaches presented in Section 3.3.3, for the sake of simplicity and ease of implementation within our finite element (FE) framework, we have chosen to neglect the term $\delta^4\Delta^2$. This decision is grounded on both practical and feasibility considerations, as outlined below.

1. *Practical Reasons:*

    - When δ is small, the term δ^4 becomes negligible, rapidly approaching zero. This diminishes the overall contribution of the $\delta^4\Delta^2$ term to the solution.

- The Tikhonov regularization method is designed to handle general filter operators F^r. However, in our specific application, we are using a tailored filter. The Lavrentiev methods, discussed in Section 3.1.3, are better suited for self-adjoint and positive definite filters, such as the one described in (2.1.8). For a rigorous proof of the self-adjointness and positive definiteness of this filter within a Reduced Order Model (ROM) framework, we refer the reader to [98].

2. *Feasibility Considerations:*

- The calculation of fourth-order operators like Δ^2 poses significant challenges in a ROM context. The computational overhead associated with such calculations is non-trivial, particularly when dealing with large-scale systems.

Given these practical and feasibility constraints and since we will ultimately not be using the Tikhonov method in our full simulations, we have opted to omit the $\delta^4\Delta^2$ term. Although this simplification renders the formulation more manageable, it remains an open question whether including this term would lead to a substantial improvement, especially when considering filters other than the specific one employed in this study. Future research could focus on studying this aspect in greater detail.

To proceed, we expand our desired solution as linear combinations of the ROM basis functions. This allows us to reformulate the problem as a linear system for the vector of coefficients $\boldsymbol{a}_{AD}$, which satisfies the following equation

$$[(1+\mu)M^r + 2\mu\delta^2 S^r]\boldsymbol{a}_{AD} = (M^r + \delta^2 S^r)\overline{\boldsymbol{a}}, \tag{3.1.11}$$

where $M^r \in \mathbb{R}^{r\times r}$ is the POD mass matrix, $S^r \in \mathbb{R}^{r\times r}$ is the POD stiffness matrix, $\boldsymbol{a}_{AD} \in \mathbb{R}^r$ is the vector of POD coefficients of the output AD variable $\boldsymbol{u}^r_{AD}$, and $\overline{\boldsymbol{a}} \in \mathbb{R}^r$ is the vector of POD coefficients of the input filtered variable $\overline{\boldsymbol{u}^r}$.

By neglecting the Δ^2 term, the Tikhonov regularization linear system in (3.1.11) simplifies significantly, resulting in a formulation that closely resembles the Lavrentiev regularization system, with the primary distinction being the additional factor of 2.

3.1.3 The Lavrentiev AD

The Lavrentiev operator is an AD operator that refines an overdiffusive solution by solving of a linear system. Below, we define its continuous and ROM versions

Definition 3.1.5 (Continuous Lavrentiev operator)**.** The continuous Lavrentiev operator $D_\mu : C^0(\Omega;\ \mathbb{R}^d) \to C^0(\Omega;\ \mathbb{R}^d)$ is defined as the map

$$\mathbf{\Phi} \mapsto D_\mu \mathbf{\Phi} \doteq (F + \mu I)^{-1} \mathbf{\Phi}, \tag{3.1.12}$$

where $\mu \in \mathbb{R}^+$ is a positive constant.

Note that the Lavrentiev parameter μ helps balance the contributions from the filtering operator F and the identity operator I. The parameter μ plays a crucial role in determining the trade-off between fidelity to the original data and the smoothing effect introduced by the filter.

Definition 3.1.6 (ROM Lavrentiev operator)**.** The ROM Lavrentiev operator $D^r_{T,\mu} : L^2(\Omega;\ \mathbb{R}^d) \to \boldsymbol{X}^r$ is defined as the map

$$\mathbf{\Phi} \mapsto D^r_\mu \mathbf{\Phi} \doteq (F^r + \mu I)^{-1}\ \mathbf{\Phi}. \tag{3.1.13}$$

where $\mu \in \mathbb{R}^+$ is a positive constant.

The Lavrentiev method modifies the Tikhonov regularization technique to adapt it for self-adjoint, positive definite operators. This makes it particularly suitable for problems where these properties hold, such as those involving the filter described in (2.1.8). These properties of the filter in (2.1.8) are rigorously established in [98], demonstrating its theoretical soundness in a ROM context.

Although the ROM Lavrentiev operator described in (3.1.13) is mathematically applicable to any function $\boldsymbol{u} \in L^2(\Omega;\ \mathbb{R}^d)$, in practical scenarios, it is predominantly applied to ROM filtered velocity fields $\overline{\boldsymbol{u}^r}$. This ensures that the Lavrentiev method is computationally efficient while retaining the desired filtering properties. To elaborate, given $\mu \in \mathbb{R}^+$, the Lavrentiev AD velocity is given by

$$\boldsymbol{u}^r_{AD} = D^r_\mu \overline{\boldsymbol{u}} = (F^r + \mu I)^{-1} \overline{\boldsymbol{u}^r}. \tag{3.1.14}$$

We solve for $\boldsymbol{u}^r_{AD}$ by substituting F^r with its own definition as

$$\begin{aligned}
\boldsymbol{u}^r_{AD} &= \left[(I - \delta^2\Delta)^{-1} + \mu I\right]^{-1} \overline{\boldsymbol{u}^r}, \\
\left[(I - \delta^2\Delta)^{-1} + \mu I\right] \boldsymbol{u}^r_{AD} &= \overline{\boldsymbol{u}^r}, \\
\left[I + \mu(I - \delta^2\Delta)\right] \boldsymbol{u}^r_{AD} &= (I - \delta^2\Delta)\overline{\boldsymbol{u}^r}, \\
\left[(1 + \mu)I - \mu\delta^2\Delta\right] \boldsymbol{u}^r_{AD} &= (I - \delta^2\Delta)\overline{\boldsymbol{u}^r}.
\end{aligned}$$

Expanding our solution as a linear combination of the ROM basis functions, multiplying by a test function, and integrating by parts, we obtain the following linear system to be solved for the vector of coefficients $\boldsymbol{a}_{AD}$

$$\left[(1+\mu)M^r + \mu\delta^2 S^r\right] \boldsymbol{a}_{AD} = \left(M^r + \delta^2 S^r\right) \overline{\boldsymbol{a}}, \tag{3.1.15}$$

where $M^r \in \mathbb{R}^{r\times r}$ is the POD mass matrix, $S^r \in \mathbb{R}^{r\times r}$ is the POD stiffness matrix, $\boldsymbol{a}_{AD} \in \mathbb{R}^r$ is the vector of POD coefficients of the output AD variable $\boldsymbol{u}^r_{AD}$, and $\overline{\boldsymbol{a}} \in \mathbb{R}^r$ is the vector of POD coefficients of the input filtered variable $\overline{\boldsymbol{u}^r}$.

3.2 Approximate Deconvolution Leray Reduced Order Model (ADL-ROM)

The Leray ROM (L-ROM), introduced in Section 2.2, has been a powerful tool for the numerical simulation of convection-dominated flows. However, one of its main limitations is the potential for overdiffusiveness, especially when the filter radius is chosen to be too large [86]. Overdiffusiveness can lead to an overly smoothed solution that may not capture critical flow features accurately.

To mitigate this issue, we propose incorporating Approximate Deconvolution (AD) techniques into the L-ROM framework to enhance its accuracy while maintaining its stability. The AD approach builds on the same basic differential filtering concept used in L-ROM, which ensures a straightforward implementation from existing L-ROM codes. This leads to the formulation of a new regularized ROM, termed the *Approximate Deconvolution Leray ROM (ADL-ROM)*. The ADL-ROM is inspired by earlier works in the finite element setting by Adrian Dunca [59, 61, 62], which demonstrated the potential of AD techniques to improve numerical accuracy in similar contexts.

The core idea of the ADL-ROM is simple yet effective: replace the filtered L-ROM velocity, $\overline{\boldsymbol{u}^r}$, with an approximate deconvolution velocity, $D^r(\overline{\boldsymbol{u}^r})$, where D^r is one of the AD operators defined in Section 3.1. This substitution refines the nonlinear term in the L-ROM, improving accuracy without compromising stability. Numerical experiments, presented in Section 3.3.3, demonstrate that computationally this modification enhances the representation of convective effects while retaining the robustness of the original L-ROM. Furthermore, theoretical results in Sections 4.1 and 4.2 establish error bounds for the AD operators and the ADL-ROM, validating its mathematical foundations.

Thus, the ADL-ROM strikes a balance between the numerically unstable G-ROM and the potentially overdiffusive L-ROM, offering an effective compromise for convection-dominated flow simulations.

3.2.1 ADL-ROM Time Discretization

Using the same time stepping scheme[1] as in the FOM and other ROMs, specifically the backward differentiation formula of second order (BDF2), the ADL-ROM time discretization is expressed as follows: Given $\boldsymbol{u}_k^r$ and $\boldsymbol{u}_{k-1}^r$, find $\boldsymbol{u}_{k+1}^r$ such that

$$\left(\frac{\boldsymbol{u}_{k+1}^r - \frac{4}{3}\boldsymbol{u}_k^r + \frac{1}{3}\boldsymbol{u}_{k-1}^r}{\Delta t}, \boldsymbol{\varphi}_i\right) + \frac{2}{3}\nu\left(\nabla \boldsymbol{u}_{k+1}^r, \nabla \boldsymbol{\varphi}_i\right) + \frac{2}{3}\left(\left(D^r(\overline{\boldsymbol{u}_{k+1}^r}) \cdot \nabla\right)\boldsymbol{u}_{k+1}^r, \boldsymbol{\varphi}_i\right) = \left(\boldsymbol{f}_{k+1}, \boldsymbol{\varphi}_i\right), \tag{3.2.1}$$

for all $k = 1, \ldots, K-1$ and $i = 1, \ldots, r$, where Δt is the time step and $D^r(\cdot)$ is one of the AD operators defined in Section 3.1.

3.2.2 Definition of AD Convective Term

The AD convective term in (3.2.1) is defined using the ROM differential filter (2.1.8) as follows:

$$\boldsymbol{u}_{AD,k+1}^r(\boldsymbol{x}, t) \doteq \mathrm{D}^r(\overline{\boldsymbol{u}_{k+1}^r}) \doteq \sum_{j=1}^{r} a_{AD,k+1,j}(t)\ \boldsymbol{\varphi}_j(\boldsymbol{x}). \tag{3.2.2}$$

The coefficients $\boldsymbol{a}_{AD,k+1}$ in (3.2.2) are obtained by solving a reduced linear system, which is derived from either (3.1.6), (3.1.11), or (3.1.15), using the current approximation of the filtered coefficients $\overline{\boldsymbol{a}}_{k+1}$. These coefficients, in turn, are computed by solving the linear system (2.2.3).

At each time step t^{k+1}, the ADL-ROM discretization leads to the following nonlinear system:

$$\frac{1}{\Delta t}M^r\left(\boldsymbol{a}^{k+1} - \frac{4}{3}\boldsymbol{a}^k + \frac{1}{3}\boldsymbol{a}^{k-1}\right) + \frac{2}{3}\nu S^r \boldsymbol{a}^{k+1} + \frac{2}{3}C(\boldsymbol{a}_{AD,k+1})\boldsymbol{a}^{k+1} = \boldsymbol{b}_{k+1}, \tag{3.2.3}$$

where $\boldsymbol{a}_{k+1}$ represents the unknown reduced coefficient vectors of the unfiltered velocity field, and $\boldsymbol{a}_{AD,k+1}$ corresponds to the coefficients of the AD ROM velocity field. These coefficients are derived from (2.2.3) and one of the approximate deconvolution linear systems. The vector $\boldsymbol{b}_{k+1}$ is defined as in (1.3.13), and the operator $C(\overline{\boldsymbol{a}}_{k+1})$ is given by

$$C(\boldsymbol{a}_{AD,k+1})_{ij} = \left((\mathrm{D}^r(\overline{\boldsymbol{u}_{k+1}^r}) \cdot \nabla)\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_j\right). \tag{3.2.4}$$

[1]The time discretization scheme can be safely replaced with another scheme.

The nonlinear system (3.2.3) is solved using the Newton method. Each Newton iteration involves solving two additional reduced $r \times r$ linear systems to compute the filtered and deconvolved coefficients. These operations introduce minimal computational overhead due to the low dimensionality of the systems.

3.2.3 ADL-ROM Implementation

The pseudocode for the ADL-ROM implementation is summarized in Algorithm 3.

Algorithm 3 ADL-ROM Pseudocode

$\boldsymbol{u}_{-1}$, $\boldsymbol{u}_0$, $\boldsymbol{u}_{in}$, r ▷ Inputs needed
for $k = 1, \dots, K-1$ **do** ▷ Time loop
    Perform FOM simulation to compute $\boldsymbol{u}^h_{k+1}$
end for
Compute $\boldsymbol{X}^r \doteq \text{POD}\left(\{\boldsymbol{u}^h_k\}_{k=1}^K; r\right)$ ▷ POD basis generation
for $k = 1, \dots, K-1$ **do** ▷ Time loop for ROM
    Solve (3.2.3) to compute $\boldsymbol{a}^{k+1}$, which requires:
        Solve (2.2.3) at each nonlinear iteration ▷ Leray filtering
        Solve one of (3.1.6), (3.1.11), or (3.1.15)
         at each nonlinear iteration ▷ Approximate Deconvolution
end for

This algorithm efficiently combines approximate deconvolution with the Leray ROM framework, offering a robust and accurate reduced order modeling approach for convection-dominated flows.

3.3 Numerical Results

In this section, we aim to present and analyze a series of computational results that demonstrate the effectiveness of the proposed methods. Specifically, we provide an extensive set of numerical simulations to investigate the behavior and performance of both the AD operators and the ADL-ROM itself. As said before, some of the results included in Section 3.3.3 have already been published in our first work on ADL-ROM [86]. We omit the results on Burgers' equations since they have been produced by Ian Moore, while the NSE results have been completely produced by this thesis author.

In Section 3.3.3, we conduct a comprehensive comparison involving the three distinct AD operators introduced in Section 3.1. Additionally, we compare the three ROM approaches explored in this thesis: G-ROM (introduced in Section 1.3), L-ROM (detailed in Section 2.2), and ADL-ROM (presented in Section 3.2). It is important to note that we utilize only the Lavrentiev and van Cittert AD operators to construct the ADL-ROM in this work.

Furthermore, all the numerical implementations in this work are carried out using the RBniCSx software. To offer context and insight into this tool, Section 3.3.1 includes a concise introduction to RBniCSx, see [79] for further details.

3.3.1 RBniCSx

RBniCSx represents the latest evolution of the RBniCS software [78], specifically designed for reduced order modeling. The primary distinction between RBniCS and RBniCSx lies in their foundational frameworks: while RBniCS is built on FEniCS, RBniCSx is developed on FEniCSx. A key enhancement in FEniCSx, and consequently RBniCSx, is its support for computations involving complex numbers, which is not available in the earlier version.

In our implementation, RBniCSx was chosen for two major reasons. Firstly, this software is purposefully developed for constructing reduced order models, making it exceptionally convenient for researchers. It offers an extensive amount of prebuilt functions that not only streamline the theoretical steps of implementation but also enhance users' understanding of how these functions operate. Secondly, as a FEniCSx-based platform, RBniCSx integrates several advanced reduced order modeling techniques, including the Certified Reduced Basis (CRB) method and Proper Orthogonal Decomposition-Galerkin (POD-Galerkin) methods. The truth solution, or high-fidelity approximation, is computed using a high-order finite element solver provided by FEniCSx.

Currently, RBniCS and RBniCSx are actively developed and maintained at SISSA mathLab under the guidance of Prof. Francesco Ballarin and supervised by Prof. Gianluigi Rozza as part of the AROMA-CFD ERC CoG project. For further details, readers are encouraged to visit the RBniCS [78] and RBniCSx [79] websites.

FEniCSx

The FEniCS Project [25] is an influential research and software initiative dedicated to the development of mathematical methods and software for automating computational mathematical modeling. Its central aim is to create user-friendly, efficient, and adaptable software for solving partial differential equations (PDEs) using finite element methods.

Launched in 2003 as a collaborative effort between the University of Chicago and Chalmers University of Technology, the project quickly garnered global participation. Notable contributors include researchers from institutions such as the University of Cambridge, Royal Institute of Technology, University of Luxembourg, and Argonne National Laboratory. Over the years, FEniCS has seen significant updates and enhancements, including the expansion of online tutorials and the release of new versions, making FEniCS a remarkably stable software. This process culminates in the development of FEniCSx. This latest version of FEniCS incorporates support for computations over the complex field. For additional information, refer to the official FEniCSx website [25].

FEniCSx comprises a suite of free and open-source software (FOSS) components, ensuring universal accessibility. These components facilitate work with computational meshes, finite-element variational formulations of differential equations, and numerical linear algebra. Key building blocks of FEniCSx include DOLFINx [11], FFCx [54], UFL [3], Basix [88, 87], among others. In order to have an overview of FEniCSx, the reader may refer to [64]. To aid users in optimizing their experience with FEniCSx, we provide a brief overview of its primary components:

- **UFL** implements the abstract mathematical language that allows users to express variational problems intuitively.

- **FFCx** is the fundamental code generation engine of FEniCSx (the form compiler), responsible for generating efficient C++ code from high-level mathematical abstractions.

- **DOLFINx** is the computational high-performance C++ backend of FEniCSx. Moreover, it implements data structures such as meshes, function spaces and functions, computationally intensive algorithms such as finite element assembly and mesh refinement, and interfaces to linear algebra solvers and data structures such as PETSc[2]. DOLFINx also implements the FEniCSx problem-solving environment in both C++ and Python.

- **Basix** is a finite element definition and tabulation runtime library. It is part of FEniCSx, alongside UFL, FFCx and DOLFINx (C++ docs, Python docs). Basix can create finite elements on intervals, triangles,

[2]The Portable, Extensible Toolkit for Scientific computation (PETSc, pronounced PET-see) is a suite of open source software libraries for the parallel solution of linear and nonlinear algebraic equations. PETSc uses the Message Passing Interface (MPI) for all its parallelism.

quadrilaterals, tetrahedra, hexahedra, prisms, and pyramids. FEniCSx supports all the principal types of finite elements and the notation to use them is the one defined in *Periodic Table of Finite Elements*[3] [4].

3.3.2 Flow Past a Cylinder

Goals This section presents the first set of numerical tests conducted as part of this study. The primary objective of these experiments was to verify whether employing the AD operators produces results that are intermediary between the reduced solution and the filtered reduced solution. Essentially, we aimed to validate our initial hypothesis. To achieve this, we started by selecting a well-known domain: the 2D flow past a cylinder. Using this domain, we computed both the Full Order Model (FOM) solution and the Galerkin Reduced Order Model (G-ROM) solution, as detailed in the subsequent sections.

Instead of directly solving the Leray Reduced Order Model (L-ROM) or the Approximate Deconvolution Leray Reduced Order Model (ADL-ROM), we opted to apply the ROM differential filter (ROM-DF) outlined in (2.1.8) and one of the AD operators described in Section 3.1 to the G-ROM solution during post-processing.

For consistency with the only previous instance where AD methods were applied in ROMs [98] and for reasons elaborated in Section 3.3.3, this section focuses on the Lavrentiev AD method presented in Section 3.1.3. Consequently, any references to ADL-ROM in our numerical results should be understood as referring specifically to ADL-ROMs constructed using the Lavrentiev regularization method.

Thus, the overarching goals are to validate our conceptual approach to AD operators (specifically the Lavrentiev AD operator), demonstrate why exact deconvolution is impractical, and investigate how to tune the Lavrentiev parameter μ.

Computational setting We consider the motion of an incompressible flow within the computational domain $\Omega \doteq \{[0, 2.2] \times [0, 0.41]\} \setminus \{(x, y) \in \mathbb{R}^2 : (x - 0.2)^2 + (y - 0.2)^2 - 0.05^2 = 0\}$, illustrated in Figure 3.1. The kinematic viscosity is set to 10^{-3}, and no-slip boundary conditions are applied to $\partial\Omega_D^{\text{wall}} \doteq \partial\Omega_{\text{w}}$, which includes the bottom, cylinder, and top walls of the channel (indicated by the solid blue boundary in Figure 3.1). The inlet boundary $\partial\Omega_D^{in}$ (dashed red line in Figure 3.1) features a time-varying velocity profile

[3] https://www.femtable.org.

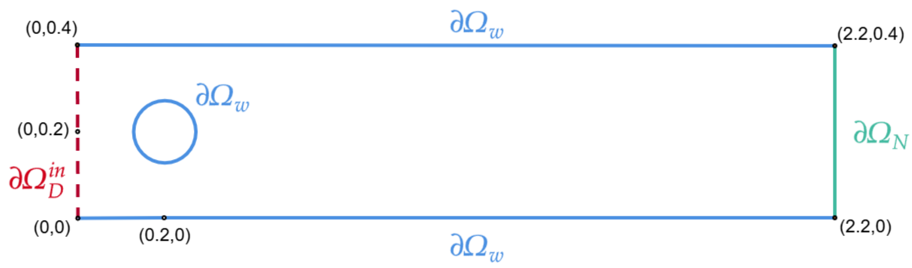

Figure 3.1: The computational domain, Ω. $\partial\Omega_D \doteq \partial\Omega_D^{in} \cup \partial\Omega_D^{\text{wall}}$, where the inlet boundary $\partial\Omega_D^{in}$ is represented by a dashed red line and the no-slip boundaries by a solid blue line.

$\boldsymbol{u}_{in}$, prescribed as follows

$$\boldsymbol{u}_{in} \doteq \left(\frac{0.6}{0.41^2} \sin\left(\frac{\pi t}{8}\right) y(0.41 - y), 0 \right), \tag{3.3.1}$$

for $t \leq 4$, and $\boldsymbol{u}_{in} = (1, 0)$ otherwise. Homogeneous Neumann boundary conditions are applied to $\partial\Omega_N$ (denoted by the solid teal line in Figure 3.1).

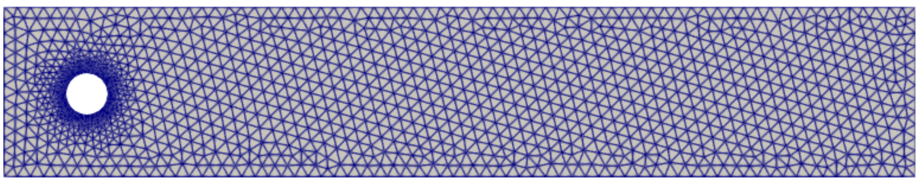

Figure 3.2: The FE mesh.

Snapshot generation The numerical tests are performed on a triangular mesh with minimum element size of $h_{\min} = 0.004$ and maximum element size of $h_{\max} = 0.032$ (see Figure 3.2). Spatial discretization is achieved using the inf-sup stable Taylor-Hood $\mathbb{P}^2 - \mathbb{P}^1$ finite element pair for velocity and pressure, respectively. This discretization results in a finite element space with a total dimension of $D \doteq D_h \doteq D_h^u + D_h^p = 12488 + 1621 = 14109$. A second-order backward differentiation formula (BDF2) scheme is employed for temporal discretization with a time step size of $\Delta t = 10^{-3}$ for both FOM and G-ROM simulations. The FOM is run over the time interval $[T_0, T] = [0, 20]$, with initial conditions $\boldsymbol{u}_{-1} = \boldsymbol{u}_0 = (\mathbf{0}, \mathbf{0})$.

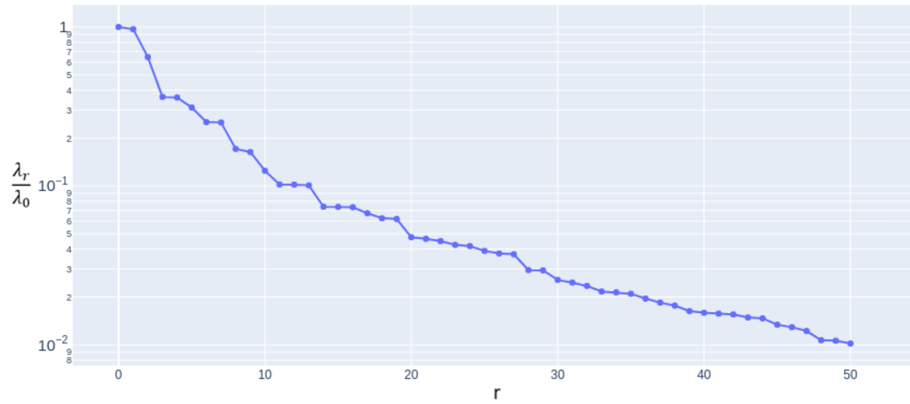

Figure 3.3: Normalized singular values from POD procedure with $r_{max} = 50$.

ROM construction To construct the Galerkin reduced order model (G-ROM) basis functions, 400 snapshots of the velocity field are collected over the whole time interval $[0, 20]$. The first $r = 50$ POD modes capture approximately 98% of the flow's energy, as shown in Figure 3.3. We deliberately restrict the number of POD basis functions to this value to highlight the effects of regularization in scenarios where the ROM cannot fully resolve all scales of motion, that is a common occurrence in practical applications. These modes are utilized to construct the G-ROM, which is analyzed over the entire time interval $[0, 20]$.

Parameters The inlet velocity, the dimensions of the domain and the viscosity value, i.e., $\nu = 10^{-3}$, lead to a Reynolds number $Re = 1000$.

One of the most important parameters of ROM-DF and AD operators is the radius of the ROM differential filter introduced in Section 2.1.1, δ. To illustrate the case of an aggressive choice of the L-ROM filter radius, in our numerical investigation we used $\delta = 0.1$, as it is significantly larger than $h_{\max}$.

The other parameter of interest is the Lavrentiev regularization parameter, μ. In this case, by keeping the value of δ fixed, we chose to use three different values of μ:

- $\mu = 0$, corresponding to perform an exact deconvolution instead of an approximate one;
- $\mu = 0.03$, corresponding to give more importance to the original data, i.e., the G-ROM solution;

- $\mu = 0.3$, corresponding to give more importance to the filter data, i.e., the filtered G-ROM solution.

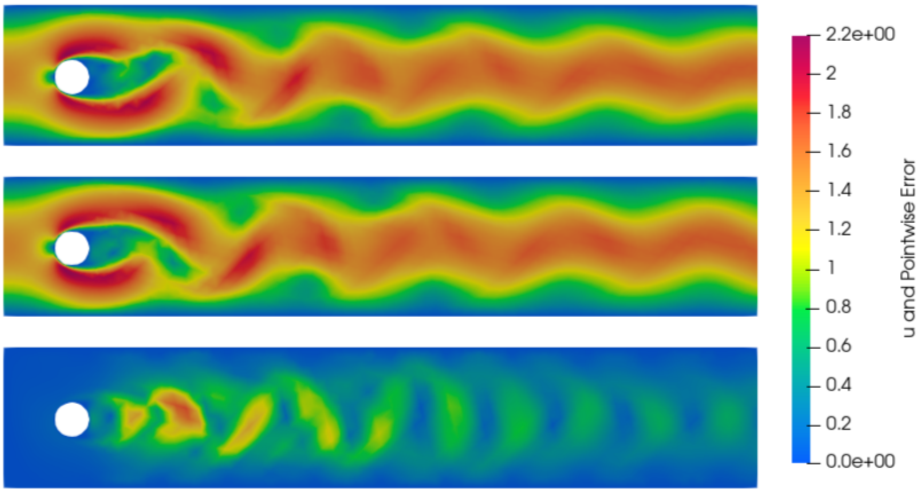

Figure 3.4: At $t = 10$, 1st: FOM solution $\boldsymbol{u}^h$, 2nd: G-ROM solution $\boldsymbol{u}^r$ for $r = 50$, 3rd: absolute pointwise error between the solutions $\boldsymbol{u}^h$ and $\boldsymbol{u}^r$.

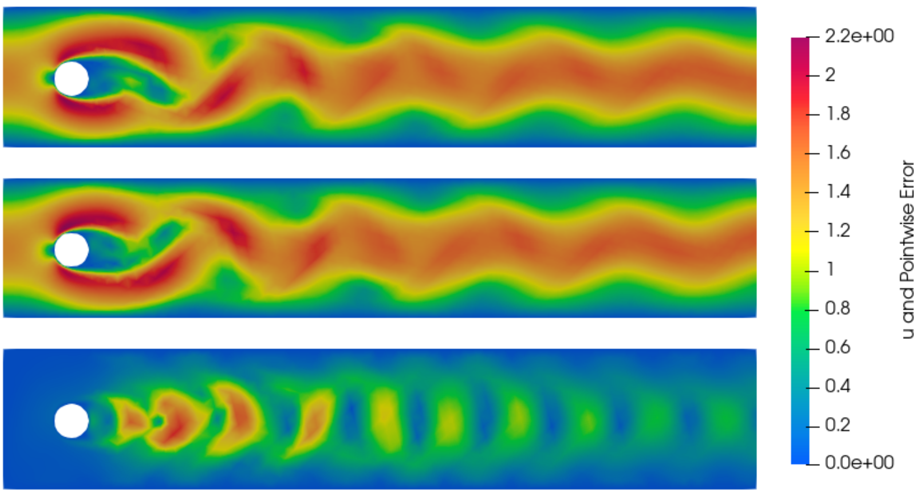

Figure 3.5: At the final time $T = 20$, 1st: FOM solution $\boldsymbol{u}^h$, 2nd: G-ROM solution $\boldsymbol{u}^r$ for $r = 50$, 3rd: absolute pointwise error between the solutions $\boldsymbol{u}^h$ and $\boldsymbol{u}^r$.

Numerical results First of all, we present the representative FOM and G-ROM solutions of the velocity field at $t = 10$ and at the final time $T = 20$

in Figures 3.4 and 3.5, respectively. The plots reveal that G-ROM is not able to perfectly reconstruct the solution provided by FOM, even if a quite large number of basis functions is used, i.e., $r = 50$.

As additional proof, we exhibit the plot of absolute pointwise error functions, i.e., the pointwise difference between the FOM and the G-ROM solutions, depiscted in the last row of Figures 3.4 and 3.5. The previously observed difference in the accuracy of the G-ROM solution is even clearer in this plot. Indeed, the maximum pointwise error value in both plots further accentuates the limited accuracy of the G-ROM solution in reconstructing the FOM results.

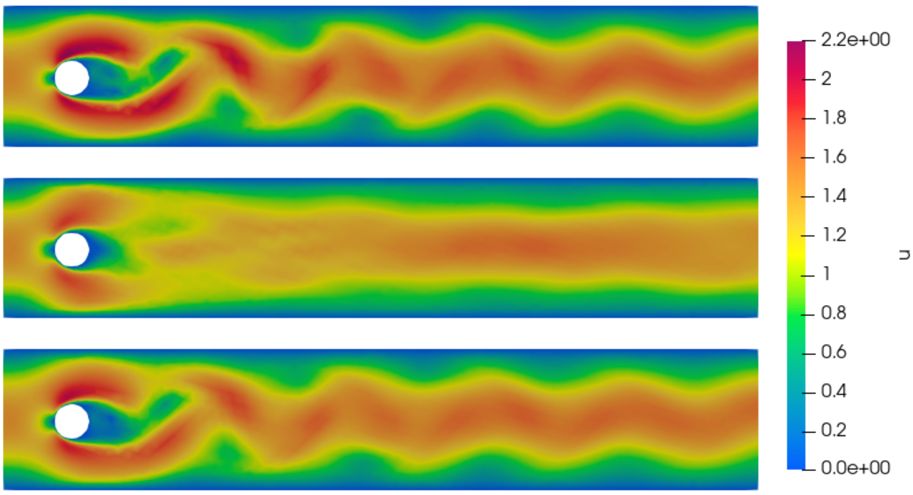

Figure 3.6: At the final time $T = 20$ for $r = 50$, 1st: G-ROM solution $\boldsymbol{u}^r$, 2nd: filtered G-ROM solution $F^r(\boldsymbol{u}^r)$ for $\delta = 0.1$, 3rd: AD filtered G-ROM solution $D^r_\mu\left(F^r(\boldsymbol{u}^r)\right)$ for $\delta = 0.1$ and $\mu = 0.03$.

Next, Figure 3.6 compares the performance of the G-ROM solution $\boldsymbol{u}^r$, the filtered G-ROM solution $F^r(\boldsymbol{u}^r)$, and the Approximate Deconvolution (AD) filtered G-ROM solution $D^r_\mu\left(F^r(\boldsymbol{u}^r)\right)$ for $r = 50$, $\delta = 0.1$, and $\mu = 0.03$ at the final time $T = 20$. The results demonstrate that the filtered G-ROM solution is severely overdiffusive. In contrast, the AD filtered G-ROM solution achieves a balance between the solutions $\boldsymbol{u}^r$ and $F^r(\boldsymbol{u}^r)$, preserving the primary flow characteristics by mitigating excessive diffusion. These observations confirm our initial hypothesis.

For our third and final test in this section, we want to show how, with some adjustments to the value of μ, the AD reduced solution can mitigate the noise introduced by the filtered reduced solution. Furthermore, we provide some practical evidences to demonstrate why exact deconvolution is

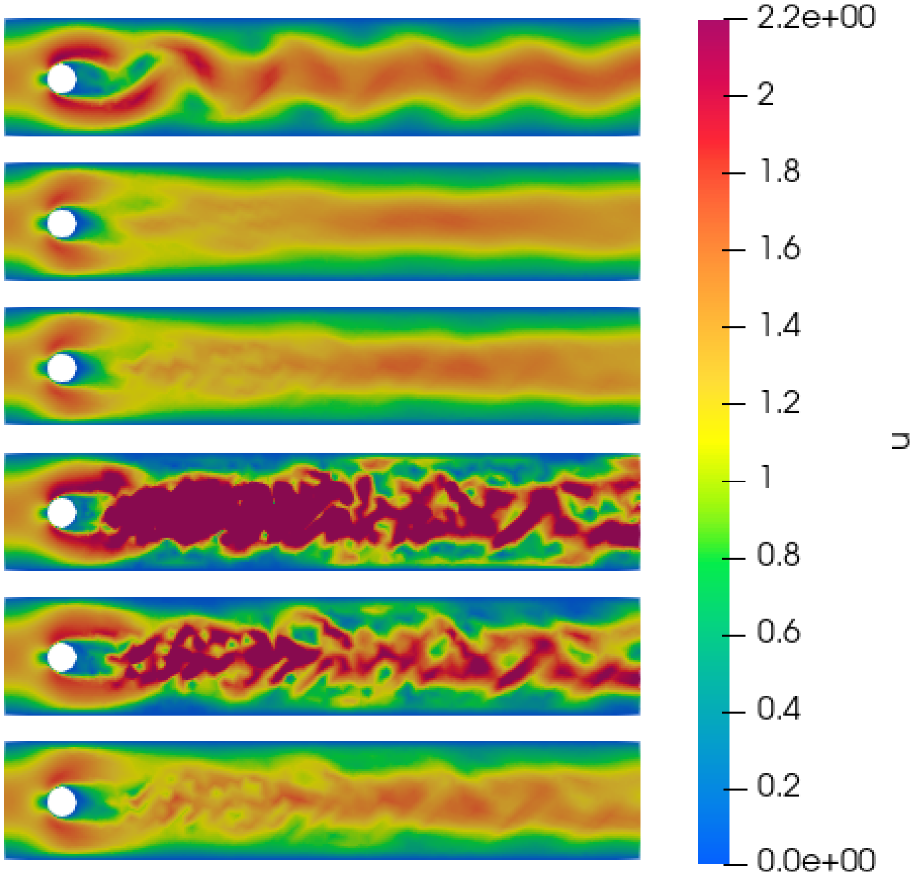

Figure 3.7: At the final time $T = 20$ for $r = 50$ 1st: $\boldsymbol{u}^r$, 2nd: $F^r(\boldsymbol{u}^r)$ for $\delta = 0.1$, 3rd: little disturbed $F^r(\boldsymbol{u}^r)$, i.e., $\mathcal{F}^r(\boldsymbol{u}^r)$, for $\delta = 0.1$, 4th: $D^r_\mu\left(\mathcal{F}^r(\boldsymbol{u}^r)\right)$ for $\delta = 0.1$ and $\mu = 0.0$, 5th: $D^r_\mu\left(\mathcal{F}^r(\boldsymbol{u}^r)\right)$ for $\delta = 0.1$ and $\mu = 0.03$, 6th: $D^r_\mu\left(\mathcal{F}^r(\boldsymbol{u}^r)\right)$ for $\delta = 0.1$ and $\mu = 0.3$.

ineffective since it amplifies numerical errors. The steps of this test are as follows:

1. filter the G-ROM solution $\boldsymbol{u}^r$ and obtain $F^r(\boldsymbol{u}^r)$;
2. add some noises to $F^r(\boldsymbol{u}^r)$ and obtain the disturbed filtered reduced solution $\mathcal{F}^r(\boldsymbol{u}^r)$;
3. apply the AD operator D^r_μ to $\mathcal{F}^r(\boldsymbol{u}^r)$ for different values of the Lavrentiev regularization parameter μ.

As said previously, we set the filter radius δ to 0.1 in order to have a highly diffusive filtered reduced solution $F^r(\boldsymbol{u}^r)$ and make more challenging to AD operator restore the flow behavior.

For the results shown in Figure 3.7, we added to $F^r(\boldsymbol{u}^r)$ random noise with a magnitude between -10^{-2} and 10^{-2} to each component. As shown by the plots in Figure 3.7, at the final time the noise is visible in the third row. This causes the complete failure of the exact deconvolution reduced solution (fourth row), but also not so great results for the AD reduced solutions both with $\mu = 0.03$ (fifth row) and with $\mu = 0.3$ (sixth row).

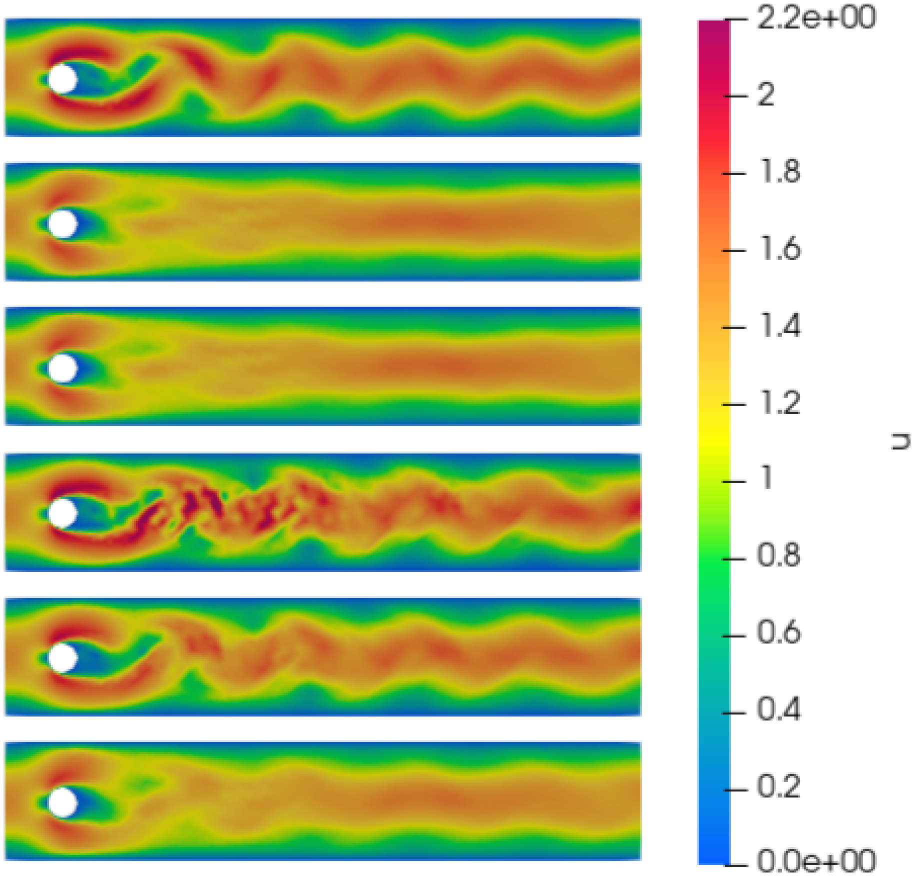

Figure 3.8: At the final time $T = 20$ for $r = 50$ 1st: $\boldsymbol{u}^r$, 2nd: $F^r(\boldsymbol{u}^r)$ for $\delta = 0.1$, 3rd: very little disturbed $F^r(\boldsymbol{u}^r)$, i.e., $\mathcal{F}^r(\boldsymbol{u}^r)$, for $\delta = 0.1$, 4th: $D^r_\mu\left(\mathcal{F}^r(\boldsymbol{u}^r)\right)$ for $\delta = 0.1$ and $\mu = 0.0$, 5th: $D^r_\mu\left(\mathcal{F}^r(\boldsymbol{u}^r)\right)$ for $\delta = 0.1$ and $\mu = 0.03$, 6th: $D^r_\mu\left(\mathcal{F}^r(\boldsymbol{u}^r)\right)$ for $\delta = 0.1$ and $\mu = 0.3$.

For the results shown in Figure 3.8, we added to $F^r(\boldsymbol{u}^r)$ random noise with a magnitude between -10^{-3} and 10^{-3} to each component. This time, as shown by the plots in Figure 3.8, at the final time the noise is not visually detectable in the third row. Even in this case, the exact deconvolution approach (fourth row) still leads to an amplification of the error, but the approximate deconvolution both with $\mu = 0.03$ (fifth row) and with $\mu = 0.3$ (sixth row) behaves a lot better. In particular, for $\mu = 0.03$ the AD reduced solution $D^r_\mu\left(\mathscr{F}^r(\boldsymbol{u}^r)\right)$ is very similar to the G-ROM solution $\boldsymbol{u}^r$, albeit slightly too chaotic. Instead, for $\mu = 0.3$ the AD reduced solution is too diffusive as the original filtered reduced solution $F^r(\boldsymbol{u}^r)$. This suggests that an optimal μ could lay between $\mu = 0.03$ and $\mu = 0.3$.

All the results of this section corroborate our expectations regarding the AD operators, particularly the Lavrentiev AD operator, demonstrating its efficacy in addressing the challenges of overcoming noises and overdiffusion.

3.3.3 Backward Facing Step

Goals In this section, we undertake a detailed analysis and comparison of the performance of the three previously introduces reduced order modeling approaches: the Galerkin Reduced Order Model (G-ROM) described by (1.3.22), the Leray Reduced Order Model (L-ROM) given by (2.2.1), and the Approximate Deconvolution Leray Reduced Order Model (ADL-ROM) outlined in (3.2.1). These methods are applied to the Navier-Stokes Equations (NSE) (1.1.1)-(1.1.2) to evaluate their effectiveness under the chosen test conditions.

The primary objective of this investigation is to assess the accuracy and reliability of the ADL-ROM approach in capturing the essential dynamics of the flow. To this end, we focus on a benchmark case of 2D incompressible fluid flow over a backward-facing step. The flow is characterized by a Reynolds number of $Re = 1429$. This test case serves as an ideal scenario for measuring the capabilities of ADL-ROM, particularly in terms of its ability to approximate key flow characteristics with reduced computational cost.

By systematically comparing the results of G-ROM, L-ROM, and ADL-ROM, we aim to highlight the advantages of ADL-ROM, especially in handling the challenges posed by this type of flow configuration. The analysis includes both quantitative metrics and qualitative assessments to provide a comprehensive understanding of ADL-ROM performance.

Computational setting We consider the motion of an incompressible flow in the same domain as that used in [6, section 4.4] and in [69, section 3.5], i.e, $\Omega \doteq \{[0, 44] \times [0, 9]\}\backslash\{[0, 4] \times [0, 1]\}$, which is depicted in Figure 3.9. We set a kinematic viscosity of 7×10^{-4} and use no-slip boundary conditions on $\partial\Omega_D^{\text{wall}} \doteq \partial\Omega_{\text{B}} \cup \partial\Omega_{\text{T}}$, representing the union of the bottom $\partial\Omega_{\text{B}}$ and top walls $\partial\Omega_{\text{T}}$ of the channel (solid blue boundary in Figure 3.9), with a

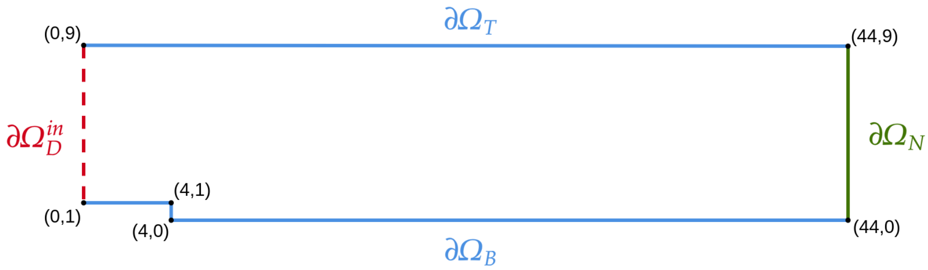

Figure 3.9: The computational domain, Ω. $\partial\Omega_D \doteq \partial\Omega_D^{in} \cup \partial\Omega_D^{\text{wall}}$, where the inlet boundary $\partial\Omega_D^{in}$ is represented by a dashed red line and the no-slip boundaries by a solid blue line.

constant inlet velocity profile $\boldsymbol{u}_{in} = (1, 0)$ on $\partial\Omega_D^{in}$ (red dashed line in Figure 3.9). Furthermore, we employ homogeneous Neumann boundary conditions on $\partial\Omega_N$ (green line in Figure 3.9).

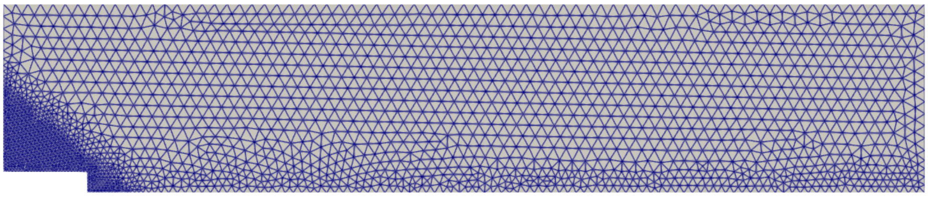

Figure 3.10: The FE mesh.

Snapshot generation We perform our tests on a triangular mesh with $h_{\text{min}} = 0.15$ and $h_{\text{max}} = 0.6$ (Figure 3.10). For the spatial discretization, we employ the inf-sup stable Taylor-Hood $\mathbb{P}^2 - \mathbb{P}^1$ FE pair for velocity and pressure, respectively, and this leads to a FE space of dimension $D \doteq D_h \doteq D_h^u + D_h^p = 18404 + 2370 = 20774$. A second-order backward differences (BDF2) scheme has been used with the time step $\Delta t = 5 \times 10^{-2}$ for both FOM and ROM time discretizations. The time interval on which FOM is performed is $[T_0, T] = [0, 150]$. The value of the initial conditions $\boldsymbol{u}_{-1}$ and $\boldsymbol{u}_0$ is $(\mathbf{0}, \mathbf{0})$.

At the FOM level, since we work in the under-resolved regime, we apply a regularization strategy. Specifically, we employ the evolve-filter-relax (EFR) strategy (for details, see [91]). The EFR stabilization strategy allows us to obtain accurate results in the convection-dominated regime on the coarse mesh in Figure 3.10. We use EFR as a stabilization strategy for the following reasons: EFR has been widely used at the FOM level, is easy to implement, is effective, and has been used successfully in our numerical investigations [40, 67, 70, 91, 95, 96]. We emphasize, however, that other regularization strategies (e.g., Leray or even ADL, which would ensure FOM-ROM consistency [91]) or stabilization approaches could be investigated. This could be subject of future work. In Figure 3.11, we plot the time evolution of the FOM kinetic energy on the time interval $[100, 150]$. This plot shows that the flow over Ω is not periodic or periodic-like.

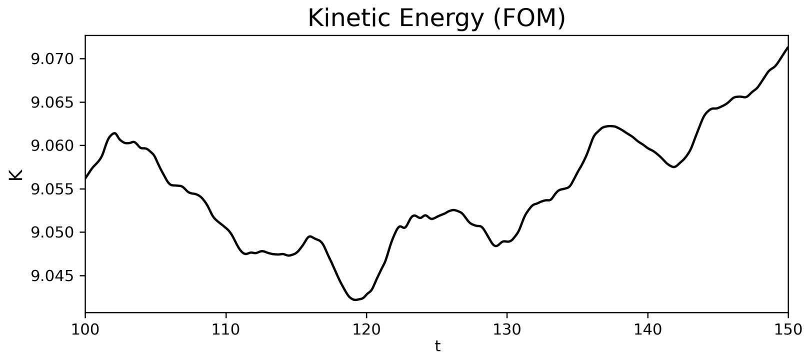

Figure 3.11: Time evolution of FOM kinetic energy.

ROM construction To build the ROM basis functions, we collect 1000 snapshots of the velocity field over the time interval $[100.05, 150]$, as in [69] and [81]. The first $r = 10$ POD modes capture about 80% of the flow's energy, as it is shown from Figure 3.12. We limit ourselves to a small number of POD basis functions for the purpose of showing the effect of regularization when the ROM cannot accurately describe all scales of motion, which is the case in many realistic applications. These modes are used to construct all ROMs. All the ROMs are investigated on the reduced time interval $[100.05, 150]$.

AD Comparison Before proceeding to the discussion of ADL-ROM results, we report here on the choice of AD to use given the options discussed in Section 3.1. To do this, we will briefly examine the effects that these AD methods have on a standard G-ROM solution. The G-ROM solution will be calculated following the methods discussed in Section 1.3 and in the previous paragraph. In this case, we use the time interval $[0, 20]$ with a fixed time step $\Delta t = 10^{-2}$.

In Figures 3.13 and 3.14, we plot the pointwise difference between the computed G-ROM solution at the final time $T = 20$ and each of the AD methods applied to the G-ROM solution. Figure 3.13 also shows that, with

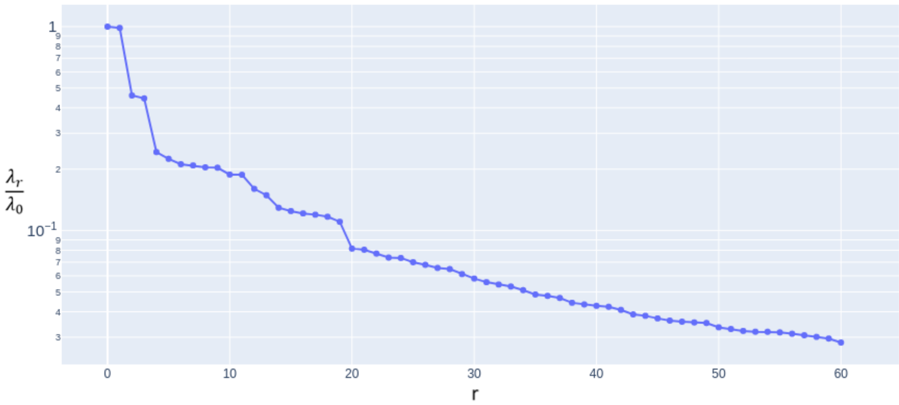

Figure 3.12: Normalized singular values from POD procedure with $r_{max} = 60$.

Figure 3.13: Pointwise difference between AD applied to G-ROM and G-ROM, $r = 10$, $\delta = 0.6$ and . 1^{st}: Lavrentiev with $\mu = 0.1$; 2^{nd}: Tikhonov with $\mu = 0.1$; 3^{rd}: van Cittert with $N = 10$.

AD parameters chosen as labeled, the Lavrentiev AD method produces results which are the least similar to the Tikhonov or van Cittert methods. However, this is mostly a matter of tuning: Figure 3.14 shows that altering the AD parameters of μ for the Tikhonov or N for the van Cittert method will make those methods produce very similar results as to the Lavrentiev method in the first plot of Figure 3.13.

Each of the AD methods shown in Figures 3.13 and 3.14 affects the computed G-ROM solution most strongly near the backward facing step, which

Figure 3.14: Pointwise difference between AD applied to G-ROM and G-ROM, $r = 10$, $\delta = 0.6$. 1^{st}: Tikhonov with $\mu = 0.05$. 2^{nd}: van Cittert with $N = 12$.

is the primary place where vortices are formed and is where we want the smoothing to be focused. Additionally, by varying either δ in the original filter or one of the AD parameters μ or N as in Figure 3.14, each of the AD regularization methods can produce similar results to the other methods. For these reasons, the numerical results of this chapter focus only on the Lavrentiev and the van Cittert AD methods described in Section 3.1.3 and 3.1.1.

Parameters As mentioned before, the inlet velocity and the size of the step lead to a Reynolds number $Re = 1429$. For the EFR FOM stabilization strategy, there are two parameters: the filtering radius of the differential filter, δ_{EFR}, employed in the *Filter* step, and the relaxation parameter, $\chi \in [0, 1]$, employed in the *Relax* step. For the rest of the chapter, we present results for $\delta_{EFR} = 0.01$ and $\chi = 0.01$. This choice limits the amount of dissipation introduced by the differential filter in the EFR algorithm.

One of the most important parameters of L-ROM and ADL-ROM is the radius of the ROM differential filter introduced in Section 2.1.1, δ. To avoid confusion between the filtering radius of the EFR step δ_{EFR}, and the filtering radius of the ROM-DF, in the discussion we will denote by δ_{ROM} the radius of the ROM-DF, instead of δ. To illustrate the case of an aggressive choice of the L-ROM filter radius, in our numerical investigation we used $\delta_{ROM} = 0.6$, that is the same value as $h_{\max}$.

Two other important ADL-ROM parameters are

- The Lavrentiev regularization parameter, μ. In the image processing and inverse problem communities (see [16, 43]), numerous approaches are proposed to determine the regularization parameters. In our numerical investigation, for the fixed $\delta_{ROM} = 0.6$ value, we choose the μ value that ensures that the ADL-ROM solution is as close as possible to the FOM solution. This approach yields the value $\mu = 0.06$.

- The van Cittert order, N. Also in this case, in our numerical investigation, for the fixed $\delta_{ROM} = 0.6$ value, we choose the N value that ensures that the ADL-ROM solution is as close as possible to the FOM solution. This approach yields the value $N = 8$. We will give more explanation on this decision later on.

Criteria We test the ROMs accuracy by using the velocity field L^2 absolute and relative errors, and the kinetic energy. The errors are defined as

$$E_{\boldsymbol{u}}^a(t) \doteq \left\|\boldsymbol{u}(t) - \boldsymbol{u}_r(t)\right\|_{L^2(\Omega)} \quad \text{and} \quad E_{\boldsymbol{u}}^r(t) \doteq \frac{E_{\boldsymbol{u}}^a(t)}{\|\boldsymbol{u}(t)\|_{L^2(\Omega)}}, \tag{3.3.2}$$

while the kinetic energy is defined as

$$K_{\boldsymbol{u}}(t) \doteq \frac{1}{2}\|\boldsymbol{u}(t)\|^2_{L^2(\Omega)}. \tag{3.3.3}$$

To compare the ROMs' performance, we also use the following error metric

$$\text{time-average } \; L^2 \; \text{ norm}: \quad \mathcal{E}^{a/r}(L^2) \doteq \frac{1}{K}\sum_{j=1}^{K} E_{\boldsymbol{u}}^{a/r}(t_j), \tag{3.3.4}$$

where the a and r superscripts refer to the absolute and relative errors, respectively.

Numerical results In Table 3.1, we list the average absolute and relative L^2 errors (3.3.4) for the G-ROM, L-ROM, and ADL-ROM. In particular, for ADL-ROM we use both Lavrentiev AD with $\mu = 0.06$ and van Cittert AD with $N = 5, 8, 10$. These results show that both the G-ROM and the ADL-ROM are more accurate than the L-ROM, with the G-ROM being about 30% more accurate than the L-ROM, and the ADL-ROM about 60% more accurate than the L-ROM. Furthermore, the ADL-ROM is consistently more accurate than the G-ROM with a difference of about 30%.

Remark 3.3.1. We have to emphasize that L-ROM works worse than G-ROM and ADL-ROM since we chose a quite large value of δ_{ROM}, i.e., $\delta_{ROM} = 0.6$. This ensures that the L-ROM solutions are overdissipative and inaccurate. Anyway, we chose to use this value since lower values of δ_{ROM} are less

Average L^2 errors			
Model	μ/N	$\mathscr{E}^a(L^2)$	$\mathscr{E}^r(L^2)$
G-ROM	–	1.2668	7.25e-02
L-ROM	–	1.6061	8.88e-02
Lavrentiev ADL-ROM	0.06	1.0144	5.60e-02
van Cittert ADL-ROM	5	1.0135	5.60e-02
van Cittert ADL-ROM	8	9.45e-01	5.23e-02
van Cittert ADL-ROM	10	9.76e-01	5.40e-02

Table 3.1: Average absolute and relative L^2 errors for G-ROM, L-ROM, and ADL-ROM with two different AD operators and different N values.

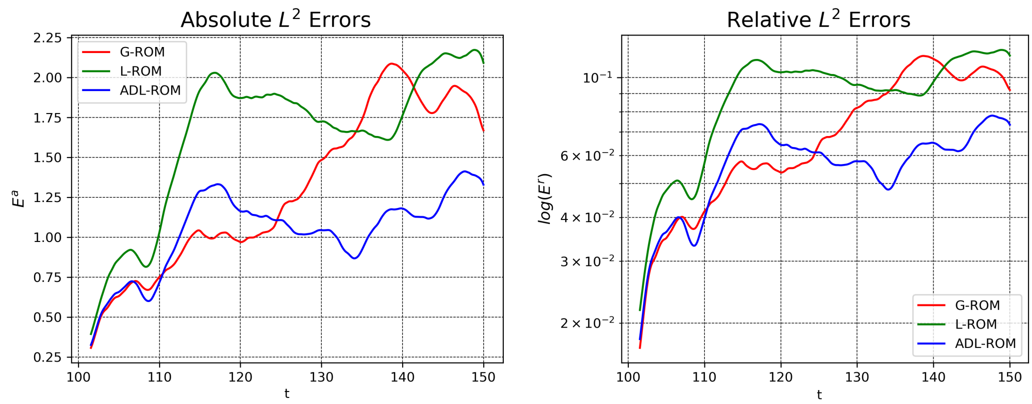

Figure 3.15: Time evolution of the absolute and relative L^2 errors for G-ROM (red), L-ROM (green), and Lavrentiev ADL-ROM (blue).

challenging for ADL-ROM. Moreover, from our tests, L-ROM seems to be more sensitive to parameter choice than ADL-ROM and this is another reason why it is easier to choose effective parameter values and get good results for ADL-ROM than for L-ROM. ♠

Moreover, the results in Table 3.1 for the van Cittert operator show that the average absolute and relative L^2 errors are the smallest ones for $N = 8$. This supports our decision to show the van Cittert AD solutions only for $N = 8$.

Also the temporal trend of the absolute and relative L^2 errors (3.3.2), displayed in Figures 3.15 and 3.16, confirms this conclusion. It shows that both Lavrentiev and van Cittert ADL-ROM always performs better than L-ROM and starts to perform better than G-ROM after a while. On the other

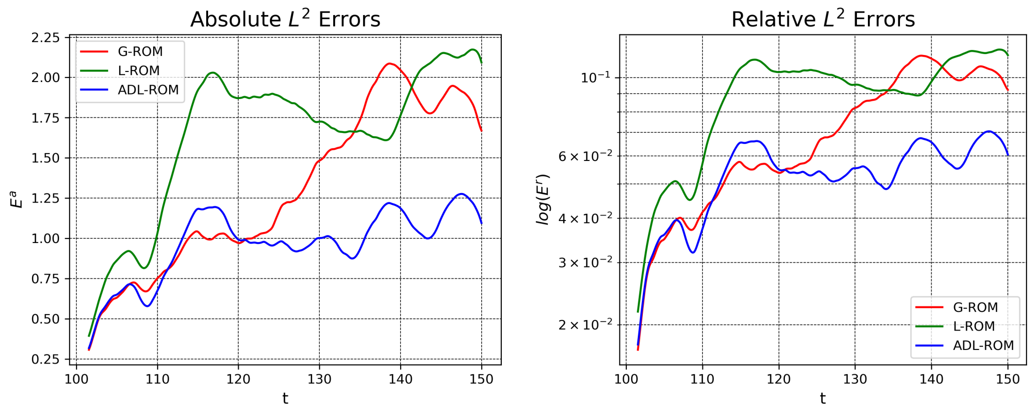

Figure 3.16: Time evolution of the absolute and relative L^2 errors for G-ROM (red), L-ROM (green), and van Cittert ADL-ROM (blue).

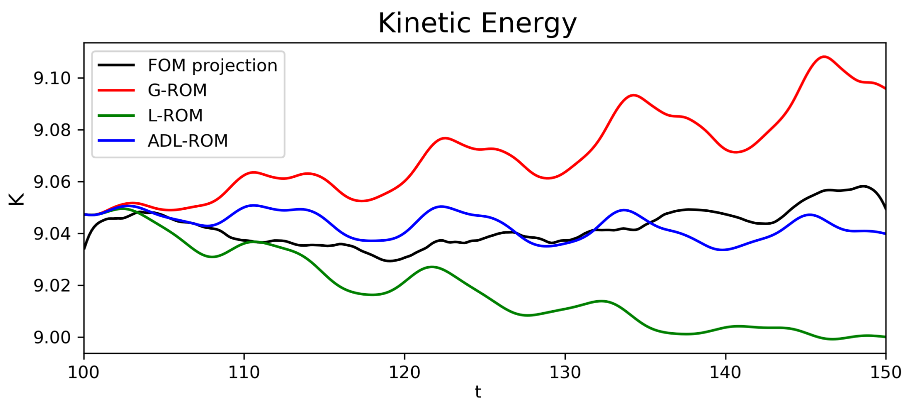

Figure 3.17: Time evolution of the kinetic energy for FOM projection (black), G-ROM (red), L-ROM (green), and Lavrentiev ADL-ROM (blue).

hand, G-ROM performs better than L-ROM except for the time interval $[134, 142]$. In Figures 3.17 and 3.18, we plot the time evolution of the kinetic energy (3.3.3) of the FOM projection, G-ROM, L-ROM, and Lavrentiev and van Cittert ADL-ROM, respectively. These plots support the conclusions in Table 3.1 and in Figure 3.15. Specifically, the G-ROM and L-ROM results are relatively inaccurate, while the ADL–ROM results are significantly more accurate.

As last test to compare the consistency of the van Cittert ADL-ROM in respect to the filter radius δ and its order N , we compute the average kinetic

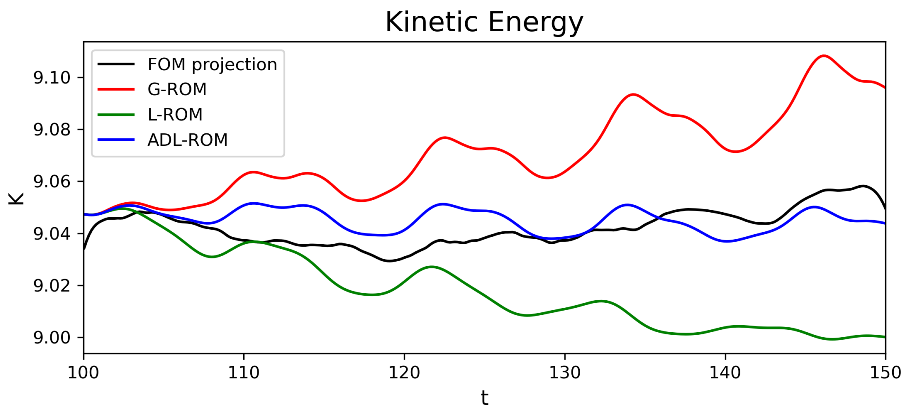

Figure 3.18: Time evolution of the kinetic energy for FOM projection (black), G-ROM (red), L-ROM (green), and van Cittert ADL-ROM (blue).

Average L^2 kinetic energy				
	L-ROM	ADL-ROM		
δ	$\mathcal{K}(L^2)$	$\mathcal{K}(L^2),\ N=5$	$\mathcal{K}(L^2),\ N=10$	$\mathcal{K}(L^2),\ N=15$
0.6	9.0182	9.0362	9.0498	9.0580
0.3	9.0268	9.0626	9.0689	9.0700
0.1	9.0549	9.0703	9.0703	9.0703
0.05	9.0655	9.0703	9.0703	9.0703
0.01	9.0701	9.0703	9.0703	9.0703

Table 3.2: L-ROM and van Cittert ADL-ROM average kinetic energies for various filter radii (δ) and van Cittert orders (N). FOM $\mathcal{K}(L^2)$ is 9.0417, G-ROM $\mathcal{K}(L^2)$ is 9.0703.

energy for t in [100.05,150] as

$$\mathcal{K}(L^2) \doteq \frac{1}{K}\sum_{j=1}^{K} K_{\boldsymbol{u}}(t_j). \tag{3.3.5}$$

We use this quantity to show that the van Cittert ADL-ROM produces results which are more consistent across various ranges of δ and across the AD parameter N. Indeed, in Table 3.2, we plot $\mathcal{K}(L^2)$, the average kinetic energy for t in [100.05,150], in which the ADL-ROM results are more similar to each other than the L-ROM results. If the ADL-ROM is a good model for the problem at hand, then less tuning of the filter radius δ is required.

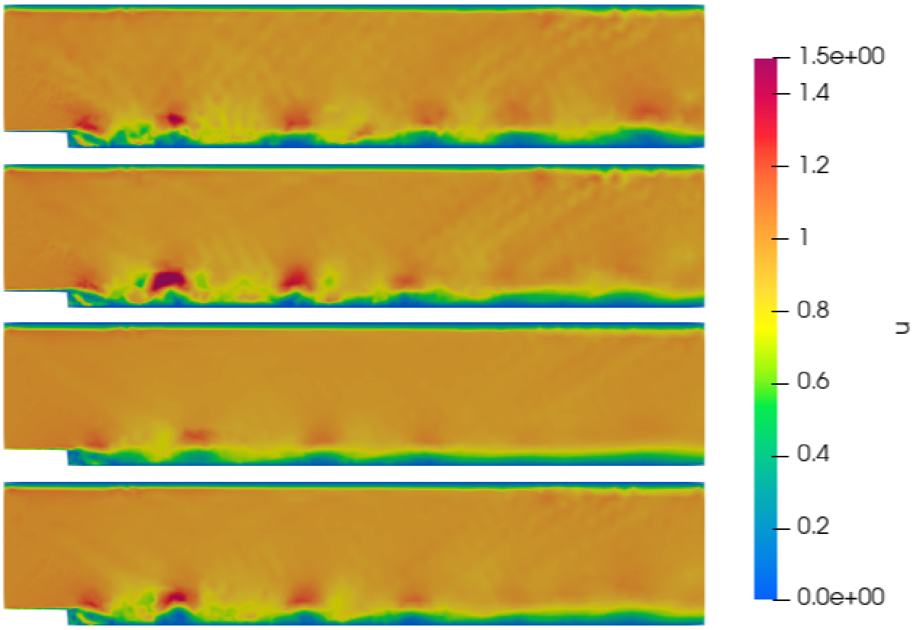

Figure 3.19: At the final time $T = 150$, 1^{st}: FOM solution; 2^{nd}: G-ROM solution; 3^{rd}: L-ROM solution; and 4^{th}: Lavrentiev ADL-ROM solution for $r = 10$ with $\delta = 0.6$ and $\mu = 0.06$.

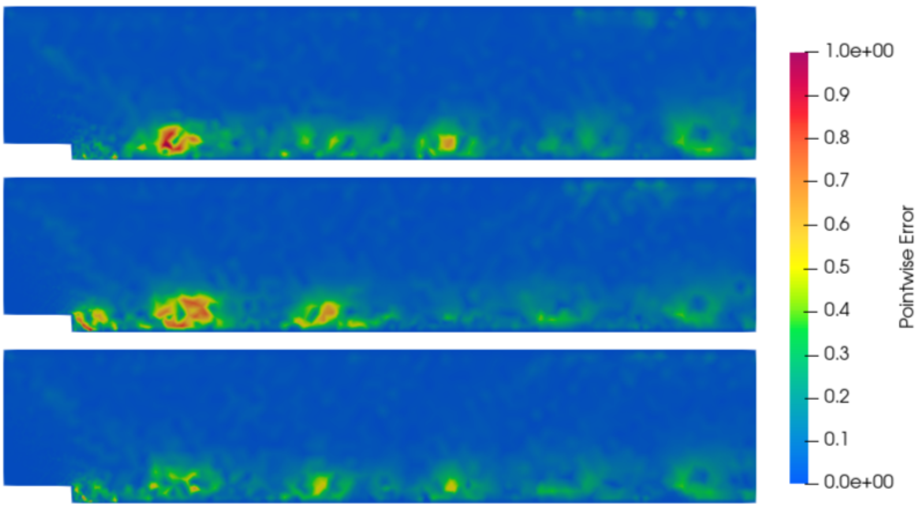

Figure 3.20: At the final time $T = 150$, 1^{st}: absolute pointwise error between FOM and G-ROM solutions; 2^{nd}: absolute pointwise error between FOM and L-ROM solutions; and 3^{rd}: absolute pointwise error between FOM and Lavrentiev ADL-ROM solutions for $r = 10$ with $\delta = 0.6$ and $\mu = 0.06$.

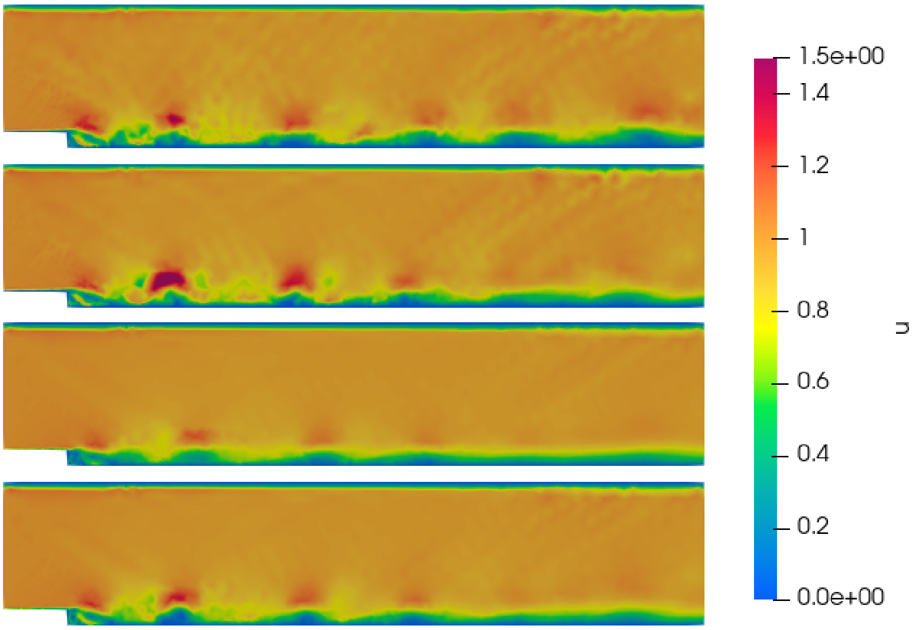

Figure 3.21: At the final time $T = 150$, 1^{st}: FOM solution; 2^{nd}: G-ROM solution; 3^{rd}: L-ROM solution; and 4^{th}: van Cittert ADL-ROM solution for $r = 10$ with $\delta = 0.6$ and $N = 8$.

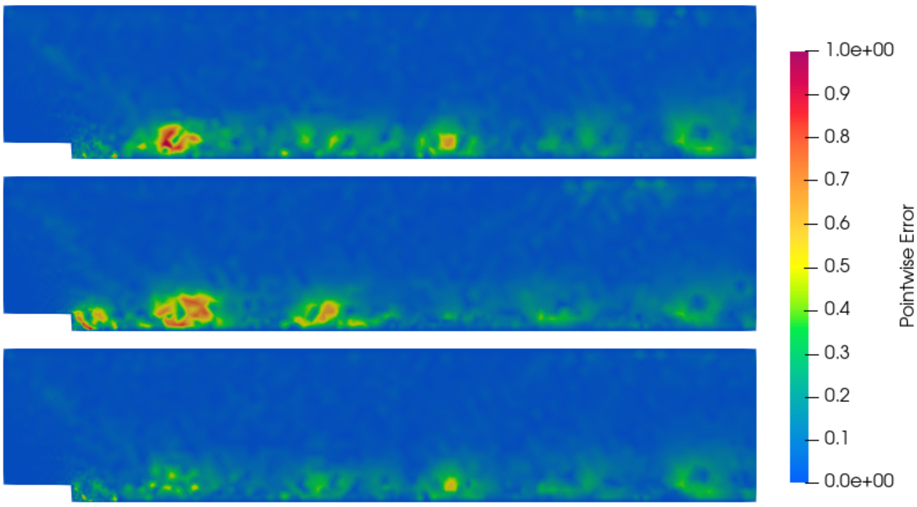

Figure 3.22: At the final time $T = 150$, 1^{st}: absolute pointwise error between FOM and G-ROM solutions; 2^{nd}: absolute pointwise error between FOM and L-ROM solutions; and 3^{rd}: absolute pointwise error between FOM and van Cittert ADL-ROM solutions for $r = 10$ with $\delta = 0.6$ and $\mu = 0.06$.

We show representative solutions of the velocity field for the final time $T = 150$ in Figures 3.19 and 3.21. It is clear that the L-ROM is not able to reconstruct the solution provided by the FOM, while the G-ROM leads to more accurate results and the ADL-ROM leads to the best results both when the Lavrentiev AD operator is used and when the van Cittert operator is used. As additional proof, we exhibit the plot of the absolute pointwise error functions, i.e., the pointwise difference between the FOM and the ROMs solutions in Figures 3.20 and 3.22. The previously observed difference in the accuracy of the ROMs solutions is even clearer in this plot by considering the maximum pointwise value in each of the plots.

The errors listed in Table 3.1 and all the plots show that both the G–ROM and the ADL–ROM are more accurate than the L-ROM. Furthermore, the ADL–ROM is more accurate than the G–ROM.

Computational performance The FE solution has 20774 degrees of freedom. The CPU time required by the 3000 time steps of the FE simulation is $9900\,s$, and the CPU time required by the last 1000 final time steps is $3000\,s$. The final 1000 time steps of the ROM solutions plotted in Figures 3.19 and 3.21 require $2.20\,s$ (G-ROM), $2.35\,s$ (L-ROM), $2.47\,s$ (Lavrentiev ADL-ROM), and $2.56\,s$ (van Cittert ADL-ROM). The mild increase in CPU time from G-ROM to L-ROM is due to line 8 in Algorithm 2, which is required only by L-ROM. Similarly, the mild increase in CPU time from L-ROM to the two ADL-ROMs is due to line 9 in Algorithm 3, which is required only by ADL-ROM. We emphasize, however, that all the ROMs yield a significant speedup: 1363.63 (G-ROM), 1276.59 (L-ROM), 1214.58 (Lavrentiev ADL-ROM), and 1171.88 (van Cittert ADL-ROM).

Chapter 4

Numerical Analysis

In general, reduced order model (ROM) stabilizations (or closures), including our newly proposed Reg-ROM, are typically assessed through heuristic approaches. Specifically, these models are implemented in numerical simulations, where their effectiveness is evaluated by demonstrating improvements in numerical accuracy compared to the standard Galerkin ROM (G-ROM) and/or other ROM stabilization techniques.

However, many fundamental questions in the numerical analysis of ROM stabilizations and closures remain largely unresolved. Issues such as stability, convergence, and the rate of convergence are still open problems for the majority of existing ROM closures and stabilization methods. While some initial steps have been taken to address these challenges, as highlighted in the numerical analysis of ROM closures [9, Section 4.6] and stabilizations [9, Section 4.5], these efforts are still in their beginning. In comparison, the numerical analysis of full order model (FOM) closures and stabilizations is significantly more developed, leaving a noticeable gap in understanding these topics at the ROM level.

In our second paper [68], we take a significant step toward addressing this gap. Specifically, we establish rigorous *a priori* error bounds for both the newly proposed ROM van Cittert approximate deconvolution operator and the novel Reg-ROM, namely the Approximate Deconvolution Leray ROM (ADL-ROM), which was first introduced in [86].

To the best of our knowledge, these results represent the first instance of numerical analysis for approximate deconvolution operators at the ROM level. The derived *a priori* error bounds provide a detailed characterization of how the ADL-ROM error depends on both the ROM parameters (e.g., the number of modes and filter parameters) and the FOM parameters (e.g., viscosity and mesh size).

In this chapter, we present a detailed exposition of our theoretical results, supported by lemmas, theorems, and their proofs. The structure of this chapter is as follows:

- In Section 4.1, we begin by introducing error bounds for the approximate deconvolution (AD) operator itself. For this analysis, we focus exclusively on the van Cittert regularization method.
- In Section 4.2, we present the main contributions of our work, namely rigorous numerical analysis results for the ADL-ROM. These results include proofs of stability and convergence, which constitute a significant advancement in the numerical analysis of ROM stabilizations.
- Finally, in Section 4.3, we provide numerical results that validate the theoretical findings. In particular, we illustrate the *a priori* error bounds derived in the previous sections through carefully designed numerical experiments.

It is worth noting that some of the theoretical and computational content presented in this chapter is derived from our initial work on the numerical analysis of the ADL-ROM, as in [68].

4.1 Numerical Analysis of AD Operators

In this section, we undertake a rigorous numerical analysis of the AD operators at the Reduced Order Model (ROM) level. To the best of our knowledge, this represents the first formal and detailed treatment of such stability properties. For related results in the context of the more thoroughly investigated finite element (FE) framework, we refer the reader to foundational works such as [59] and [23].

Lemma 4.1.1. *Let $\boldsymbol{u} \in \boldsymbol{X}$. Then*

$$\|D_N^r \overline{\boldsymbol{u}}\| \leq C(N) \|\boldsymbol{u}\| \tag{4.1.1}$$

$$\|\nabla D_N^r \overline{\boldsymbol{u}}\| \leq C(N) \sqrt{\|S^r\|_2}\, \|\boldsymbol{u}\|. \tag{4.1.2}$$

Furthermore, if $\boldsymbol{u} \in \boldsymbol{X}^r$, then

$$\|\nabla D_N^r \overline{\boldsymbol{u}}\| \leq C(N) \|\nabla \boldsymbol{u}\|. \tag{4.1.3}$$

Similarly to Lemma 2.1.6, the first bound (4.1.1) implies that the composition $\widetilde{D}_N^r \circ \widetilde{F}^r$ is a bounded operator on $\boldsymbol{X}$, where $\widetilde{D}_N^r \doteq D_N^r|_{\boldsymbol{X}^r}$ is the restriction of D_N^r to $\boldsymbol{X}^r$ and $\widetilde{F}^r \doteq F^r|_{\boldsymbol{X}^r}$ is the restriction of F^r to $\boldsymbol{X}^r$.

Proof. Since the uniform in N boundedness of $\widetilde{F}^r$ is guaranteed by (2.1.11), it is clear that to prove (4.1.1) we only need to prove that $\widetilde{D}^r_N$ is a bounded operator from $\boldsymbol{X}^r$ to itself. When $\boldsymbol{\Phi} \in \boldsymbol{X}^r$, F^r in (3.1.2) can be replaced by $\widetilde{F}^r$. Upon such replacement, the right-hand side of (3.1.2) contains linear combinations and compositions of the continuous operator $I - \widetilde{F}^r$, implying that $\widetilde{D}^r_N$ is continuous as well, with a continuity constant that depends on N.

To prove (4.1.2), we use the POD inverse inequality (1.3.3) and the bound in (4.1.1):

$$\|\nabla D^r_N \overline{\boldsymbol{u}}\| \overset{(1.3.3)}{\leq} \sqrt{\|S^r\|_2}\; \|D^r_N \overline{\boldsymbol{u}}\| \overset{(4.1.1)}{\leq} C(N)\sqrt{\|S^r\|_2}\; \|\boldsymbol{u}\|. \tag{4.1.4}$$

Upon recognizing that D^r_N can be replaced by $\widetilde{D}^r_N$ in (4.1.3), we finally prove (4.1.3) by induction. The base case $N = 0$ follows from (2.1.13), and results in $C(0) = 1$. For the induction step, assume that (4.1.3) holds for $N - 1$. We first rewrite $\widetilde{D}^r_N$ as

$$\begin{aligned}\widetilde{D}^r_N \boldsymbol{\Phi} &= \sum_{n=0}^{N} (I - \widetilde{F}^r)^n \boldsymbol{\Phi} = \boldsymbol{\Phi} + \sum_{n=1}^{N} (I - \widetilde{F}^r)^n \boldsymbol{\Phi} \\ &= \boldsymbol{\Phi} + (I - \widetilde{F}^r) \sum_{n=0}^{N-1} (I - \widetilde{F}^r)^n \boldsymbol{\Phi} = (I + (I - \widetilde{F}^r)\widetilde{D}^r_{N-1})\boldsymbol{\Phi} \end{aligned} \tag{4.1.5}$$

to obtain, in particular,

$$\widetilde{D}^r_N \overline{\boldsymbol{u}} = \overline{\boldsymbol{u}} + \widetilde{D}^r_{N-1}\overline{\boldsymbol{u}} - \overline{\widetilde{D}^r_{N-1}\overline{\boldsymbol{u}}},$$

and thus

$$\begin{aligned}\|\nabla \widetilde{D}^r_N \overline{\boldsymbol{u}}\| &\overset{\substack{\text{triangle}\\ \text{inequality}}}{\leq} \|\nabla \overline{\boldsymbol{u}}\| + \|\nabla \widetilde{D}^r_{N-1}\overline{\boldsymbol{u}}\| + \|\nabla \overline{\widetilde{D}^r_{N-1}\overline{\boldsymbol{u}}}\| \\ &\overset{(2.1.13)}{\leq} \|\nabla \boldsymbol{u}\| + 2\|\nabla \widetilde{D}^r_{N-1}\overline{\boldsymbol{u}}\| \overset{(4.1.3)\text{ for } N-1}{\leq} (1 + 2C(N-1))\; \|\nabla \boldsymbol{u}\|,\end{aligned}$$

which concludes the proof. □

Building on the results established in Lemma 2.1.5, we now present and prove the main ROM AD lemma of our work. This result is used to bound the AD term in our ADL-ROM scheme in (3.2.1). While this lemma is tailored specifically to the van Cittert regularization method, the approach can be adapted to other approximate deconvolution methods.

Lemma 4.1.2. *Let* $\boldsymbol{u}_k = \boldsymbol{u}(\boldsymbol{x}, t_k) \in \boldsymbol{X}$ *be a solution to* (1.2.1) *with* $\Delta \boldsymbol{u}_k \in L^2(\Omega)$. *If Assumption 1.3.1 holds, then, for all* k

$$\begin{aligned}
\|\boldsymbol{u}_k - D_N^r \overline{\boldsymbol{u}_k}\|^2 \leq C(N) &\left(h^{2m+4} + \Delta t^4 + \sum_{j=r+1}^{R} \lambda_j \right) \\
&+ C(N)\delta^2 \left(h^{2m+2} + \|S^r\|_2 \; h^{2m+4} + (1 + \|S^r\|_2)\Delta t^4 \right. \\
&\left. + \sum_{j=r+1}^{R} \|\nabla \boldsymbol{\varphi}_j\|^2 \lambda_j \right) + C(N)\delta^4 \|\Delta \boldsymbol{u}_k\|^2.
\end{aligned}$$

Proof. First, we recall that from (4.1.5) the computation of $D_N^r \overline{\boldsymbol{u}}$ can be equivalently carried out by means of the following iterative process

$$\begin{cases} \boldsymbol{u}^{(0)} = \overline{\boldsymbol{u}} \doteq \widetilde{F}^r \boldsymbol{u}, \\ \boldsymbol{u}^{(n+1)} = \widetilde{F}^r \boldsymbol{u} + (I - \widetilde{F}^r)\boldsymbol{u}^{(n)}, \quad n = 0, \dots, N-1. \end{cases} \tag{4.1.6}$$

We wish now to obtain a recurrence formula for the AD error

$$\boldsymbol{e}^{(N)} \doteq D_N^r \overline{\boldsymbol{u}} - \boldsymbol{u} = \boldsymbol{u}^{(N)} - \boldsymbol{u}.$$

To do so, replace $\widetilde{F}^r \boldsymbol{u}$ in (4.1.6) by exploiting the identity

$$\widetilde{F}^r \boldsymbol{u} = \boldsymbol{u} - (I - \widetilde{F}^r)\boldsymbol{u}$$

to obtain

$$\begin{cases} \boldsymbol{e}^{(0)} = \overline{\boldsymbol{u}} - \boldsymbol{u} = -(I - \widetilde{F}^r)\boldsymbol{u}, \\ \boldsymbol{e}^{(n+1)} = (I - \widetilde{F}^r)\boldsymbol{e}^{(n)}, \qquad\qquad n = 0, \dots, N-1, \end{cases} \tag{4.1.7}$$

and hence, in particular,

$$\boldsymbol{e}^{(N)} = \left(I - \widetilde{F}^r \right)^N (\overline{\boldsymbol{u}} - \boldsymbol{u}). \tag{4.1.8}$$

We now consider $\left(I - \widetilde{F}^r \right)^N$ as an operator on $\boldsymbol{X}$, and show that

$$\left\| \left(I - \widetilde{F}^r \right)^N \right\| \leq \|I - F^r\|^N \leq \left(\|I\| + \|F^r\| \right)^N \leq C(N),$$

by standard properties of operator norms and the triangle inequality, where the final inequality is due to the fact that $\widetilde{F}_r$ is a bounded operator on $\boldsymbol{X}$, as shown by (2.1.11). Using this, we obtain

$$\|D_N^r \overline{\boldsymbol{u}} - \boldsymbol{u}\| \leq C(N)\|\overline{\boldsymbol{u}} - \boldsymbol{u}\|,$$

and conclude by bounding the right-hand side with Lemma 2.1.5. □

Remark 4.1.3. Lemma 4.1.2 may be extended to establish deconvolution accuracy scaling as δ^{2N+2}, a result that aligns with the findings in [59, Lemma 2.14] for similar analyses conducted at the finite element level. However, achieving such scaling relies on the additional assumption that the function $\boldsymbol{u}$ resides within the Sobolev space H_0^{2N+2} or exhibits smooth periodicity across the domain. This prerequisite introduces a constraint on the regularity of $\boldsymbol{u}$, which is not universally applicable in practical scenarios.

While we have indeed observed instances of deconvolution accuracy exceeding the baseline δ^2 scaling for select problems, it is crucial to recognize that these occurrences are problem-dependent and may not generalize. In our study, as elaborated in Section 4.3, the numerical tests are intentionally designed to operate within non-smooth settings. This decision reflects the realities encountered in many practical simulations where flows are inherently irregular and do not conform to idealized smoothness assumptions.

Consequently, our numerical analysis of the ADL-ROM methodology adopts a conservative approach by imposing minimal assumptions on the regularity and smoothness of the underlying flow to be simulated. By doing so, we aim to ensure that our findings remain broadly applicable to the ROM configurations most commonly utilized by practitioners. For alternative approaches that incorporate more stringent regularity assumptions to derive enhanced error bounds, readers are referred to works such as [72].

This perspective underscores our commitment to balancing theoretical rigor with practical applicability, thereby making our analysis relevant to a wider range of real-world scenarios. ♠

4.2 Numerical Analysis of ADL-ROM

The following lemma gives a *a priori* bound, or stability estimate, for the BDF2 ADL-ROM scheme in 3.2.1. We note that our study was the first one in which the stability of the ADL-ROM is proved.

Lemma 4.2.1. *The approximation scheme in* (3.2.1) *has a solution* $\boldsymbol{u}_k^r$ *at* *each time step $k = 2, \ldots, K$ which satisfies the following* a priori *bound*

$$\|\boldsymbol{u}_K^r\|^2 + 2\nu\Delta t \sum_{\kappa=1}^{K-1} \left\|\nabla \boldsymbol{u}_{\kappa+1}^r\right\|^2 \leq \|\boldsymbol{u}_1^r\|^2 + \|2\boldsymbol{u}_1^r - \boldsymbol{u}_0^r\|^2 + 2\nu^{-1}\Delta t \sum_{\kappa=1}^{K-1} \|\boldsymbol{f}_{\kappa+1}\|^2 = C\left(\boldsymbol{u}_0^r, \boldsymbol{u}_1^r, \boldsymbol{f}, \nu^{-1}\right). \tag{4.2.1}$$

Proof. We choose $\boldsymbol{v}^r = \boldsymbol{u}_{k+1}^r$ in (3.2.1) and note that $b^*(D_N^r \overline{\boldsymbol{u}_{k+1}^r}, \boldsymbol{u}_{k+1}^r, \boldsymbol{u}_{k+1}^r) = 0$ by Lemma 1.2.1. Thus, we obtain

$$\frac{1}{2}\left(3\boldsymbol{u}_{k+1}^r - 4\boldsymbol{u}_k^r + \boldsymbol{u}_{k-1}^r, \boldsymbol{u}_{k+1}^r\right) + \nu\Delta t\left(\nabla \boldsymbol{u}_{k+1}^r, \nabla \boldsymbol{u}_{k+1}^r\right) = \Delta t\left(\boldsymbol{f}_{k+1}, \boldsymbol{u}_{k+1}^r\right). \tag{4.2.2}$$

The proof follows by using the Cauchy-Schwarz and Young inequalities, as well as the algebraic identity used in the proof of [66, Lemma 4.1], i.e.,

$$\frac{1}{2}(3a - 4b + c)a = \frac{1}{4}\left[a^2 + (2a-b)^2\right] - \frac{1}{4}\left[b^2 + (2b-c)^2\right] + \frac{1}{4}(a - 2b + c)^2. \tag{4.2.3}$$

□

The following lemma plays a crucial role in ensuring the optimality of the error bound presented in Theorem 4.2.3. Specifically, it explains how Assumption 1.3.1 is used to derive optimal error bounds within the ADL-ROM framework. By leveraging this assumption, we can rigorously demonstrate the theoretical foundations that underlie the efficiency and accuracy of reduced order modeling in this context.

It is important to highlight that this lemma, while indispensable in the context of reduced order models (ROMs), is not required when dealing with numerical methods based on finite element techniques. This distinction underscores a fundamental difference between the numerical analysis of ROMs and that of more traditional numerical methods, such as the finite element method. In the finite element context, the inherent structure and design of the method naturally provide the requisite stability and convergence properties without necessitating such additional lemmas.

This additional lemma addresses these challenges by ensuring that the approximation properties and stability conditions of the ADL-ROM are preserved, thus enabling the derivation of robust error bounds. This divergence in analytical requirements between ROMs and traditional methods exemplifies the specialized considerations necessary for advancing ROM-based numerical analysis.

Lemma 4.2.2. *For any $k = 1, \ldots, K-1$, the following inequality holds*

$$\Delta t \sum_{k=1}^{K-1} \left\|\boldsymbol{u}_{k+1}^r\right\| \left\|\nabla \boldsymbol{u}_{k+1}^r\right\| \left\|\nabla\left(\boldsymbol{u}_{k+1} - \boldsymbol{P}^r(\boldsymbol{u}_{k+1})\right)\right\|^2$$
$$\leq C\left(h^{2m+2} + h^{2m+4}\left\|S^r\right\|_2 + \Delta t^4\left(1 + \left\|S^r\right\|_2\right) + \sum_{j=r+1}^{R}\left\|\nabla \boldsymbol{\varphi}_j\right\|^2 \lambda_j\right), \tag{4.2.4}$$

where $\boldsymbol{P}^r$ is the ROM projection in Definition 1.3.2.

Proof. To prove (4.2.4), we use Lemma 4.2.1 and Assumption 1.3.1:

$$\begin{aligned}
&\Delta t \sum_{k=1}^{K-1} \left\|\boldsymbol{u}_{k+1}^r\right\| \left\|\nabla \boldsymbol{u}_{k+1}^r\right\| \left\|\nabla\left(\boldsymbol{u}_{k+1} - \boldsymbol{P}^r(\boldsymbol{u}_{k+1})\right)\right\|^2 \\
&\overset{(4.2.1)}{\leq} \widetilde{C}\Delta t \sum_{k=1}^{K-1} \left\|\nabla \boldsymbol{u}_{k+1}^r\right\| \left\|\nabla\left(\boldsymbol{u}_{k+1} - \boldsymbol{P}^r(\boldsymbol{u}_{k+1})\right)\right\|^2 \\
&\leq \ \widetilde{C} \max_{k=0,\ldots,K-1} \left\|\nabla\left(\boldsymbol{u}_{k+1} - \boldsymbol{P}^r(\boldsymbol{u}_{k+1})\right)\right\|^2 \Delta t \sum_{k=1}^{K-1} \left\|\nabla \boldsymbol{u}_{k+1}^r\right\| \\
&\overset{(4.2.1)}{\leq} C \max_{k=0,\ldots,K-1} \left\|\nabla\left(\boldsymbol{u}_{k+1} - \boldsymbol{P}^r(\boldsymbol{u}_{k+1})\right)\right\|^2 \\
&\overset{(1.3.8)}{\leq} C\left(h^{2m+2} + h^{2m+4}\left\|S^r\right\|_2 + \Delta t^4\left(1 + \left\|S^r\right\|_2\right) + \sum_{j=r+1}^{R}\left\|\nabla \boldsymbol{\varphi}_j\right\|^2 \lambda_j\right).
\end{aligned} \tag{4.2.5}$$

In the first inequality in (4.2.5), we employ the first term on the left-hand side of the stability estimate (4.2.1) in Lemma 4.2.1 to obtain that $\left\|\boldsymbol{u}_{k+1}^r\right\|$ is bounded by a constant $\widetilde{C}$ for every $k = 1, \ldots, K-1$. In the second inequality, we pull the maximum of $\left\|\nabla\left(\boldsymbol{u}_{k+1} - \boldsymbol{P}^r(\boldsymbol{u}_{k+1})\right)\right\|^2$ out of the sum, in order to be able to use again (4.2.1) on $\Delta t \sum_{k=1}^{K-1}\left\|\nabla \boldsymbol{u}_{k+1}^r\right\|$ next. In the third inequality, we estimate $\Delta t \sum_{k=1}^{K-1}\left\|\nabla \boldsymbol{u}_{k+1}^r\right\|$ using the Cauchy-Schwarz inequality first, and then the second addend on the left-hand side of (4.2.1), namely:

$$\Delta t \sum_{k=1}^{K-1}\left\|\nabla \boldsymbol{u}_{k+1}^r\right\| \leq \Delta t\ \widehat{C}\left(\sum_{k=1}^{K-1}\left\|\nabla \boldsymbol{u}_{k+1}^r\right\|^2\right)^{\frac{1}{2}} K^{\frac{1}{2}}$$

$$\leq \Delta t^{\frac{1}{2}} \,\widehat{\widehat{C}} \left(\Delta t \sum_{k=1}^{K-1} \left\| \nabla \boldsymbol{u}_{k+1}^r \right\|^2 \right)^{\frac{1}{2}} \frac{1}{\Delta t^{\frac{1}{2}}} \leq \widehat{\widehat{C}} \widetilde{C} = C.$$

In the fourth and final inequality in (4.2.5), we use (1.3.8) in Assumption 1.3.1 to conclude. We emphasize that we would have obtained a suboptimal estimate if we had used (1.3.6) instead of (1.3.8) in Assumption 1.3.1. Indeed, multiplying both sides of (1.3.6) by $K+1$

$$\begin{aligned}
&\|\nabla(\boldsymbol{u}_k - \boldsymbol{P}^r(\boldsymbol{u}_k))\|^2 \\
&\leq \sum_{k=1}^{K} \|\nabla(\boldsymbol{u}_k - \boldsymbol{P}^r(\boldsymbol{u}_k))\|^2 \\
&\leq C \left(h^{2m+2} + \|S^r\|_2 \, h^{2m+4} + (1 + \|S^r\|_2)\Delta t^4 + \sum_{j=r+1}^{R} \|\nabla \boldsymbol{\varphi}_j\|^2 \lambda_j \right) (K+1),
\end{aligned}$$

which would have multiplied (4.2.33) by $K+1$, which scales as $(\Delta t)^{-1}$ and would have caused the suboptimality. □

We are now in a position to present the primary contribution of this chapter, which is the establishment of a rigorous theoretical result for the Approximate Deconvolution Leray Reduced Order Model (ADL-ROM). This main result represents the first instance of providing *a priori* error bounds for the ADL-ROM framework. By meticulously addressing the mathematical and computational challenges inherent to this framework, we demonstrate the ability to derive robust bounds on the error of our approximation. These bounds not only validate the practical applicability of the ADL-ROM in capturing complex flow dynamics but also offer new insights into its stability and accuracy.

Theorem 4.2.3. *Let* $(\boldsymbol{u}, p)$ *be a strong solution of the NSE* (1.1.1)-(1.1.2) *with homogeneous Dirichlet boundary conditions*[1] *for* $\boldsymbol{u}$. *Assume the initial conditions are* $\boldsymbol{u}_0^r = \boldsymbol{P}^r(\boldsymbol{u}_0)$ *and* $\boldsymbol{u}_1^r = \boldsymbol{P}^r(\boldsymbol{u}_1)$. *Then, under the regularity assumption on the exact solution (Assumption 1.2.1), the assumption on the FE approximation (Assumption 1.2.2), and the assumption on the ROM projection error (Assumption 1.3.1), there exists* $\Delta t^* > 0$ *such that, for all*

[1] The results can be extended to the inhomogeneous case with some additional work, but without major difficulties.

$\Delta t < \Delta t^*$, *the following bound holds:*

$$\begin{aligned}
&\|\boldsymbol{u}_K - \boldsymbol{u}_K^r\|^2 + \Delta t \sum_{k=1}^{K-1} \left\| \nabla \left(\boldsymbol{u}_{k+1} - \boldsymbol{u}_{k+1}^r \right) \right\|^2 \\
\leq & C \mathcal{G}(\delta, \Delta t, h, \|S^r\|_2, \{\lambda_j\}_{j=r+1}^R, \{\|\nabla \boldsymbol{\varphi}_j\|\}_{j=r+1}^R),
\end{aligned} \tag{4.2.6}$$

where $C = C(N, \boldsymbol{u}_0^r, \boldsymbol{u}_1^r, \boldsymbol{f}, \nu)$ *and* $\mathcal{G}$ *is defined as*

$$\begin{aligned}
&\mathcal{G}(\delta, \Delta t, h, \|S^r\|_2, \{\lambda_j\}_{j=r+1}^R, \{\|\nabla \boldsymbol{\varphi}_j\|\}_{j=r+1}^R) \\
&= \left(1 + \|S^r\|_2^{\frac{1}{2}}\right) \left(h^{2m+4} + \Delta t^4 + \sum_{j=r+1}^R \lambda_j \right) \\
&+ \left(1 + \delta^2 \|S^r\|_2^{\frac{1}{2}}\right) \left(h^{2m+2} + \|S^r\|_2 \; h^{2m+4} + (1 + \|S^r\|_2) \Delta t^4 + \sum_{j=r+1}^R \|\nabla \boldsymbol{\varphi}_j\|^2 \lambda_j \right) \\
&+ \delta^4 \|S^r\|_2^{\frac{1}{2}} .
\end{aligned} \tag{4.2.7}$$

Proof. We begin with the weak formulation of the NSE given in (1.2.1). At a continuous level, recall that for $\boldsymbol{v}, \boldsymbol{w}, \boldsymbol{u} \in \boldsymbol{X}$,

$$b^*(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}) = b(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}) \doteq (\boldsymbol{u} \cdot \nabla \boldsymbol{v}, \boldsymbol{w}).$$

Then, at time t_{k+1}, the NSE in (1.2.1) with $\boldsymbol{v} = \boldsymbol{v}^r \in \boldsymbol{X}^r$ yield

$$\begin{aligned}
\left(\boldsymbol{u}_t^{k+1}, \boldsymbol{v}^r \right) + b^* \left(\boldsymbol{u}_{k+1}, \boldsymbol{u}_{k+1}, \boldsymbol{v}^r \right) + \nu \left(\nabla \boldsymbol{u}_{k+1}, \nabla \boldsymbol{v}^r \right) \\
- \left(p_{k+1}, \nabla \cdot \boldsymbol{v}^r \right) = \left(\boldsymbol{f}_{k+1}, \boldsymbol{v}^r \right).
\end{aligned} \tag{4.2.8}$$

Subtracting (3.2.1) from (4.2.8), we obtain

$$\begin{aligned}
&\left(\boldsymbol{u}_t^{k+1} - \frac{3\boldsymbol{u}_{k+1}^r - 4\boldsymbol{u}_k^r + \boldsymbol{u}_{k-1}^r}{2\Delta t}, \boldsymbol{v}^r \right) \\
&+ b^* \left(\boldsymbol{u}_{k+1}, \boldsymbol{u}_{k+1}, \boldsymbol{v}^r \right) - b^* \left(D_N^r \overline{\boldsymbol{u}_{k+1}^r}, \boldsymbol{u}_{k+1}^r, \boldsymbol{v}^r \right) \\
&+ \nu \left(\nabla \boldsymbol{u}_{k+1} - \nabla \boldsymbol{u}_{k+1}^r, \nabla \boldsymbol{v}^r \right) - \left(p_{k+1}, \nabla \cdot \boldsymbol{v}^r \right) = 0.
\end{aligned} \tag{4.2.9}$$

Adding and subtracting $\left(\dfrac{3\boldsymbol{u}_{k+1} - 4\boldsymbol{u}_k + \boldsymbol{u}_{k-1}}{2\Delta t}, \boldsymbol{v}^r\right)$, we have

$$\begin{aligned}
&\left(\boldsymbol{u}_t^{k+1} - \frac{3\boldsymbol{u}_{k+1} - 4\boldsymbol{u}_k + \boldsymbol{u}_{k-1}}{2\Delta t}, \boldsymbol{v}^r\right) \\
&+ \left(\frac{3\boldsymbol{u}_{k+1} - 4\boldsymbol{u}_k + \boldsymbol{u}_{k-1}}{2\Delta t} - \frac{3\boldsymbol{u}_{k+1}^r - 4\boldsymbol{u}_k^r + \boldsymbol{u}_{k-1}^r}{2\Delta t}, \boldsymbol{v}^r\right) \\
&b^*\left(\boldsymbol{u}_{k+1}, \boldsymbol{u}_{k+1}, \boldsymbol{v}^r\right) - b^*\left(D_N^r \overline{\boldsymbol{u}_{k+1}^r}, \boldsymbol{u}_{k+1}^r, \boldsymbol{v}^r\right) \\
&+\nu\left(\nabla \boldsymbol{u}_{k+1} - \nabla \boldsymbol{u}_{k+1}^r, \nabla \boldsymbol{v}^r\right) = \left(p_{k+1}, \nabla \cdot \boldsymbol{v}^r\right). \qquad (4.2.10)
\end{aligned}$$

We decompose the error as follows:

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

where $\boldsymbol{w}_{k+1}^r = \boldsymbol{P}^r\left(\boldsymbol{u}_{k+1}\right)$ is the L^2 projection of $\boldsymbol{u}_{k+1}$ on $\boldsymbol{X}^r$, i.e.,

$$\left(\boldsymbol{\eta}_{k+1}, \boldsymbol{\varphi}^r\right) = \left(\boldsymbol{u}_{k+1} - \boldsymbol{w}_{k+1}^r, \boldsymbol{\varphi}^r\right) = 0, \quad \forall \boldsymbol{\varphi}^r \in \boldsymbol{X}^r. \qquad (4.2.11)$$

Choosing $\boldsymbol{v}^r = \boldsymbol{\Phi}_{k+1}^r$ and letting $\boldsymbol{r}_{k+1} = \boldsymbol{u}_t^{k+1} - \dfrac{3\boldsymbol{u}_{k+1} - 4\boldsymbol{u}_k + \boldsymbol{u}_{k-1}}{2\Delta t}$, (4.2.10) becomes

$$\begin{aligned}
&\frac{1}{2\Delta t}\left(3(\boldsymbol{\eta}_{k+1} - \boldsymbol{\Phi}_{k+1}^r) - 4\left(\boldsymbol{\eta}_k - \boldsymbol{\Phi}_k^r\right) + (\boldsymbol{\eta}_{k-1} - \boldsymbol{\Phi}_{k-1}^r), \boldsymbol{\Phi}_{k+1}^r\right) \\
&-\nu\left(\nabla \boldsymbol{\Phi}_{k+1}^r, \nabla \boldsymbol{\Phi}_{k+1}^r\right) + \nu\left(\nabla \boldsymbol{\eta}_{k+1}, \nabla \boldsymbol{\Phi}_{k+1}^r\right) \\
&+b^*\left(\boldsymbol{u}_{k+1}, \boldsymbol{u}_{k+1}, \boldsymbol{\Phi}_{k+1}^r\right) - b^*\left(D_N^r \overline{\boldsymbol{u}_{k+1}^r}, \boldsymbol{u}_{k+1}^r, \boldsymbol{\Phi}_{k+1}^r\right) \\
&+\left(\boldsymbol{r}_{k+1}, \boldsymbol{\Phi}_{k+1}^r\right) = \left(p_{k+1}, \nabla \cdot \boldsymbol{\Phi}_{k+1}^r\right). \qquad (4.2.12)
\end{aligned}$$

Since $\boldsymbol{\Phi}_{k+1}^r \in \boldsymbol{X}^r$, (4.2.11) implies that $\left(\boldsymbol{\eta}_\kappa, \boldsymbol{\Phi}_{k+1}^r\right) = 0$ for $\kappa = k-1, k, k+1$. Thus, (4.2.12) becomes

$$\begin{aligned}
&\frac{1}{2\Delta t}\left(3\boldsymbol{\Phi}_{k+1}^r - 4\boldsymbol{\Phi}_k^r + \boldsymbol{\Phi}_{k-1}^r, \boldsymbol{\Phi}_{k+1}^r\right) + \nu\left(\nabla \boldsymbol{\Phi}_{k+1}^r, \nabla \boldsymbol{\Phi}_{k+1}^r\right) \\
&= \left(\boldsymbol{r}_{k+1}, \boldsymbol{\Phi}_{k+1}^r\right) + \nu\left(\nabla \boldsymbol{\eta}_{k+1}, \nabla \boldsymbol{\Phi}_{k+1}^r\right) + b^*\left(\boldsymbol{u}_{k+1}, \boldsymbol{u}_{k+1}, \boldsymbol{\Phi}_{k+1}^r\right) \\
&\quad -b^*\left(D_N^r \overline{\boldsymbol{u}_{k+1}^r}, \boldsymbol{u}_{k+1}^r, \boldsymbol{\Phi}_{k+1}^r\right) - \left(p_{k+1}, \nabla \cdot \boldsymbol{\Phi}_{k+1}^r\right). \qquad (4.2.13)
\end{aligned}$$

We write the left-hand side of (4.2.13) as

$$\frac{1}{4\Delta t}\left(\left\|\boldsymbol{\Phi}_{k+1}^r\right\|^2+\left\|2\boldsymbol{\Phi}_{k+1}^r-\boldsymbol{\Phi}_k^r\right\|^2\right)-\frac{1}{4\Delta t}\left(\left\|\boldsymbol{\Phi}_k^r\right\|^2+\left\|2\boldsymbol{\Phi}_k^r-\boldsymbol{\Phi}_{k-1}^r\right\|^2\right)$$
$$+\frac{1}{4\Delta t}\left\|\boldsymbol{\Phi}_{k+1}^r-2\boldsymbol{\Phi}_k^r+\boldsymbol{\Phi}_{k-1}^r\right\|^2+\nu\left\|\nabla\boldsymbol{\Phi}_{k+1}^r\right\|^2 \tag{4.2.14}$$

by applying the algebraic identity (4.2.3).

We now use the Cauchy-Schwarz, Young, and Poincaré inequalities to bound the first two addends on the right-hand side of (4.2.13) as follows:

$$\begin{aligned}\left(\boldsymbol{r}_{k+1},\boldsymbol{\Phi}_{k+1}^r\right)&\le\|\boldsymbol{r}_{k+1}\|\left\|\boldsymbol{\Phi}_{k+1}^r\right\|\\&\le C_1\left(\frac{1}{4\varepsilon}\|\boldsymbol{r}_{k+1}\|^2+\varepsilon\left\|\nabla\boldsymbol{\Phi}_{k+1}^r\right\|^2\right),\end{aligned} \tag{4.2.15}$$

$$\begin{aligned}\nu\left(\nabla\boldsymbol{\eta}_{k+1},\nabla\boldsymbol{\Phi}_{k+1}^r\right)&\le\nu\|\nabla\boldsymbol{\eta}_{k+1}\|\left\|\nabla\boldsymbol{\Phi}_{k+1}^r\right\|\\&\le C_2\left(\frac{1}{4\varepsilon}\|\nabla\boldsymbol{\eta}_{k+1}\|^2+\varepsilon\left\|\nabla\boldsymbol{\Phi}_{k+1}^r\right\|^2\right),\end{aligned} \tag{4.2.16}$$

with $\varepsilon>0$ to be chosen later.

The nonlinear terms on the right-hand side of (4.2.13) can be written as follows:

$$\begin{aligned}&b^*\left(\boldsymbol{u}_{k+1},\boldsymbol{u}_{k+1},\boldsymbol{\Phi}_{k+1}^r\right)-b^*\left(D_N^r\overline{\boldsymbol{u}_{k+1}^r},\boldsymbol{u}_{k+1}^r,\boldsymbol{\Phi}_{k+1}^r\right)\\
&=b^*\left(\boldsymbol{u}_{k+1},\boldsymbol{u}_{k+1},\boldsymbol{\Phi}_{k+1}^r\right)-b^*\left(D_N^r\overline{\boldsymbol{u}_{k+1}},\boldsymbol{u}_{k+1},\boldsymbol{\Phi}_{k+1}^r\right)\\
&\quad+b^*\left(D_N^r\overline{\boldsymbol{u}_{k+1}},\boldsymbol{u}_{k+1},\boldsymbol{\Phi}_{k+1}^r\right)-b^*\left(D_N^r\overline{\boldsymbol{u}_{k+1}^r},\boldsymbol{u}_{k+1}^r,\boldsymbol{\Phi}_{k+1}^r\right)\\
&=b^*\left(\boldsymbol{u}_{k+1}-D_N^r\overline{\boldsymbol{u}_{k+1}},\boldsymbol{u}_{k+1},\boldsymbol{\Phi}_{k+1}^r\right)+b^*\left(D_N^r\overline{\boldsymbol{u}_{k+1}},\boldsymbol{u}_{k+1},\boldsymbol{\Phi}_{k+1}^r\right)\\
&\quad-b^*\left(D_N^r\overline{\boldsymbol{u}_{k+1}^r},\boldsymbol{u}_{k+1},\boldsymbol{\Phi}_{k+1}^r\right)+b^*\left(D_N^r\overline{\boldsymbol{u}_{k+1}^r},\boldsymbol{u}_{k+1},\boldsymbol{\Phi}_{k+1}^r\right)-b^*\left(D_N^r\overline{\boldsymbol{u}_{k+1}^r},\boldsymbol{u}_{k+1}^r,\boldsymbol{\Phi}_{k+1}^r\right)\\
&=b^*\left(\boldsymbol{u}_{k+1}-D_N^r\overline{\boldsymbol{u}_{k+1}},\boldsymbol{u}_{k+1},\boldsymbol{\Phi}_{k+1}^r\right)+b^*\left(D_N^r\overline{\boldsymbol{\eta}_{k+1}},\boldsymbol{u}_{k+1},\boldsymbol{\Phi}_{k+1}^r\right)\\
&\quad-b^*\left(D_N^r\overline{\boldsymbol{\Phi}_{k+1}^r},\boldsymbol{u}_{k+1},\boldsymbol{\Phi}_{k+1}^r\right)+b^*\left(D_N^r\overline{\boldsymbol{u}_{k+1}^r},\boldsymbol{\eta}_{k+1},\boldsymbol{\Phi}_{k+1}^r\right)-\cancel{b^*\left(D_N^r\overline{\boldsymbol{u}_{k+1}^r},\boldsymbol{\Phi}_{k+1}^r,\boldsymbol{\Phi}_{k+1}^r\right)},\end{aligned} \tag{4.2.17}$$

where in the last term we have used Lemma 1.2.1.

For the last three addends in (4.2.17), we apply Lemmas 1.2.1 and 4.1.1, and Young inequality:

$$\begin{aligned}
b^*\left(D_N^r\overline{\boldsymbol{u}_{k+1}^r},\boldsymbol{\eta}_{k+1},\boldsymbol{\Phi}_{k+1}^r\right) &\overset{1.2.1}{\leq} C\left\|D_N^r\overline{\boldsymbol{u}_{k+1}^r}\right\|^{\frac{1}{2}}\left\|\nabla D_N^r\overline{\boldsymbol{u}_{k+1}^r}\right\|^{\frac{1}{2}}\left\|\nabla\boldsymbol{\eta}_{k+1}\right\|\left\|\nabla\boldsymbol{\Phi}_{k+1}^r\right\| \\
&\overset{4.1.1}{\leq} C\left\|\boldsymbol{u}_{k+1}^r\right\|^{\frac{1}{2}}\left\|\nabla\boldsymbol{u}_{k+1}^r\right\|^{\frac{1}{2}}\left\|\nabla\boldsymbol{\eta}_{k+1}\right\|\left\|\nabla\boldsymbol{\Phi}_{k+1}^r\right\| \\
&\leq C_3\left(\frac{1}{4\varepsilon}\left\|\boldsymbol{u}_{k+1}^r\right\|\left\|\nabla\boldsymbol{u}_{k+1}^r\right\|\left\|\nabla\boldsymbol{\eta}_{k+1}\right\|^2+\varepsilon\left\|\nabla\boldsymbol{\Phi}_{k+1}^r\right\|^2\right).
\end{aligned} \tag{4.2.18}$$

For the second

$$\begin{aligned}
b^*\left(D_N^r\overline{\boldsymbol{\eta}_{k+1}},\boldsymbol{u}_{k+1},\boldsymbol{\Phi}_{k+1}^r\right) &\overset{1.2.1}{\leq} C\left\|D_N^r\overline{\boldsymbol{\eta}_{k+1}}\right\|^{\frac{1}{2}}\left\|\nabla D_N^r\overline{\boldsymbol{\eta}_{k+1}}\right\|^{\frac{1}{2}}\left\|\nabla\boldsymbol{u}_{k+1}\right\|\left\|\nabla\boldsymbol{\Phi}_{k+1}^r\right\| \\
&\overset{4.1.1}{\leq} C\left\|\boldsymbol{\eta}_{k+1}\right\|^{\frac{1}{2}}\left\|S^r\right\|_2^{\frac{1}{4}}\left\|\boldsymbol{\eta}_{k+1}\right\|^{\frac{1}{2}}\left\|\nabla\boldsymbol{u}_{k+1}\right\|\left\|\nabla\boldsymbol{\Phi}_{k+1}^r\right\| \\
&\leq C_4\left(\frac{1}{4\varepsilon}\left\|S^r\right\|_2^{\frac{1}{2}}\left\|\boldsymbol{\eta}_{k+1}\right\|^2\left\|\nabla\boldsymbol{u}_{k+1}\right\|^2+\varepsilon\left\|\nabla\boldsymbol{\Phi}_{k+1}^r\right\|^2\right).
\end{aligned} \tag{4.2.19}$$

Lastly,

$$\begin{aligned}
b^*\left(D_N^r\overline{\boldsymbol{\Phi}_{k+1}^r},\boldsymbol{u}_{k+1},\boldsymbol{\Phi}_{k+1}^r\right) &\overset{1.2.1}{\leq} C\left\|D_N^r\overline{\boldsymbol{\Phi}_{k+1}^r}\right\|^{\frac{1}{2}}\left\|\nabla D_N^r\overline{\boldsymbol{\Phi}_{k+1}^r}\right\|^{\frac{1}{2}}\left\|\nabla\boldsymbol{u}_{k+1}\right\|\left\|\nabla\boldsymbol{\Phi}_{k+1}^r\right\| \\
&\overset{4.1.1}{\leq} C\left\|\boldsymbol{\Phi}_{k+1}^r\right\|^{\frac{1}{2}}\left\|\nabla\boldsymbol{u}_{k+1}\right\|\left\|\nabla\boldsymbol{\Phi}_{k+1}^r\right\|^{\frac{3}{2}} \\
&\leq C_5\left(\frac{1}{4\varepsilon}\left\|\nabla\boldsymbol{u}_{k+1}\right\|^4\left\|\boldsymbol{\Phi}_{k+1}^r\right\|^2+\varepsilon\left\|\nabla\boldsymbol{\Phi}_{k+1}^r\right\|^2\right).
\end{aligned} \tag{4.2.20}$$

Similarly, for the first addend in (4.2.17), we get

$$\begin{aligned}
&b^*\left(\boldsymbol{u}_{k+1}-D_N^r\overline{\boldsymbol{u}_{k+1}},\boldsymbol{u}_{k+1},\boldsymbol{\Phi}_{k+1}^r\right) \\
&\overset{1.2.1}{\leq} C\left\|\boldsymbol{u}_{k+1}-D_N^r\overline{\boldsymbol{u}_{k+1}}\right\|^{\frac{1}{2}}\left\|\nabla\left(\boldsymbol{u}_{k+1}-D_N^r\overline{\boldsymbol{u}_{k+1}}\right)\right\|^{\frac{1}{2}}\left\|\nabla\boldsymbol{u}_{k+1}\right\|\left\|\nabla\boldsymbol{\Phi}_{k+1}^r\right\| \\
&\leq C_6\left(\frac{1}{4\varepsilon}\left\|\boldsymbol{u}_{k+1}-D_N^r\overline{\boldsymbol{u}_{k+1}}\right\|\left\|\nabla\left(\boldsymbol{u}_{k+1}-D_N^r\overline{\boldsymbol{u}_{k+1}}\right)\right\|\left\|\nabla\boldsymbol{u}_{k+1}\right\|^2\right. \\
&\qquad\left.+\varepsilon\left\|\nabla\boldsymbol{\Phi}_{k+1}^r\right\|^2\right).
\end{aligned} \tag{4.2.21}$$

Since $\boldsymbol{\Phi}_{k+1}^r \in \boldsymbol{X}^r$ and $\boldsymbol{X}^r$ is a space of weakly divergence-free functions, the pressure term on the right-hand side of (4.2.13) can be written as

$$-\left(p_{k+1}, \nabla \cdot \boldsymbol{\Phi}_{k+1}^r\right) = -\left(p_{k+1} - q^h, \nabla \cdot \boldsymbol{\Phi}_{k+1}^r\right),$$

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

$$-\left(p_{k+1}, \nabla \cdot \boldsymbol{\Phi}_{k+1}^r\right) \leq C_7 \left(\frac{1}{4\varepsilon} \left\|p_{k+1} - q^h\right\|^2 + \varepsilon \left\|\nabla \boldsymbol{\Phi}_{k+1}^r\right\|^2\right). \qquad (4.2.22)$$

Now choose $\varepsilon > 0$ such that $4\nu - 4\varepsilon\left(C_1 + C_2 + C_3 + C_4 + C_5 + C_6 + C_7\right) = C_8$ for some $C_8 \in (0, 4\nu)$. We also define $C_9 \doteq C_9(\nu^{-1}) = \max_i\{\frac{C_i}{\epsilon}\}_{i=1,\ldots,7}$ once the choice of ϵ for C_8 has been made. Then, substituting inequalities (4.2.14)-(4.2.16), (4.2.18)-(4.2.21), and (4.2.22) in (4.2.13), we obtain

$$\begin{aligned}
&\left\|\boldsymbol{\Phi}_{k+1}^r\right\|^2 + \left\|2\boldsymbol{\Phi}_{k+1}^r - \boldsymbol{\Phi}_k^r\right\|^2 - \left(\left\|\boldsymbol{\Phi}_k^r\right\|^2 + \left\|2\boldsymbol{\Phi}_k^r - \boldsymbol{\Phi}_{k-1}^r\right\|^2\right) \\
&\qquad + \left\|\boldsymbol{\Phi}_{k+1}^r - 2\boldsymbol{\Phi}_k^r + \boldsymbol{\Phi}_{k-1}^r\right\|^2 + C_8 \Delta t \left\|\nabla \boldsymbol{\Phi}_{k+1}^r\right\|^2 \\
&\leq C_9 \Big(\Delta t \left\|\boldsymbol{r}_{k+1}\right\|^2 + \Delta t \left\|\nabla \boldsymbol{\eta}_{k+1}\right\|^2 + \Delta t \left\|\boldsymbol{u}_{k+1}^r\right\| \left\|\nabla \boldsymbol{u}_{k+1}^r\right\| \left\|\nabla \boldsymbol{\eta}_{k+1}\right\|^2 \\
&\qquad + \Delta t \left\|S^r\right\|_2^{\frac{1}{2}} \left\|\boldsymbol{\eta}_{k+1}\right\|^2 \left\|\nabla \boldsymbol{u}_{k+1}\right\|^2 + \Delta t \left\|\nabla \boldsymbol{u}_{k+1}\right\|^4 \left\|\boldsymbol{\Phi}_{k+1}^r\right\|^2 \\
&\qquad + \Delta t \left\|\boldsymbol{u}_{k+1} - D_N^r \overline{\boldsymbol{u}_{k+1}}\right\| \left\|\nabla\left(\boldsymbol{u}_{k+1} - D_N^r \overline{\boldsymbol{u}_{k+1}}\right)\right\| \left\|\nabla \boldsymbol{u}_{k+1}\right\|^2 \\
&\qquad + \Delta t \left\|p_{k+1} - q_h\right\|^2\Big).
\end{aligned} \qquad (4.2.23)$$

Summing (4.2.23) from $k = 1$ to $k = K - 1$, we have

$$\begin{aligned}
&\left\|\boldsymbol{\Phi}_K^r\right\|^2 + C_8 \Delta t \sum_{k=1}^{K-1} \left\|\nabla \boldsymbol{\Phi}_{k+1}^r\right\|^2 \leq \left\|\boldsymbol{\Phi}_1^r\right\|^2 + \left\|2\boldsymbol{\Phi}_1^r - \boldsymbol{\Phi}_0^r\right\|^2 \\
&+ C_9 \Delta t \Bigg(\sum_{k=1}^{K-1} \left\|\boldsymbol{r}_{k+1}\right\|^2 + \sum_{k=1}^{K-1} \left\|\nabla \boldsymbol{\eta}_{k+1}\right\|^2 + \sum_{k=1}^{K-1} \left\|\boldsymbol{u}_{k+1}^r\right\| \left\|\nabla \boldsymbol{u}_{k+1}^r\right\| \left\|\nabla \boldsymbol{\eta}_{k+1}\right\|^2 \\
&+ \sum_{k=1}^{K-1} \left\|\boldsymbol{u}_{k+1} - D_N^r \overline{\boldsymbol{u}_{k+1}}\right\| \left\|\nabla\left(\boldsymbol{u}_{k+1} - D_N^r \overline{\boldsymbol{u}_{k+1}}\right)\right\| \left\|\nabla \boldsymbol{u}_{k+1}\right\|^2 \\
&+ \left\|S^r\right\|_2^{\frac{1}{2}} \sum_{k=1}^{K-1} \left\|\nabla \boldsymbol{u}_{k+1}\right\|^2 \left\|\boldsymbol{\eta}_{k+1}\right\|^2 + \sum_{k=1}^{K-1} \left\|p_{k+1} - q_h\right\|^2
\end{aligned}$$

$$+ \sum_{k=1}^{K-1} \|\nabla \boldsymbol{u}_{k+1}\|^4 \left\|\boldsymbol{\Phi}_{k+1}^r\right\|^2 \Bigg) . \tag{4.2.24}$$

The first two terms on the right-hand side of (4.2.24) vanish, since $\boldsymbol{u}_0^r = \boldsymbol{w}_0^r$ and $\boldsymbol{u}_1^r = \boldsymbol{w}_1^r$.

To estimate the third term on the right-hand side of (4.2.24), we expand $\boldsymbol{u}^k$ and $\boldsymbol{u}^{k-1}$ about t_{k+1} with Taylor's theorem. Then,

$$\boldsymbol{u}^k = \boldsymbol{u}^{k+1} - \boldsymbol{u}_t^{k+1}\Delta t + \frac{1}{2}\boldsymbol{u}_{tt}^{k+1}\Delta t^2 - \int_{t_k}^{t_{k+1}} \frac{1}{2}\boldsymbol{u}_{ttt}(s)(t_k - s)^2 ds, \tag{4.2.25}$$

$$\boldsymbol{u}^{k-1} = \boldsymbol{u}^{k+1} - 2\boldsymbol{u}_t^{k+1}\Delta t + 2\boldsymbol{u}_{tt}^{k+1}\Delta t^2 - \int_{t_{k-1}}^{t_{k+1}} \frac{1}{2}\boldsymbol{u}_{ttt}(s)(t_{k-1} - s)^2 ds. \tag{4.2.26}$$

Multiplying (4.2.25) by -4, adding the result to (4.2.26), and rearranging terms, we obtain

$$\begin{aligned} 2\boldsymbol{u}_t^{k+1}\Delta t - \left(3\boldsymbol{u}^{k+1} - 4\boldsymbol{u}^k + \boldsymbol{u}^{k-1}\right) = -2\int_{t_k}^{t_{k+1}} &\boldsymbol{u}_{ttt}(s)(t_k - s)^2 ds \\ + \int_{t_{k-1}}^{t_{k+1}} &\frac{1}{2}\boldsymbol{u}_{ttt}(s)(t_{k-1} - s)^2 ds. \end{aligned} \tag{4.2.27}$$

The left-hand side of (4.2.27) divided by $2\Delta t$ is $\boldsymbol{r}_{k+1}$. When taking the L^2 norm in space of (4.2.27), the result is the norm of a difference of two similar integrals. We use the triangle inequality to bound these norms individually such that both have a factor of $\Delta t^{3/2}$ multiplied by a constant. We now demonstrate how to bound the first term on the right-hand side of (4.2.27):

$$\left\| -\frac{1}{2\Delta t} \cdot 2\int_{t_k}^{t_{k+1}} \boldsymbol{u}_{ttt}(s)(t_k - s)^2 ds \right\| = \left[\int_\Omega \left(\int_{t_k}^{t_{k+1}} \boldsymbol{u}_{ttt}(s)\frac{(t_k - s)^2}{\Delta t} ds\right)^2 d\boldsymbol{x}\right]^{\frac{1}{2}} . \tag{4.2.28}$$

Using the Cauchy-Schwarz inequality for the time integral yields

$$\begin{aligned} &\left[\int_\Omega \left(\int_{t_k}^{t_{k+1}} \boldsymbol{u}_{ttt}(s)\frac{(t_k - s)^2}{\Delta t} ds\right)^2 d\boldsymbol{x}\right]^{\frac{1}{2}} \\ &\quad \leq \left[\int_\Omega \left(\int_{t_k}^{t_{k+1}} \big(\boldsymbol{u}_{ttt}(s)\big)^2 ds\right)\left(\int_{t_k}^{t_{k+1}} \frac{(t_k - s)^4}{\Delta t^2} ds\right) d\boldsymbol{x}\right]^{\frac{1}{2}} . \end{aligned} \tag{4.2.29}$$

Noting that $|t_k - s| \leq \Delta t$ and evaluating the integrals in (4.2.29), we obtain

$$\left[\int_\Omega \left(\int_{t_k}^{t_{k+1}} \big(\boldsymbol{u}_{ttt}(s)\big)^2 ds\right)\left(\int_{t_k}^{t_{k+1}} \frac{(t_k - s)^4}{\Delta t^2} ds\right) d\boldsymbol{x}\right]^{\frac{1}{2}} \leq \|\boldsymbol{u}_{ttt}\|_{L^2(L^2)} \Delta t^{\frac{3}{2}}, \tag{4.2.30}$$

where $\|\cdot\|_{L^2(L^2)}$ refers to the L^2 norm in both space and time over Ω and the time interval $[t_{k-1}, t_{k+1}]$. This change to a larger time interval than in (4.2.29) is informed by the next point, which is that the second integral in (4.2.27) may be bounded in a similar manner by noting that $|t_{k-1} - s| \leq 2\Delta t$ over $[t_{k-1}, t_{k+1}]$.

Using (4.2.27)-(4.2.30) with a constant C_{10} depending on $\|\boldsymbol{u}_{ttt}\|_{L^2(L^2)}$, we obtain the following bound:

$$\Delta t \sum_{k=1}^{K-1} \|\boldsymbol{r}_{k+1}\|^2 \leq C_{10} \Delta t^4. \tag{4.2.31}$$

Using (1.3.6), the fourth term on the right-hand side of (4.2.24) can be estimated as follows:

$$\begin{aligned} &\Delta t \sum_{k=1}^{K-1} \|\nabla \boldsymbol{\eta}_{k+1}\|^2 \\ &\leq C_{11} \left(h^{2m+2} + h^{2m+4} \|S^r\|_2 + \Delta t^4 \left(1 + \|S^r\|_2\right) + \sum_{j=r+1}^{R} \left\|\nabla \boldsymbol{\varphi}_j\right\|^2 \lambda_j \right). \end{aligned} \tag{4.2.32}$$

To estimate the fifth term on the right-hand side of (4.2.24), we use Lemma 4.2.2:

$$\begin{aligned} &\Delta t \sum_{k=1}^{K-1} \left\|\boldsymbol{u}_{k+1}^r\right\| \left\|\nabla \boldsymbol{u}_{k+1}^r\right\| \|\nabla \boldsymbol{\eta}_{k+1}\|^2 \\ &\leq C_{12} \left(h^{2m+2} + h^{2m+4} \|S^r\|_2 + \Delta t^4 \left(1 + \|S^r\|_2\right) + \sum_{j=r+1}^{R} \left\|\nabla \boldsymbol{\varphi}_j\right\|^2 \lambda_j \right). \end{aligned} \tag{4.2.33}$$

To estimate the sixth term on the right-hand side of (4.2.24), we use

(4.1.2), Lemma 4.1.2, and Assumption 1.2.1:

$$\begin{aligned}
&\Delta t \sum_{k=1}^{K-1} \left\| \boldsymbol{u}_{k+1} - D_N^r \overline{\boldsymbol{u}_{k+1}^r} \right\| \left\| \nabla \left(\boldsymbol{u}_{k+1} - D_N^r \overline{\boldsymbol{u}_{k+1}^r} \right) \right\| \left\| \nabla \boldsymbol{u}_{k+1} \right\|^2 \\
&\overset{(4.1.2)}{\leq} \hat{C}_{13} \|S^r\|_2^{\frac{1}{2}} \ \Delta t \sum_{k=1}^{K-1} \left\| \boldsymbol{u}_{k+1} - D_N^r \overline{\boldsymbol{u}_{k+1}^r} \right\|^2 \left\| \nabla \boldsymbol{u}_{k+1} \right\|^2 \\
&\overset{4.1.2}{\leq} \widetilde{C}_{13} \|S^r\|_2^{\frac{1}{2}} \ \Delta t \sum_{k=1}^{K-1} \left\| \nabla \boldsymbol{u}_{k+1} \right\|^2 \left[\left(h^{2m+4} + \Delta t^4 + \sum_{j=r+1}^{R} \lambda_j \right) \right. \\
&\quad \left. + \delta^2 \left(h^{2m+2} + h^{2m+4} \left\| S^r \right\|_2 + \Delta t^4 \left(1 + \left\| S^r \right\|_2 \right) + \sum_{j=r+1}^{R} \left\| \nabla \boldsymbol{\varphi}_j \right\|^2 \lambda_j \right) + \delta^4 \right] \\
&\overset{1.2.1}{\leq} C_{13} \|S^r\|_2^{\frac{1}{2}} \ \left[\left(h^{2m+4} + \Delta t^4 + \sum_{j=r+1}^{R} \lambda_j \right) \right. \\
&\quad \left. + \delta^2 \left(h^{2m+2} + h^{2m+4} \left\| S^r \right\|_2 + \Delta t^4 \left(1 + \left\| S^r \right\|_2 \right) + \sum_{j=r+1}^{R} \left\| \nabla \boldsymbol{\varphi}_j \right\|^2 \lambda_j \right) + \delta^4 \right] .
\end{aligned} \tag{4.2.34}$$

Using Assumption 1.3.1, the seventh term on the right-hand side of (4.2.24) can be bounded as follows:

$$\begin{aligned}
&\|S^r\|_2^{\frac{1}{2}} \ \Delta t \sum_{k=1}^{K-1} \left\| \nabla \boldsymbol{u}_{k+1} \right\|^2 \left\| \boldsymbol{\eta}_{k+1} \right\|^2 \\
&\overset{(1.3.7)}{\leq} \widetilde{C}_{14} \Delta t \left\| S^r \right\|_2^{\frac{1}{2}} \sum_{k=1}^{K-1} \left\| \nabla \boldsymbol{u}_{k+1} \right\|^2 \left(h^{2m+4} + \Delta t^4 + \sum_{j=r+1}^{R} \lambda_j \right) \\
&\overset{1.2.1}{\leq} C_{14} \left\| S^r \right\|_2^{\frac{1}{2}} \left(h^{2m+4} + \Delta t^4 + \sum_{j=r+1}^{R} \lambda_j \right) .
\end{aligned} \tag{4.2.35}$$

Since in (4.2.22) q^h is an arbitrary function in Q^h, we can use Assumption

1.2.2 to bound the eighth term on the right-hand side of (4.2.24) as follows:

$$\Delta t \sum_{k=1}^{K-1} \|p_{k+1} - q_h\|^2 \leq C_{15} h^{2m+2}. \tag{4.2.36}$$

Collecting (4.2.31)-(4.2.36), equation (4.2.24) becomes

$$\begin{aligned}
&\|\boldsymbol{\Phi}_K^r\|^2 + C_8 \Delta t \sum_{k=1}^{K-1} \left\|\nabla \boldsymbol{\Phi}_{k+1}^r\right\|^2 \\
&\quad \leq C_{16} \Bigg\{ \Delta t \sum_{k=1}^{K-1} \|\nabla \boldsymbol{u}_{k+1}\|^4 \left\|\boldsymbol{\Phi}_{k+1}^r\right\|^2 + \Delta t^4 \\
&\qquad + \left(h^{2m+2} + h^{2m+4} \|S^r\|_2 + \Delta t^4 \left(1 + \|S^r\|_2\right) + \sum_{j=r+1}^{R} \left\|\nabla \boldsymbol{\varphi}_j\right\|^2 \lambda_j \right) \\
&\qquad + \left(h^{2m+2} + h^{2m+4} \|S^r\|_2 + \Delta t^4 \left(1 + \|S^r\|_2\right) + \sum_{j=r+1}^{R} \left\|\nabla \boldsymbol{\varphi}_j\right\|^2 \lambda_j \right) \\
&\qquad + \|S^r\|_2^{\frac{1}{2}} \left(h^{2m+4} + \Delta t^4 + \sum_{j=r+1}^{R} \lambda_j \right) \\
&\qquad + \delta^2 \|S^r\|_2^{\frac{1}{2}} \left(h^{2m+2} + h^{2m+4} \|S^r\|_2 + \Delta t^4 \left(1 + \|S^r\|_2\right) + \sum_{j=r+1}^{R} \left\|\nabla \boldsymbol{\varphi}_j\right\|^2 \lambda_j \right) \\
&\qquad + \delta^4 \|S^r\|_2^{\frac{1}{2}} + \|S^r\|_2^{\frac{1}{2}} \left(h^{2m+4} + \Delta t^4 + \sum_{j=r+1}^{R} \lambda_j \right) + h^{2m+2} \Bigg\} \\
&\quad = C_{17} \Bigg\{ \Delta t \sum_{k=1}^{K-1} \|\nabla \boldsymbol{u}_{k+1}\|^4 \left\|\boldsymbol{\Phi}_{k+1}^r\right\|^2 \\
&\qquad + \left(h^{2m+2} + h^{2m+4} \|S^r\|_2 + \Delta t^4 \left(1 + \|S^r\|_2\right) + \sum_{j=r+1}^{R} \left\|\nabla \boldsymbol{\varphi}_j\right\|^2 \lambda_j \right) \\
&\qquad + \delta^2 \|S^r\|_2^{\frac{1}{2}} \left(h^{2m+2} + h^{2m+4} \|S^r\|_2 + \Delta t^4 \left(1 + \|S^r\|_2\right) + \sum_{j=r+1}^{R} \left\|\nabla \boldsymbol{\varphi}_j\right\|^2 \lambda_j \right)
\end{aligned}$$

$$
\begin{aligned}
&\quad + \delta^4 \|S^r\|_2^{\frac{1}{2}} + \|S^r\|_2^{\frac{1}{2}} \left(h^{2m+4} + \Delta t^4 + \sum_{j=r+1}^{R} \lambda_j \right) \Bigg\} \\
&\overset{(4.2.7)}{\doteq} C_{17} \Bigg\{ \Delta t \sum_{k=1}^{K-1} \|\nabla \boldsymbol{u}_{k+1}\|^4 \left\|\boldsymbol{\Phi}_{k+1}^r\right\|^2 \\
&\qquad + \mathcal{G}(\delta, \Delta t, h, \|S^r\|_2, \{\lambda_j\}_{j=r+1}^R, \{\|\nabla \boldsymbol{\varphi}_j\|\}_{j=r+1}^R) \Bigg\}. \qquad (4.2.37)
\end{aligned}
$$

Lemma 1.2.2 implies that, for small enough Δt (in particular, $\Delta t < (C_{17} \max_{1\le k\le K} \|\nabla \boldsymbol{u}_k\|^4)^{-1}$), the following inequality holds

$$
\begin{aligned}
&\|\boldsymbol{\Phi}_K^r\|^2 + C_8 \Delta t \sum_{k=1}^{K-1} \left\|\nabla \boldsymbol{\Phi}_{k+1}^r\right\|^2 \\
&\quad \le C_{18}\ \mathcal{G}(\delta, \Delta t, h, \|S^r\|_2, \{\lambda_j\}_{j=r+1}^R, \{\|\nabla \boldsymbol{\varphi}_j\|\}_{j=r+1}^R). \qquad (4.2.38)
\end{aligned}
$$

By using (4.2.38), the triangle inequality, and (1.3.6)-(1.3.7), we obtain

$$
\begin{aligned}
&\|\boldsymbol{u}_K - \boldsymbol{u}_K^r\|^2 + \Delta t \sum_{k=1}^{K-1} \left\|\nabla \left(\boldsymbol{u}_{k+1} - \boldsymbol{u}_{k+1}^r\right)\right\|^2 \qquad (4.2.39)\\
&\quad \le C\mathcal{G}(\delta, \Delta t, h, \|S^r\|_2, \{\lambda_j\}_{j=r+1}^R, \{\|\nabla \boldsymbol{\varphi}_j\|\}_{j=r+1}^R). \qquad (4.2.40)
\end{aligned}
$$

This completes the proof. □

4.3 Numerical Results

Goals In this section, we perform a comprehensive numerical investigation to evaluate the theoretical results derived in Sections 4.1 and 4.2. The primary objective of this analysis is to determine whether the outcomes of our numerical experiments align with the *a priori* error bounds established for the ROM approximate deconvolution operator, as detailed in Lemma 4.1.2, and for the ADL-ROM approximation, as formulated in Theorem 4.2.3.

To ensure consistency and fidelity to the theoretical framework, this section concentrates exclusively on the van Cittert approximate deconvolution (AD) method described in Section 3.1.1. As such, any references to ADL-ROM in the context of our numerical results should be explicitly interpreted as ADL-ROMs constructed using the van Cittert regularization method.

Furthermore, it is important to note that, as in previous sections, the methodologies under consideration are applied to the Navier-Stokes Equations (NSE) (1.1.1) and (1.1.2). By grounding our investigation in this well-established mathematical framework, we aim to rigorously validate the theoretical developments and provide deeper insights into the performance and accuracy of the proposed ADL-ROM approach.

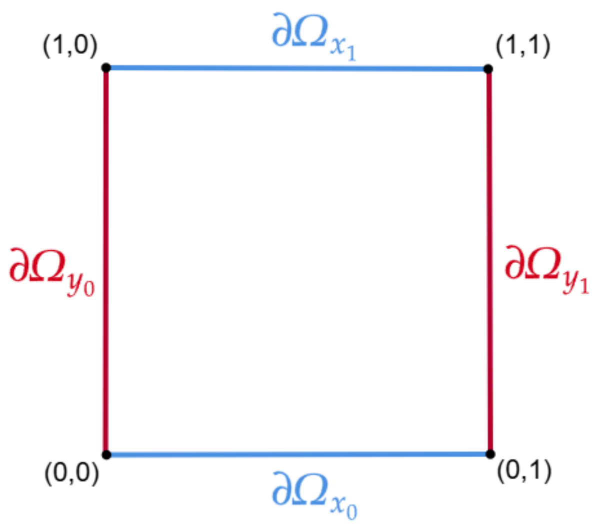

Figure 4.1: The computational domain, Ω.

Computational setting For our numerical investigation, we adopt a similar test problem and computational setup as those described in [99, Section 5]. The problem involves the 2D incompressible fluid flow with a known analytical solution. Specifically, the exact velocity, denoted by $\boldsymbol{u} = (u, v)$, is defined by

$$u = \frac{2}{\pi} \arctan\left(-\frac{1}{2\nu}(y-t)\right)\sin(\pi y) \tag{4.3.1}$$

and

$$v = \frac{2}{\pi} \arctan\left(-\frac{1}{2\nu}(x-t)\right)\sin(\pi x) \tag{4.3.2}$$

Additionally, the exact pressure is constant, given by $p = 0$.

We conduct two tests by setting two different viscosity values, i.e., $\nu = 10^{-3}$ and $\nu = 5 \times 10^{-4}$, and then select the forcing term to match the exact solution.

We consider the motion of the incompressible flow within the computational domain $\Omega \doteq [0,1] \times [0,1]$, illustrated in Figure 4.1.

The exact solution satisfies the following boundary conditions:

$$\begin{cases} u = 0 & \text{on } \partial\Omega_{x_0} \text{ and } \partial\Omega_{x_1}, \\ v = 0 & \text{on } \partial\Omega_{y_0} \text{ and } \partial\Omega_{y_1}, \\ \frac{\partial v}{\partial n} = 0 & \text{on } \partial\Omega_{x_0} \text{ and } \partial\Omega_{x_1}, \\ \frac{\partial u}{\partial n} = 0 & \text{on } \partial\Omega_{y_0} \text{ and } \partial\Omega_{y_1}. \end{cases} \tag{4.3.3}$$

Since the snapshots are obtained from the exact solution (by interpolating it on the finite element mesh), as explained after, the snapshots inherit the homogeneous Dirichlet and Neumann boundary conditions (4.3.3) satisfied by the exact solution. Thus, at the ROM level there is no need to enforce the boundary conditions since the POD basis and, as a result, the ROM solution will automatically satisfy the homogeneous Dirichlet and Neumann boundary conditions (4.3.3).

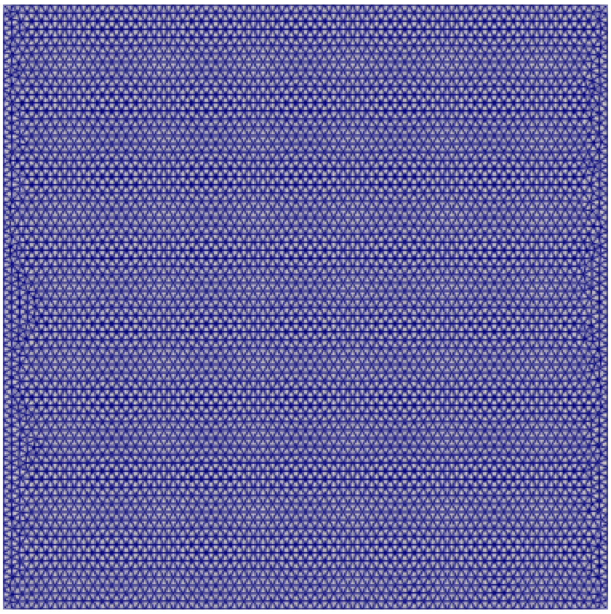

Figure 4.2: The FE mesh.

Snapshot generation The numerical tests are performed on a triangular uniform mesh with $h = \frac{1}{64}$ (see Figure 4.2). Spatial discretization is achieved using the inf-sup stable Taylor-Hood $\mathbb{P}^2 - \mathbb{P}^1$ finite element pair for velocity and pressure, respectively. This discretization results in a finite element space with a total dimension of $D \doteq D_h \doteq D_h^u + D_h^p = 38578 + 4887 = 43465$. We gather the snapshots over the time interval $[0, 1]$ at intervals of $\Delta t = 10^{-2}$, interpolating the analytical solution (u, v) on the finite element mesh.

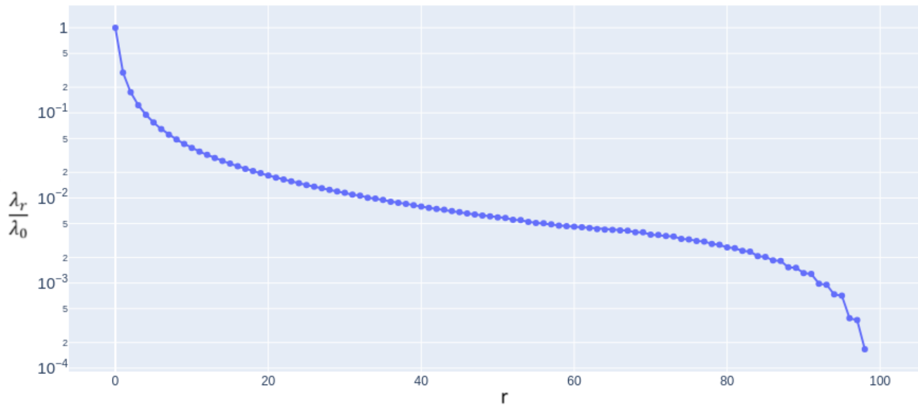

Figure 4.3: Normalized singular values from POD procedure with $\nu = 10^{-3}$ and $r_{max} = 100$.

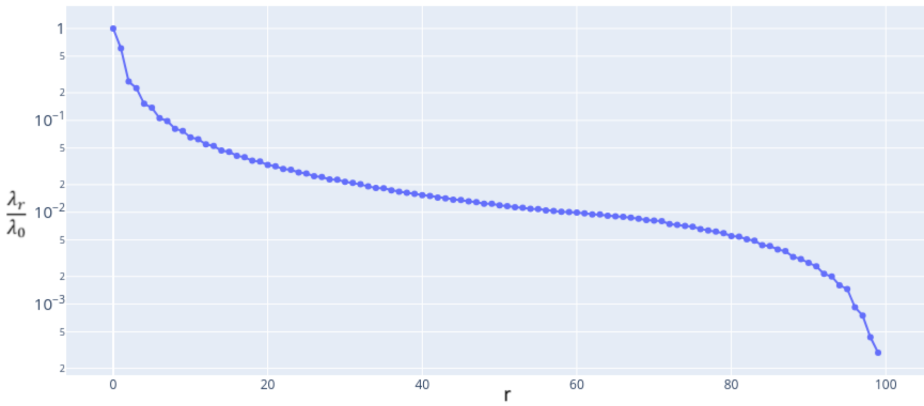

Figure 4.4: Normalized singular values from POD procedure with $\nu = 5 \times 10^{-4}$ and $r_{max} = 100$.

ROM construction To construct the POD basis functions, 200 snapshots of the velocity field are collected over the whole time interval $[0, 1]$. In this case, rather than employing the common centering trajectory approach, we directly apply the method of snapshots to the snapshot data since we did not use a proper FEM solution. The resulting POD basis has a dimension of 100. The first $r = 100$ POD modes capture more than 99.9% of the flow's energy both for $\nu = 10^{-3}$ and for $\nu = 5 \times 10^{-4}$, as shown in Figures 4.3 and 4.4, respectively. We deliberately take a large number of POD basis functions to validate the theoretical error bounds in Lemma 4.1.2 and Theorem 4.2.3 by using very accurate solutions. The POD modes are utilized to construct the ADL-ROM, which is analyzed over the entire time interval $[0, 1]$.

A second-order backward differentiation formula (BDF2) scheme is employed for temporal discretization of the ADL-ROM simulations with a time step size of $\Delta t = 10^{-3}$. The initial conditions for ADL-ROM solution are $\boldsymbol{u}_{-1}^r = \boldsymbol{u}_{-1}$ and $\boldsymbol{u}_0^r = \boldsymbol{u}_0$, where $\boldsymbol{u}_{-1}$ and $\boldsymbol{u}_0$ are the first two time steps of the analytical solution interpolated on the reduced space.

Parameters The inlet velocity, the dimensions of the domain and the viscosity values, i.e., $\nu = 10^{-3}$ and $\nu = 5 \times 10^{-4}$, lead to the following Reynolds numbers: $Re = 1000$ and $Re = 2000$.

One of the most important parameters of ROM-DF and AD operators is the radius of the ROM differential filter introduced in Section 2.1.1, δ. To validate our theoretical error bounds we employ different δ values as illustrated in Sections 4.3.2 and 4.3.3.

The other parameter of interest is the van Cittert order, N. For this case, we chose to keep the value of N fixed. In particular, after some preliminary tests we chose $N = 5$.

4.3.1 ADL-ROM Solutions

In this section, we present a detailed comparison of the exact and ADL-ROM solutions for the velocity field. The reduced solution is computed using $r = 99$ POD modes. For the approximate deconvolution process, we employ the van Cittert method with the order of approximation N set to 5.

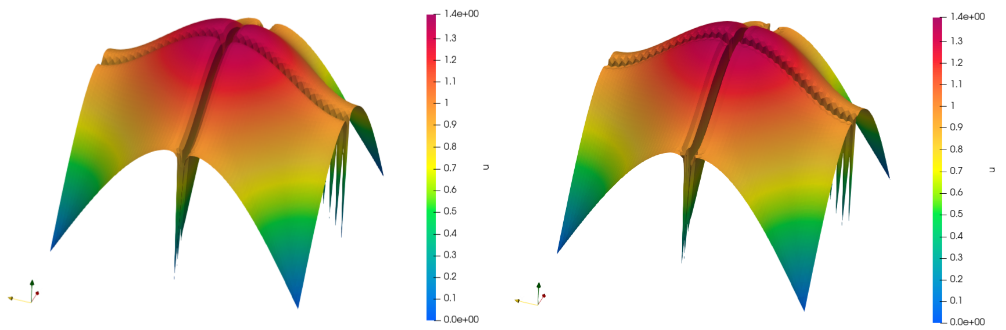

Figure 4.5: Exact solution (left) and ADL-ROM solution (right) at $t = 0.5$, for $\nu = 10^{-3}$, $r = 99$, $\delta = 6.25 \times 10^{-2}$, and $N = 5$. Color and position are determined by the magnitude of the velocity field.

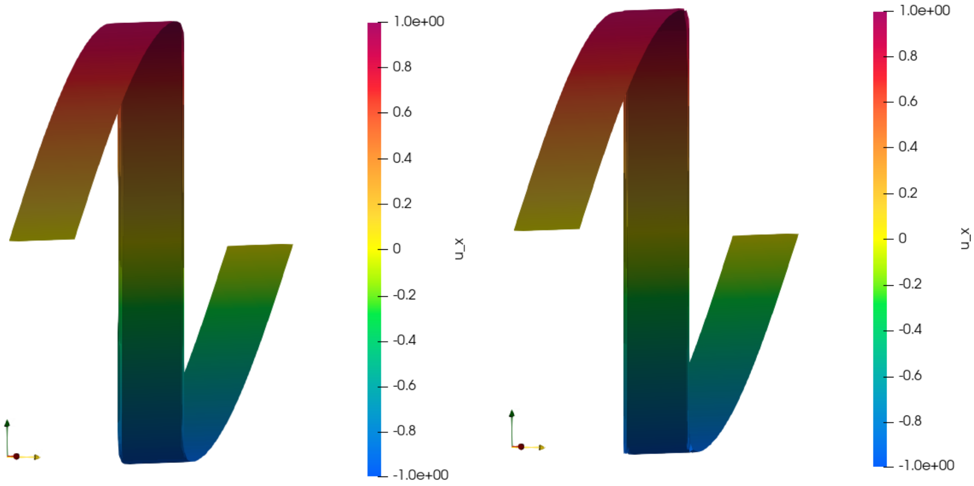

Figure 4.6: Exact solution (left) and ADL-ROM solution (right) at $t = 0.5$, for $\nu = 10^{-3}$, $r = 99$, $\delta = 6.25 \times 10^{-2}$, and $N = 5$. Color and position are determined by the x-component of the velocity field.

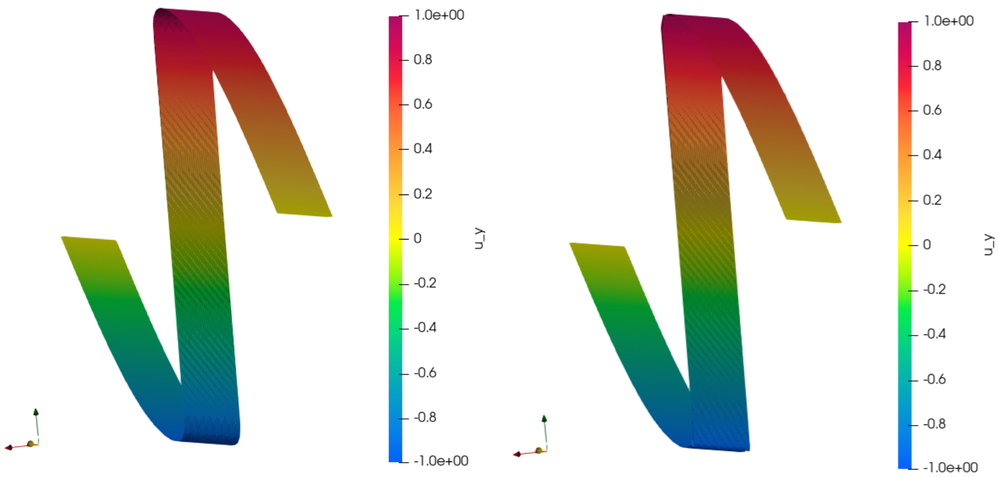

Figure 4.7: Exact solution (left) and ADL-ROM solution (right) at $t = 0.5$, for $\nu = 10^{-3}$, $r = 99$, $\delta = 6.25 \times 10^{-2}$, and $N = 5$. Color and position are determined by the y-component of the velocity field.

Figure 4.5 illustrates the solutions obtained at $t = 0.5$ with the viscosity parameter set to $\nu = 10^{-3}$ and the filter radius δ chosen as 6.25×10^{-2}. From these plots, two key points emerge:

1. The foremost challenge in this scenario is the accurate reproduction of the wave propagation present in the exact solution. This difficulty is

especially apparent in Figures 4.6 and 4.7, where the dynamics of the wave are intricately captured.

2. Despite the inherent challenges, the ADL-ROM solution demonstrates a commendable ability to approximate the exact solution, thereby validating the efficacy of the van Cittert-based approach within this ADL-ROM framework. This is particularly evident in the left-hand plot of Figure 4.11.

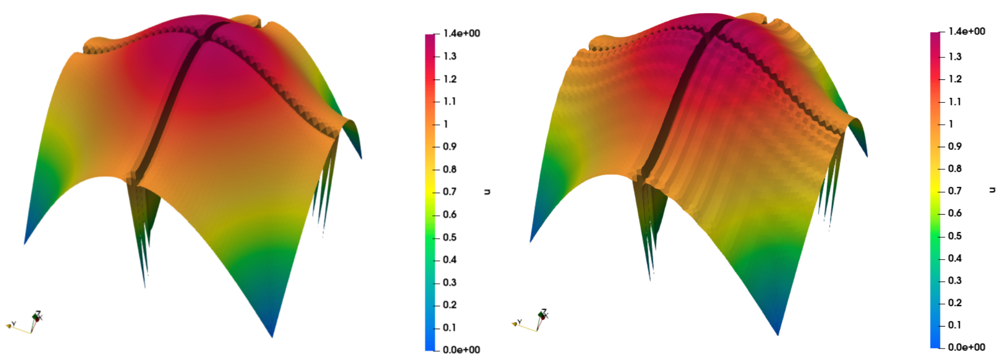

Figure 4.8: Exact solution (left) and ADL-ROM solution (right) at $t = 0.5$, for $\nu = 5 \times 10^{-4}$, $r = 99$, $\delta = 1.25 \times 10^{-1}$, and $N = 5$. Color and position are determined by the magnitude of the velocity field.

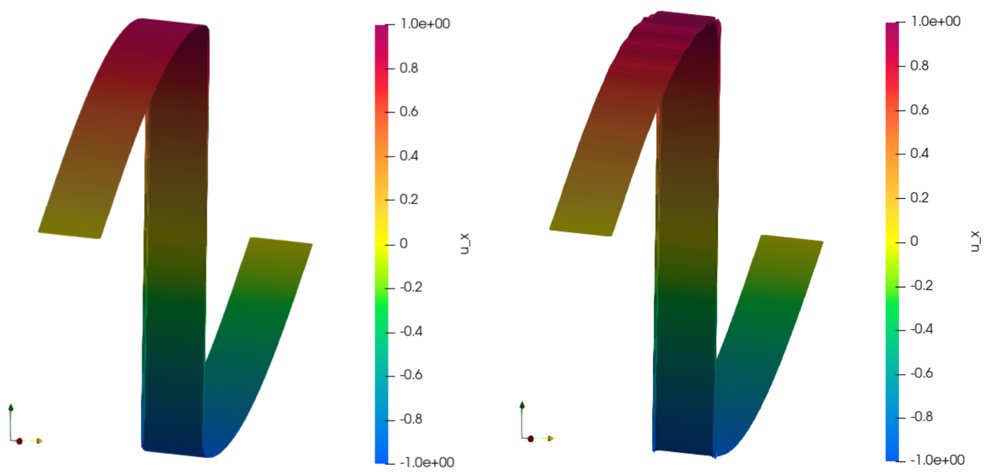

Figure 4.9: Exact solution (left) and ADL-ROM solution (right) at $t = 0.5$, for $\nu = 5 \times 10^{-4}$, $r = 99$, $\delta = 1.25 \times 10^{-1}$, and $N = 5$. Color and position are determined by the x-component of the velocity field.

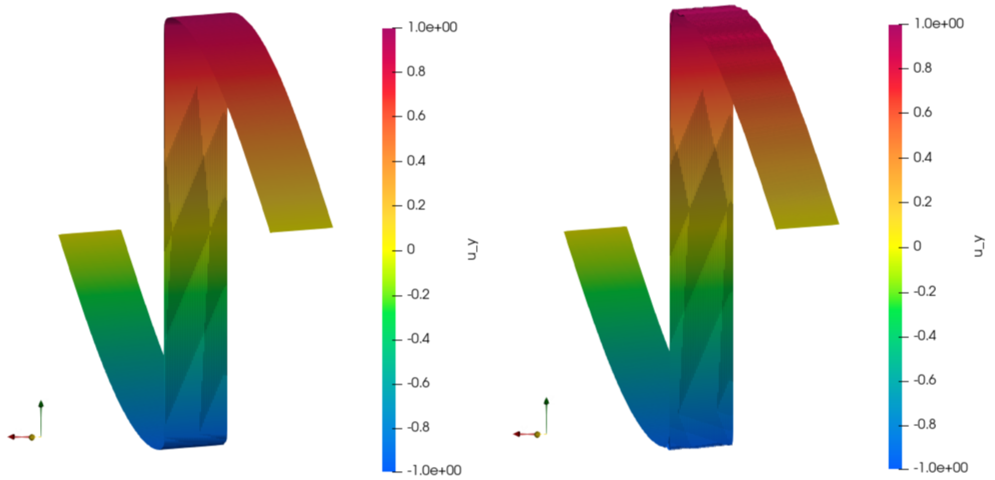

Figure 4.10: Exact solution (left) and ADL-ROM solution (right) at $t = 0.5$, for $\nu = 5 \times 10^{-4}$, $r = 99$, $\delta = 1.25 \times 10^{-1}$, and $N = 5$. Color and position are determined by the y-component of the velocity field.

Similarly, Figure 4.8 depicts the solutions at $t = 0.5$ for a different parameter setting, i.e., the viscosity is reduced to $\nu = 5 \times 10^{-4}$, and the filter radius is increased to $\delta = 1.25 \times 10^{-1}$. In this case, the same two crucial observations from the previous configuration remain valid:

1. Accurately capturing the wave propagation remains a primary difficulty, as clearly illustrated in Figures 4.9 and 4.10.

2. The ADL-ROM solution continues to provide a reliable approximation of the exact solution, even for this reduced viscosity value. This is particularly evident in the right-hand plot of Figure 4.11, which showcases the accuracy and robustness of the method across varying parameters.

Through these numerical results, we observe the strengths of the ADL-ROM framework in approximating the velocity field of the exact solution under different parameter configurations. These findings underline the potential of the van Cittert-based ADL-ROM in addressing the complexities for fluid dynamics problems.

4.3.2 AD Rates of Convergence

In this section, we perform a numerical investigation of the van Cittert approximate deconvolution method itself, independently from the online running of the ROM. Specifically, we investigate numerically how $\|\boldsymbol{u}_k - D_N^r \overline{\boldsymbol{u}_k}\|$

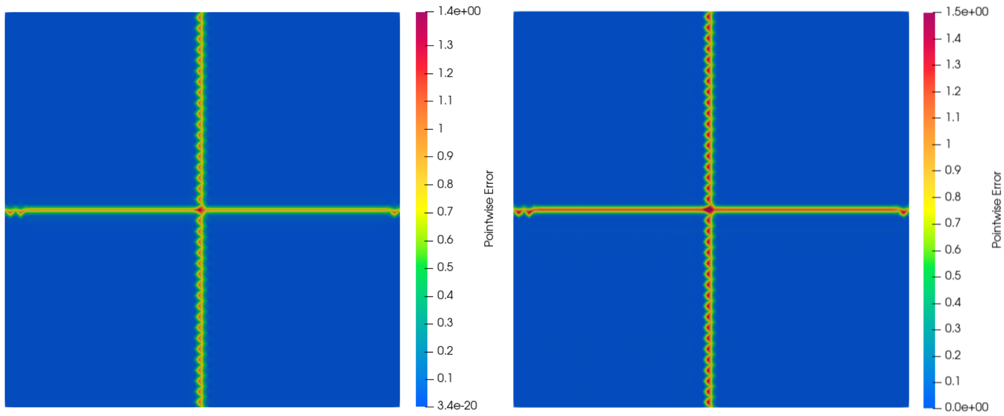

Figure 4.11: At $t = 0.5$, for $N = 5$ and $r = 99$ Pointwise absolute error between the exact and the ADL-ROM solutions for $\nu = 10^{-3}$ and $\delta = 6.25 \times 10^{-2}$ (left) and for $\nu = 5 \times 10^{-4}$ and $\delta = 1.25 \times 10^{-1}$ (right).

scales with respect to the filter radius, δ. We recall that $D_N^r \overline{\boldsymbol{u}_k}$ is the interpolated exact solution at time t_k after the van Cittert AD operator in (3.1.3) and the filter in (2.1.8) are applied to it. To this end, we note that a theoretical scaling is provided by the *a priori* error bound in Lemma 4.1.2, which, for completeness, we include below

$$\begin{aligned}\|\boldsymbol{u}_k - D_N^r \overline{\boldsymbol{u}_k}\|^2 \leq C(N) &\left(h^{2m+4} + \Delta t^4 + \sum_{j=r+1}^{R} \lambda_j \right) \\ &+ C(N)\delta^2 \Big(h^{2m+2} + \|S^r\|_2 \; h^{2m+4} + (1 + \|S^r\|_2)\Delta t^4 \\ &+ \sum_{j=r+1}^{R} \|\nabla \boldsymbol{\varphi}_j\|^2 \lambda_j \Big) + C(N)\delta^4 \|\Delta \boldsymbol{u}_k\|^2. \end{aligned} \tag{4.3.4}$$

To investigate how $\|\boldsymbol{u}_k - D_N^r \overline{\boldsymbol{u}_k}\|$ scales with respect to the filter parameter δ, we make the other sources of error (i.e., those associated with the finite element and ROM truncation errors) small compared to those associated with δ.

Our primary sources of error in Lemma 4.1.2 depend on h, Δt, δ, and the ROM truncation errors $\Lambda_{L^2}^r = \sum_{j=r+1}^{R} \lambda_j$ and $\Lambda_{H^1}^r = \sum_{j=r+1}^{R} \|\nabla \boldsymbol{\varphi}_j\|^2 \lambda_j$. Because our snapshots are projections of the analytical solution on the finite element space, the approximation error is independent of the time step, Δt. Fixing $h = 1/64$, $m = 2$ since we use the Taylor-Hood element, and $r = 99$, we have that $h^{2m} = \mathcal{O}(10^{-8})$, $\Lambda_{L^2}^r = \mathcal{O}(10^{-5})$, $\Lambda_{H^1}^r = \mathcal{O}(10^{0})$, and $\|S_r\|_2 = \mathcal{O}(10^{5})$.

δ	$E_{L^2}^{AD}$
1.00×10^{-1}	8.02×10^{-2}
8.20×10^{-2}	5.78×10^{-2}
6.40×10^{-2}	3.82×10^{-2}
4.60×10^{-2}	1.87×10^{-2}
2.80×10^{-2}	7.06×10^{-3}
1.00×10^{-2}	9.12×10^{-4}

Table 4.1: AD approximation error $E_{L^2}^{AD}$ for decreasing δ values and for $\nu = 10^{-3}$.

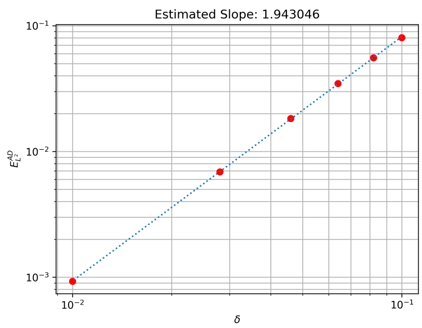

Figure 4.12: Linear regression of $E_{L^2}^{AD}$ with respect to δ for $\nu = 10^{-3}$.

The *a priori* error bound (4.3.4) suggests the scaling $\|\boldsymbol{u}_k - D_N^r \overline{\boldsymbol{u}_k}\| = \mathcal{O}(\delta^2)$ when all other sources of error are small. To investigate numerically this theoretical scaling, we use the following error metric, with $\boldsymbol{u}_K^h$ referring to the FE projection of the known solution at the final snapshot used for POD:

$$E_{L^2}^{AD} = \|\boldsymbol{u}_K^h - D_N^r \overline{\boldsymbol{u}_K^h}\|. \tag{4.3.5}$$

Results for ν=10^{-3}

In Figure 4.12, we validate the theoretical scaling $E_{L^2}^{AD} = \mathcal{O}(\delta^2)$ by examining the behavior of the L^2-norm error $E_{L^2}^{AD}$. Specifically, we compute and plot the error values for $r = 99$ POD modes. The filter radius δ is varied systematically within six evenly spaced values ranging between 10^{-2} and 10^{-1}. These filter radius values and the corresponding error data are detailed in Table 4.1, providing a clear numerical foundation for the theoretical expectations.

Results for ν=5×10^{-3}

Similarly to the previous section, in Figure 4.13, we further validate the theoretical scaling $E_{L^2}^{AD} = \mathcal{O}(\delta^2)$ by plotting the error $E_{L^2}^{AD}$ for the same configuration as before, i.e., with $r = 99$. In this case, the filter radius δ is chosen from six evenly spaced values within the range from 10^{-1} to 4×10^{-1}. The corresponding values of δ and the calculated error data are comprehensively documented in Table 4.2.

δ	$E_{L^2}^{AD}$
4.00×10^{-1}	5.49×10^{-2}
3.4×10^{-1}	2.45×10^{-2}
2.80×10^{-1}	8.82×10^{-3}
2.22×10^{-1}	4.62×10^{-3}
1.60×10^{-1}	3.82×10^{-3}
1.00×10^{-1}	3.12×10^{-3}

Table 4.2: AD approximation error $E_{L^2}^{AD}$ for decreasing δ values and for $\nu = 5 \times 10^{-4}$.

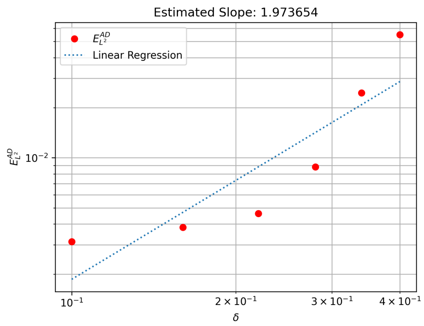

Figure 4.13: Linear regression of $E_{L^2}^{AD}$ with respect to δ for $\nu = 5 \times 10^{-4}$.

Through all these plots and tabulated results, we demonstrate that the numerical experiments are consistent with the theoretical prediction, highlighting the quadratic dependence of the error on the filter radius δ. These findings reinforce the reliability of the AD operators in the ROM framework and their alignment with the underlying theoretical analysis.

4.3.3 ADL-ROM Rates of Convergence

In this section, we perform a numerical investigation of the ADL-ROM *a priori* error bounds (4.2.6)–(4.2.7) in Theorem 4.2.3. The ADL-ROM approximation error at the final time step is $E_{L^2}^{ADL-ROM} = \left\| \boldsymbol{u}_K^h - \boldsymbol{u}_K^r \right\|$, where $\boldsymbol{u}_K^r$ is the ADL-ROM solution at the final time step. We numerically investigate the rates of convergence of $E_{L^2}^{ADL-ROM}$ with respect to the filter radius δ and the ROM truncation error $\Lambda_{H^1}^r = \sum_{j=r+1}^{R} \|\nabla \boldsymbol{\varphi}_j\|^2 \lambda_j$.

To determine the ADL-ROM approximation error rate of convergence with respect to δ, we fix $h = \frac{1}{64}$, $r = 99$, $\Delta t = 10^{-3}$, $N = 5$, and vary δ. With these choices, $h^{2m} = \mathcal{O}\left(10^{-8}\right)$, $\Delta t^4 = \mathcal{O}\left(10^{-12}\right)$, $\Lambda_{L^2}^r = \mathcal{O}\left(10^{-5}\right)$, $\Lambda_{H^1}^r = \mathcal{O}\left(10^{0}\right)$, and $\|S^r\|_2 = \mathcal{O}\left(10^{5}\right)$. Thus, the theoretical ADL-ROM *a priori* error bound predicts the following rate of convergence of $E_{L^2}^{ADL-ROM}$ with respect to δ:

$$E_{L^2}^{ADL-ROM} = \mathcal{O}\left(\delta^2\right). \tag{4.3.6}$$

δ	$E_{L^2}^{ADL-ROM}$
6.50×10^{-1}	2.66×10^{-1}
5.00×10^{-1}	8.72×10^{-2}
2.50×10^{-1}	4.74×10^{-2}
1.80×10^{-1}	2.44×10^{-2}
1.25×10^{-1}	8.05×10^{-3}
6.25×10^{-2}	2.06×10^{-3}

Table 4.3: ADL-ROM approximation error $E_{L^2}^{ADL-ROM}$ for decreasing δ values and for $\nu = 10^{-3}$.

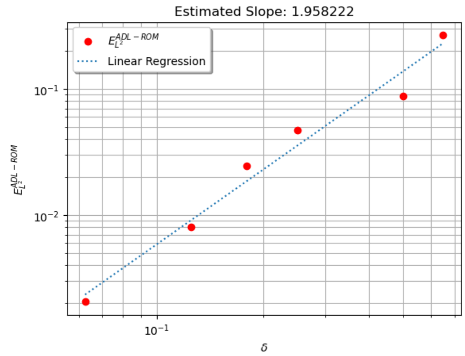

Figure 4.14: Linear regression of $E_{L^2}^{ADL-ROM}$ with respect to δ for $\nu = 10^{-3}$.

To determine the ADL-ROM approximation error rate of convergence with respect to $\Lambda_{H^1}^r$, we fix $h = \frac{1}{64}$, $\Delta t = 10^{-3}$, $\delta = 6.25 \times 10^{-2}$, $N = 5$, and vary r. With these choices, $h^{2m} = \mathcal{O}\left(10^{-8}\right)$ and $\|S^r\|_2 = \mathcal{O}\left(10^{2}\right) - \mathcal{O}\left(10^{5}\right)$. Thus, the theoretical ADL-ROM approximation error bound predicts the following rate of convergence of $E_{L^2}^{ADL-ROM}$ with respect to $\Lambda_{H^1}^r$:

$$E_{L^2}^{ADL-ROM} = \mathcal{O}\left(\Lambda_{H^1}^r\right). \tag{4.3.7}$$

Results for ν=10^{-3}

The ADL-ROM approximation error $E_{L^2}^{ADL-ROM}$ is listed in Table 4.5 for decreasing δ values. The corresponding linear regression, which is shown in Figure 4.16, indicates the following ADL-ROM approximation error rate of convergence with respect to δ:

$$E_{L^2}^{ADL-ROM} = \mathcal{O}\left(\delta^{1.96}\right). \tag{4.3.8}$$

Thus, the theoretical rate of convergence is numerically recovered.

The ADL-ROM approximation error $E_{L^2}^{ADL-ROM}$ is listed in Table 4.6 for increasing r values. The corresponding linear regression, which is shown in Figure 4.17, indicates the following ADL-ROM approximation error rate of convergence with respect to $\Lambda_{H^1}^r$:

r	$\Lambda^r_{H^1}$	$E^{ADL-ROM}_{L^2}$
10	2.14×10^4	9.55×10^{-2}
20	1.71×10^4	4.98×10^{-2}
30	1.34×10^4	3.37×10^{-2}
40	1.02×10^4	2.34×10^{-2}
50	7.36×10^3	1.83×10^{-2}

Table 4.4: ADL-ROM approximation error $E^{ADL-ROM}_{L^2}$ for increasing r values and for $\nu = 10^{-3}$.

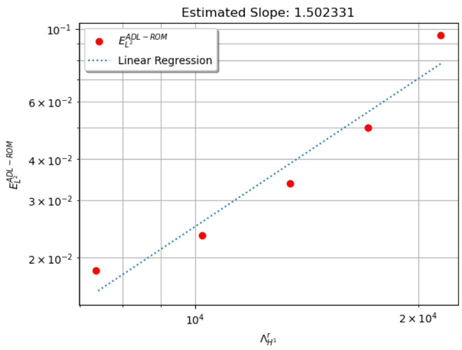

Figure 4.15: Linear regression of $E^{ADL-ROM}_{L^2}$ with respect to $\Lambda^r_{H^1}$ for $\nu = 10^{-3}$.

$$E^{ADL-ROM}_{L^2} = \mathcal{O}\left(\left(\Lambda^r_{H^1}\right)^{1.50}\right).$$

Thus, the theoretical rate of convergence is numerically recovered. We note that the rate of convergence displayed by the numerical results in Figure 4.17 (i.e., 1.50) is higher than the theoretical rate of convergence in (4.3.7). We believe that this higher rate of convergence in Figure 4.17 is in part due to the outlier data point in the top right corner of the plot, which corresponds to the largest value of $\Lambda^r_{H^1}$. This r value yields the lowest ROM resolution results, which are less accurate than the results corresponding to the other r values.

Results for ν=5 $\times$ 10^{-3}

The ADL-ROM approximation error $E^{ADL-ROM}_{L^2}$ is listed in Table 4.5 for decreasing δ values. The corresponding linear regression, which is shown in Figure 4.16, indicates the following ADL-ROM approximation error rate of convergence with respect to δ:

$$E^{ADL-ROM}_{L^2} = \mathcal{O}\left(\delta^{2.32}\right). \tag{4.3.9}$$

Thus, the theoretical rate of convergence is numerically recovered.

The ADL-ROM approximation error $E^{ADL-ROM}_{L^2}$ is listed in Table 4.6 for increasing r values. The corresponding linear regression, which is shown in Figure 4.17, indicates the following ADL-ROM approximation error rate of

δ	$E_{L^2}^{ADL-ROM}$
3.00×10^{-1}	5.37×10^{-2}
2.50×10^{-1}	4.73×10^{-2}
2.00×10^{-1}	3.08×10^{-2}
1.80×10^{-1}	2.29×10^{-2}
1.50×10^{-1}	1.23×10^{-2}
1.25×10^{-1}	7.64×10^{-3}

Table 4.5: ADL-ROM approximation error $E_{L^2}^{ADL-ROM}$ for decreasing δ values and for $\nu = 5 \times 10^{-4}$.

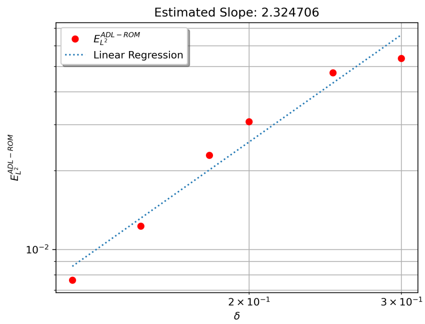

Figure 4.16: Linear regression of $E_{L^2}^{ADL-ROM}$ with respect to δ for $\nu = 5 \times 10^{-4}$.

r	$\Lambda_{H^1}^r$	$E_{L^2}^{ADL-ROM}$
5	3.66×10^4	1.28×10^{-1}
10	3.36×10^4	1.42×10^{-1}
20	2.81×10^4	6.30×10^{-2}
30	2.28×10^4	4.31×10^{-2}
40	1.78×10^4	2.80×10^{-2}
50	1.28×10^4	2.02×10^{-2}

Table 4.6: ADL-ROM approximation error $E_{L^2}^{ADL-ROM}$ for increasing r values and for $\nu = 5 \times 10^{-4}$.

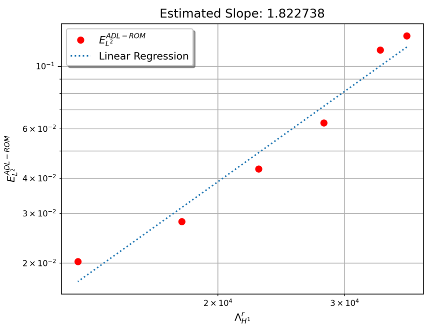

Figure 4.17: Linear regression of $E_{L^2}^{ADL-ROM}$ with respect to $\Lambda_{H^1}^r$ for $\nu = 5 \times 10^{-4}$.

convergence with respect to $\Lambda_{H^1}^r$:

$$E_{L^2}^{ADL-ROM} = \mathcal{O}\left(\left(\Lambda_{H^1}^r\right)^{1.82}\right).$$

Thus, the theoretical rate of convergence is numerically recovered. We note again that the rate of convergence displayed by the numerical results in Figure 4.17 (i.e., 1.82) is higher than the theoretical rate of convergence in (4.3.7). Also in this case, we believe that this higher rate of convergence in Figure 4.17 is in part due to the two outlier data points in the top right corner of the plot, which correspond to the largest values of $\Lambda^r_{H^1}$. These r values yield the lowest ROM resolution results, which are less accurate than the results corresponding to the other r values.

Chapter 5

Parametric Approximate Deconvolution Leray Model

In this chapter, we present and discuss a currently ongoing research project, aimed at further extending and evaluating the potentiality of the ADL-ROM methodology in a parametric setting.

More specifically, in Chapter 3, we provided a comprehensive and detailed exposition of the construction process underlying our newly proposed Regularized Reduced Order Model (Reg-ROM), referred to as the ADL-ROM. This development was carried out within the framework of model reduction performed exclusively in the time domain. In contrast, the present chapter is dedicated to a numerical investigation of the ADL-ROM in a scenario where model reduction is not limited to the time domain, as extensively examined in Chapters 3 and 4, but is also extended to the parameter domain. In particular, we introduce parametric dependence by considering variations in the viscosity parameter ν.

The organization of this chapter is as follows:

- In Section 5.1, we provide an overview of how model reduction can be performed simultaneously in both the time and parameter domains. In this context, we introduce an additional Proper Orthogonal Decomposition (POD) technique, namely the nested-POD (n-POD), which is specifically designed to handle such a double reduction framework.

- In Section 5.2, we develop and present the appropriate formulation of the ADL-ROM when model reduction is applied not only in the time domain but also in the parameter domain. This formulation accounts for the parametric dependence introduced via the viscosity coefficient ν.

- Finally, in Section 5.3, we present a set of preliminary numerical results aimed at assessing the accuracy of the ADL-ROM in reconstructing reference solutions, particularly in its parametric variant. To conduct this analysis, we specifically focus on the van Cittert AD operator.

It is important to emphasize once again that the present study represents an initial step in exploring the parametric version of ADL-ROM. Since our research in this direction is still in progress, further refinements and extensive investigations are required before arriving at a definitive and complete understanding of the topic.

5.1 Model Reduction: With Respect to Time and Viscosity

We consider the incompressible Navier-Stokes equations (NSE) in (1.1.1)-(1.1.2) with a variable kinematic viscosity ν, which is assumed to lie within a predefined range

$$\nu \in [\nu_{\min}, \nu_{\max}] \subset \mathbb{R}^+.$$

A more detailed discussion regarding the choice of the parameter and its range will be provided in Section 5.3.

Performing model reduction simultaneously in both the time domain and the parameter domain presents significant computational efforts. A standard proper orthogonal decomposition (POD) approach that compresses data concurrently in time and across the parameter space would require substantial computational resources. Consequently, to circumvent the high computational cost associated with this brute force POD approach, we employ an alternative and more efficient strategy: the nested-POD (n-POD) algorithm. This hierarchical compression technique is widely recognized in the reduced order modeling community and is often referred to by various names in the literature (see, e.g., [5, 8, 17, 45]).

The n-POD algorithm performs data compression through a sequential two-stage process, initially targeting the temporal domain before addressing the parameter space. Specifically, in the first stage, each time trajectory associated with the selected parameter undergoes compression. Subsequently, in the second stage, a POD-based reduction is applied to the already time compressed parametric solutions. By systematically separating the time domain reduction from the parameter domain reduction, the n-POD algorithm achieves notable savings in both computational time and storage requirements when compared to the conventional monolithic POD approach. In our numerical study, we adopt the n-POD algorithm as formulated in [50], which is structured into two decoupled levels:

1. *Initial Compression of the Time Trajectories:*
    - In the first stage, a set of training parameters $\{\nu_i\}_{i=1}^{r_\nu}$ is selected within the given parameter space.
    - For each selected parameter value ν_i, a standard POD procedure is applied in time, retaining only the first r^t dominant modes corresponding to the velocity field.
    - These POD modes, scaled by their associated singular values (as outlined in Section 1.3.1), are denoted as $\boldsymbol{\varphi}_{u(\nu_i),j}$ for $j = 1, \ldots, r^t$.
2. *Global Compression of the Scaled Modes:*
    - In the second stage, a global POD procedure is employed to further compress the previously obtained time-reduced modes across the parametric domain.
    - The final reduced space for the velocity field is constructed as

$$\boldsymbol{X}^r \doteq \text{POD}\left(\left\{\boldsymbol{\varphi}_{\boldsymbol{u}(\nu_1),j}\right\}_{j=1}^{r^t}, \cdots, \left\{\boldsymbol{\varphi}_{\boldsymbol{u}(\nu_{r_\nu}),j}\right\}_{j=1}^{r^t}\right) \tag{5.1.1}$$

For the sake of readability, we introduce the following compact notation for the reduced space associated with the ROM velocity field

$$\boldsymbol{X}^r \doteq \text{n-POD}\left(\left\{\boldsymbol{u}\left(\nu_i\right)\right\}_{i=1}^{r_\nu}; r^t\right). \tag{5.1.2}$$

Here, r^t corresponds to the dimension of the time-reduced basis obtained in the first compression stage, while r denotes the final reduced space dimension for the velocity field.

Consistently with the approach adopted throughout this thesis, we do not explicitly consider the pressure term in the reduced model. This decision is justified since the POD modes associated with the velocity field are inherently solenoidal and, by construction, satisfy the prescribed boundary conditions. Nevertheless, should a pressure approximation be required, the nested-POD procedure can be smoothly extended to alternative ROM formulations, such as those incorporating supremizer enrichment strategies.

Through the application of the n-POD algorithm, we achieve a substantial reduction in both computational time and storage requirements when compared to the conventional monolithic POD approach. Moreover, the reduced basis functions obtained through the n-POD are comparable with those that could be obtained through the conventional POD. Thus, these two characteristics make the n-POD method a compelling choice for parametric model reduction in complex fluid flow problems.

5.2 Parametric ADL-ROM

In Algorithm 4, we show the pseudocode for the Approximate Deconvolution Leray ROM approach coupled with the n-POD algorithm. As it is easily noticeable from the comparison with Algorithm 3, the main difference between them is an additional loop on the parameter domain.

Algorithm 4 ADL-ROM Pseudocode

$\boldsymbol{u}_{-1}$, $\boldsymbol{u}_0$, $\boldsymbol{u}_{in}$, r, r^t, r_ν ▷ Inputs needed
for $i = 1, \dots, r_\nu$ **do** ▷ Parameter loop
  for $k = 1, \dots, K-1$ **do** ▷ Time loop
    Perform FOM simulation to compute $\boldsymbol{u}^h_{k+1}(\nu_i)$
  end for
  $\{\boldsymbol{u}^h_j(\nu_i)\}_{j=1}^r \subseteq \{\boldsymbol{u}^h_k(\nu_i)\}_{k=1}^K$ ▷ Snapshot collection
end for
Compute $\boldsymbol{X}^r \doteq$ n-POD $\left(\{\boldsymbol{u}^h(\nu_i)\}_{i=1}^{r_\nu}; r^t\right)$ ▷ POD basis generation
for $k = 1, \dots, K-1$ **do** ▷ Time loop for ROM
  Solve (3.2.3) to compute $\boldsymbol{a}^{k+1}$, which requires:
    Solve (2.2.3) at each nonlinear iteration ▷ Leray filtering
    Solve one of (3.1.6), (3.1.11), or (3.1.15)
    at each nonlinear iteration ▷ Approximate Deconvolution
end for

5.3 Numerical Results

Goals In this section, we present the initial numerical investigation aimed at evaluating the performance of the Approximate Deconvolution Leray ROM when the model reduction is carried out in both the time and parameter domains. This analysis follows the methodologies detailed in Sections 5.1 and 5.2. The primary goal of this study is to assess whether the ADL-ROM solution can effectively approximate the underlying flow dynamics with a certain level of accuracy.

Given that this work represents an early stage in our research on the parametric extension of ADL-ROM, the results presented here primarily focus on analyzing the behavior of the ADL-ROM solution and computing various L^2 error metrics. A more detailed discussion regarding the evaluation criteria employed in this study is provided in the *Criteria* paragraph later in this section.

To maintain coherence with the framework established in the previous chapter, our numerical experiments are exclusively based on the van Cittert approximate deconvolution (AD) methodology, which is thoroughly described in Section 3.1.1. Consequently, any reference to ADL-ROM within the context of the numerical results presented in this section should be understood as ADL-ROM specifically constructed using the van Cittert regularization approach.

Furthermore, it is essential to emphasize that, in line with prior sections, all methodologies under consideration are applied to the Navier-Stokes Equations (NSE), as formulated in Equations (1.1.1) and (1.1.2).

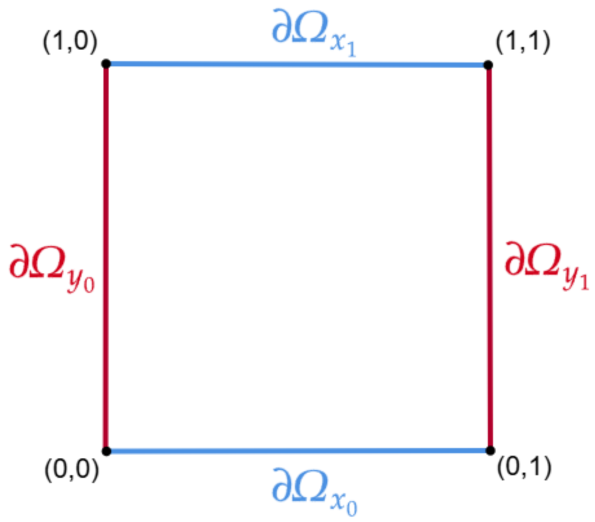

Figure 5.1: The computational domain, Ω.

Computational setting For our numerical investigation, we adopt a similar test problem and computational setup as those described in Section 4.3. The problem involves the 2D incompressible fluid flow with a known analytical solution.

Specifically, the exact velocity, denoted by $\boldsymbol{u} = (u, v)$, is defined by

$$u = \frac{2}{\pi}\sin(\pi y)\arctan\left(-\frac{1}{2\nu}\left(y - \sin^2\left(\frac{\pi}{2}t\right)\right)\right) \tag{5.3.1}$$

and

$$v = \frac{2}{\pi} \sin(\pi x) \arctan\left(-\frac{1}{2\nu}\left(x - \sin^2\left(\frac{\pi}{2}t\right)\right)\right) \tag{5.3.2}$$

Additionally, the exact pressure is constant, given by $p = 0$.

In our numerical investigation, we consider the parametric domain $\nu \in \left[10^{-3}, 2.5 \times 10^{-3}\right]$, and then select the forcing term to match the exact solution.

We consider the motion of the incompressible flow within the computational domain $\Omega \doteq [0,1] \times [0,1]$, illustrated in Figure 5.1.

The exact solution is subject to the following boundary conditions:

$$\begin{cases} u = 0 & \text{on } \partial\Omega_{x_0} \text{ and } \partial\Omega_{x_1}, \\ v = 0 & \text{on } \partial\Omega_{y_0} \text{ and } \partial\Omega_{y_1}, \\ \frac{\partial v}{\partial n} = 0 & \text{on } \partial\Omega_{x_0} \text{ and } \partial\Omega_{x_1}, \\ \frac{\partial u}{\partial n} = 0 & \text{on } \partial\Omega_{y_0} \text{ and } \partial\Omega_{y_1}. \end{cases} \tag{5.3.3}$$

Since the snapshots used in the ROM framework are directly derived from the exact solution by interpolating it onto the finite element mesh, as will be elaborated later, they inherently satisfy the homogeneous Dirichlet and Neumann boundary conditions given in Equation (5.3.3). Consequently, at the ROM level, it is unnecessary to explicitly enforce these boundary conditions, as the POD basis functions naturally adhere to them. As a result, the ROM solution inherently preserves the homogeneous Dirichlet and Neumann boundary conditions (5.3.3) without any additional constraints.

Snapshot generation The numerical tests are performed on a triangular uniform mesh with $h = \frac{1}{64}$ (see Figure 5.2). Spatial discretization is achieved using the inf-sup stable Taylor-Hood $\mathbb{P}^2 - \mathbb{P}^1$ finite element pair for velocity and pressure, respectively. This discretization results in a finite element space with a total dimension of $D \doteq D_h \doteq D_h^u + D_h^p = 38578 + 4887 = 43465$. We gather the snapshots over the time interval $[0, 1]$ at intervals of $\Delta t = 10^{-2}$, interpolating the analytical solution (u, v) on the finite element mesh.

ROM construction During the offline phase, we collect a total of 200 snapshots of the velocity field over the entire time interval $[0, 1]$. Unlike the commonly used centering trajectory approach, we opt for a direct application of the method of snapshots to the collected data. This choice is motivated by the fact that we did not employ a standard FEM solution, making the direct method more suitable for our study.

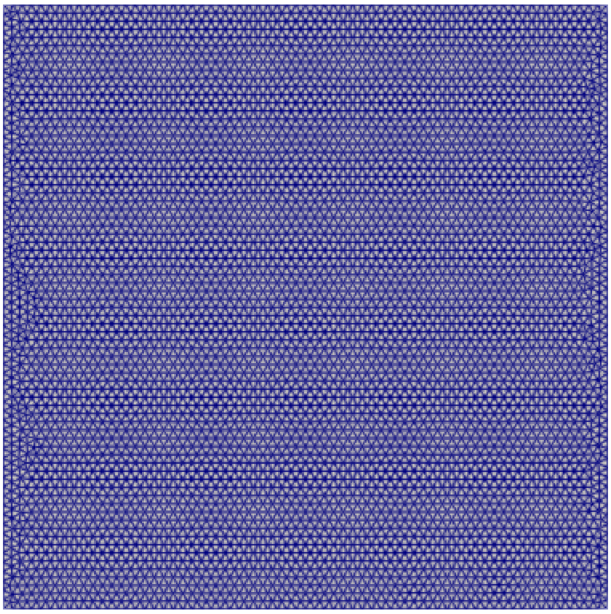

Figure 5.2: The FE mesh.

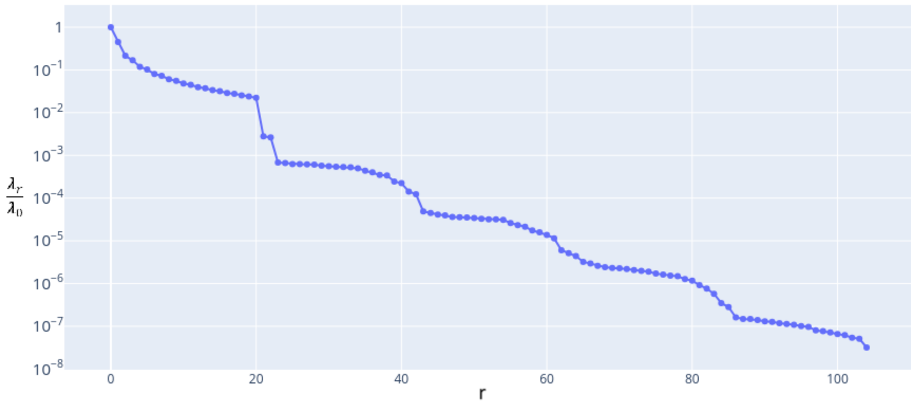

Figure 5.3: Normalized singular values from POD procedure with $\nu \in \left[10^{-3}, 2.5 \times 10^{-3}\right]$ and $r_{max} = 105$.

In the nested-POD (n-POD) algorithm, the first compression stage involves selecting a temporal basis truncation parameter, r^t, with a value of 21 for each sampled parametric instance. Additionally, we select $r_\nu = 5$ parameter values, equally distributed within the range $\left[10^{-3}, 2.5 \times 10^{-3}\right]$. The choice of r^t is made heuristically, striking a balance between achieving an accurate representation of the time evolution across different parametric snapshots and maintaining reasonable running times. Given that this study serves as an initial exploratory investigation, we prioritize computational feasibility alongside accuracy.

The decision to set $r_\nu = 5$ is also driven by the necessity to limit computational costs associated with the basis construction phase. We acknowledge that a more exhaustive investigation could likely identify an improved set of parameters that would yield more accurate reduced order approximations. Refining these parameters is one of the key aspects we aim to explore in the upcoming phases of our research.

In the second compression stage of the n-POD algorithm, we generate a POD basis with a total dimension of 105. The first $r_{max} = 105$ POD modes successfully capture more than 99.9% of the total energy of the flow field, as illustrated in Figure 5.3. We deliberately choose a relatively large number of POD basis functions to gain deeper insight into the behavior of the parametric version of the ADL-ROM. However, for the actual construction of the ADL-ROM, we employ only the first $r = 50$ POD modes, ensuring that the reduced order model remains computationally feasible while preserving a certain degree of accuracy. The resulting ADL-ROM is then analyzed over the full time interval $[0, 1]$.

To discretize the ADL-ROM in time, we utilize a second-order backward differentiation formula (BDF2) scheme, selecting a uniform time step size of $\Delta t = 10^{-3}$. The initial conditions for the ADL-ROM solution are given as $\boldsymbol{u}_{-1}^r = \boldsymbol{u}_{-1}$ and $\boldsymbol{u}_0^r = \boldsymbol{u}_0$, where $\boldsymbol{u}_{-1}$ and $\boldsymbol{u}_0$ correspond to the first two time steps of the analytical solution interpolated on the reduced space. This initialization ensures consistency between the ROM and the analytical solution.

Parameters The problem setup involves an inlet velocity, specific domain dimensions, and a viscosity parameter ν that varies within the range $\left[10^{-3}, 2.5 \times 10^{-3}\right]$. These physical conditions correspond to a Reynolds number range of $Re \in [400, 1000]$. However, instead of sampling the viscosity ν directly, in our setting we sample the parameter θ, defined as

$$\theta = -\frac{1}{2\nu},$$

which appears in the arctan function of the exact solution, as described in Equations (5.3.1)-(5.3.2).

By making this substitution, we redefine our parametric domain in terms of θ:

$$\theta \in [\theta_{\min}, \theta_{\max}] \doteq [-500, -200]. \tag{5.3.4}$$

This transformation simplifies our computational framework and allows for a more effective parameter sampling strategy. Our training set for constructing the n-POD basis is defined as

$$\boldsymbol{\theta}^{train} = [-500, -425, -350, -275, -200] \,.$$

For the numerical experiments, we also evaluate the ADL-ROM on a separate testing set, consisting of parameter values different from those used in the training phase. Since our focus is on maintaining computational feasibility during this preliminary investigation, we limit the number of test samples. The selected testing set is given by

$$\boldsymbol{\theta}^{test} = [-475, -400, -325, -250] \,.$$

A crucial parameter in the ROM differential filter (ROM-DF) and the AD operators, as introduced in Section 2.1.1, is the filter radius δ. For the purposes of this study, we adopt a value of $\delta = 0.1313$.

Another key parameter of interest is the van Cittert order, denoted by N. For this investigation, we opt to keep N fixed. Based on preliminary numerical tests and in alignment with the results obtained in the previous chapter, we select $N = 5$. This choice ensures stability and consistency in our computations.

Criteria We test the ADL-ROM accuracy by using the velocity field L^2 minimum, maximum, average, and final-time absolute and relative errors. The errors are defined as

- the L^2 minimum errors

$$\mathcal{E}_{\boldsymbol{u}}^{\min,a} \doteq \min_{j=1,\dots,K} \left\| \boldsymbol{u}(t_j) - \boldsymbol{u}_r(t_j) \right\|_{L^2(\Omega)} \tag{5.3.5}$$

and

$$\mathcal{E}_{\boldsymbol{u}}^{\min,r} \doteq \min_{j=1,\dots,K} \frac{\left\| \boldsymbol{u}(t_j) - \boldsymbol{u}_r(t_j) \right\|_{L^2(\Omega)}}{\left\| \boldsymbol{u}(t_j) \right\|_{L^2(\Omega)}}; \tag{5.3.6}$$

- the L^2 maximum errors

$$\mathcal{E}_{\boldsymbol{u}}^{\max,a} \doteq \max_{j=1,\dots,K} \left\| \boldsymbol{u}(t_j) - \boldsymbol{u}_r(t_j) \right\|_{L^2(\Omega)} \tag{5.3.7}$$

and

$$\mathcal{E}_{\boldsymbol{u}}^{\max,r} \doteq \max_{j=1,\dots,K} \frac{\left\| \boldsymbol{u}(t_j) - \boldsymbol{u}_r(t_j) \right\|_{L^2(\Omega)}}{\left\| \boldsymbol{u}(t_j) \right\|_{L^2(\Omega)}}; \tag{5.3.8}$$

- the L^2 average errors

$$\mathcal{E}_{\boldsymbol{u}}^{\text{ave},a} \doteq \frac{1}{K}\sum_{j=1}^{K} \left\|\boldsymbol{u}(t_j) - \boldsymbol{u}_r(t_j)\right\|_{L^2(\Omega)} \tag{5.3.9}$$

  and

$$\mathcal{E}_{\boldsymbol{u}}^{\text{ave},r} \doteq \frac{1}{K}\sum_{j=1}^{K} \frac{\left\|\boldsymbol{u}(t_j) - \boldsymbol{u}_r(t_j)\right\|_{L^2(\Omega)}}{\|\boldsymbol{u}(t_j)\|_{L^2(\Omega)}}; \tag{5.3.10}$$

- the L^2 final-time errors

$$\mathcal{E}_{\boldsymbol{u}}^{T,a} \doteq \left\|\boldsymbol{u}(T) - \boldsymbol{u}_r(T)\right\|_{L^2(\Omega)} \tag{5.3.11}$$

  and

$$\mathcal{E}_{\boldsymbol{u}}^{T,r} \doteq \frac{\left\|\boldsymbol{u}(T) - \boldsymbol{u}_r(T)\right\|_{L^2(\Omega)}}{\|\boldsymbol{u}(T)\|_{L^2(\Omega)}}, \tag{5.3.12}$$

where the a and r superscripts refer to the absolute and relative errors, respectively, $\boldsymbol{u}_r$ is the ADL-ROM solution, and the final time T is set to 1.

Numerical results To begin our analysis, we present a comparison of the exact and ADL-ROM solutions for the velocity field. The reduced solution is obtained using a reduced space composed of $r = 50$ POD modes. For the approximate deconvolution procedure, we employ the van Cittert method with an order of approximation of $N = 5$ and a filter radius parameter of $\delta = 0.1313$.

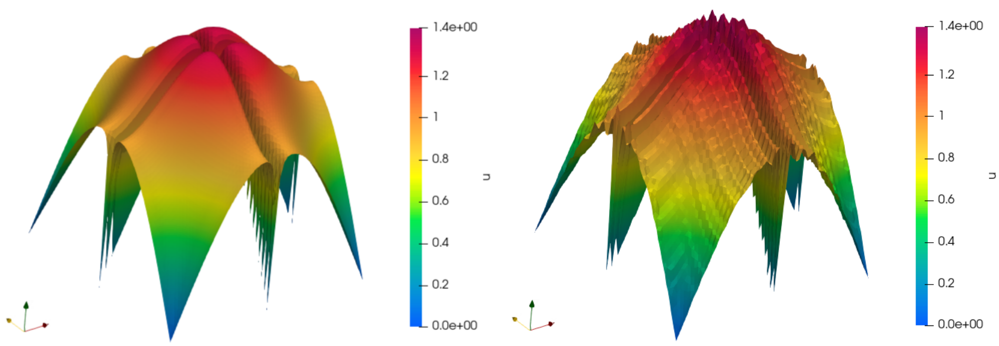

Figure 5.4: Exact solution (left) and ADL-ROM solution (right) at $t = 0.5$, for $\theta = -200$, $r = 50$, $\delta = 0.1313$, and $N = 5$. Color and position are determined by the magnitude of the velocity field.

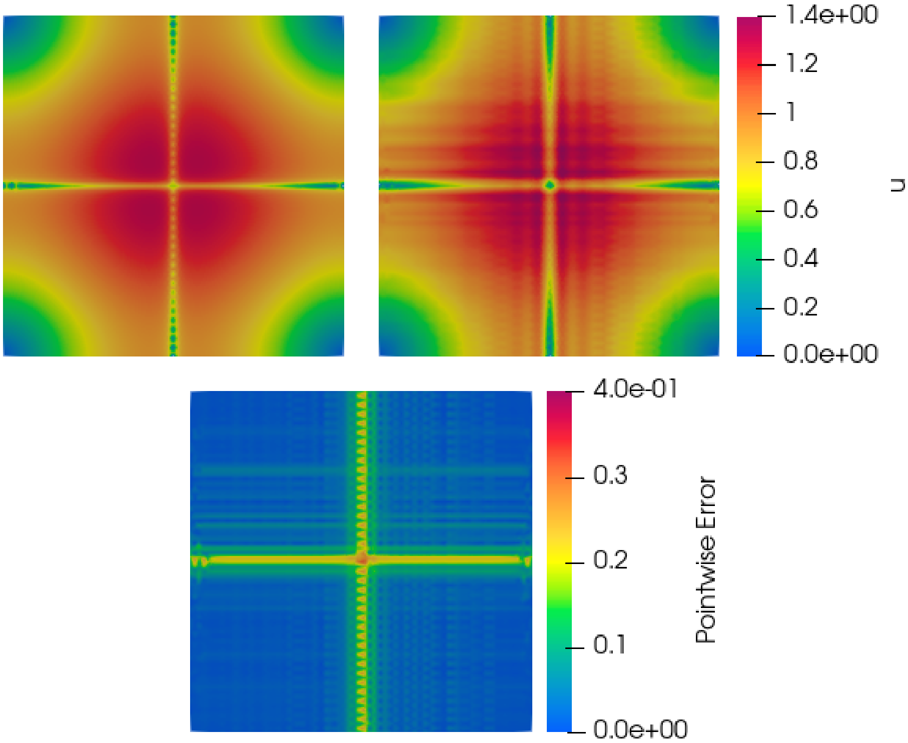

Figure 5.5: Exact solution (top left) and ADL-ROM solution (top right), and absolute pointwise error between the two solutions (bottom) at $t = 0.5$, for $\theta = -200$, $r = 50$, $\delta = 0.1313$, and $N = 5$.

Figures 5.4 and 5.5 illustrate the exact and the ADL-ROM solutions, in both three-dimensional (3D) and two-dimensional (2D) representations, at time $t = 0.5$ for $\theta = -200$, i.e., the viscosity is set to 2.5×10^{-3}, while the filter radius remains fixed at $\delta = 0.1313$.

From these plots, we observe that, as in the test case discussed in Section 4.3, the primary challenge of these numerical simulations lies in accurately capturing the propagation of waves observed in the exact solution. This behavior is particularly evident in Figure 5.4. Furthermore, in the parametric case with $r = 50$ POD modes, the ADL-ROM solution demonstrates a sufficient capability of approximating the exact solution when the selected parameter value belongs to the training set. Indeed, from Figure 5.5 we see that the error is primarily concentrated close to the discontinuity of the solution, while it is almost irrelevant in the rest of the computational domain. In particular, the absolute pointwise error is less than 7×10^{-2} on about the 89% of the computational domain.

In Tables 5.1, 5.2, 5.3, and 5.4, we present a comprehensive summary of

θ^{train}	$\mathcal{E}_{\boldsymbol{u}}^{\min,r}$
-200	7.67×10^{-3}
-275	7.82×10^{-3}
-350	7.94×10^{-3}
-425	8.04×10^{-3}
-500	8.12×10^{-3}

Table 5.1: Minimum relative L^2 errors for decreasing θ^{train} values and for $\delta = 0.1313$.

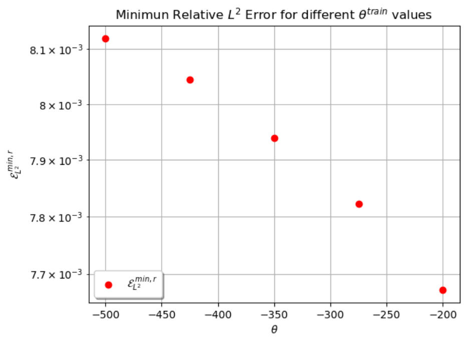

Figure 5.6: Behavior of $\mathcal{E}_{\boldsymbol{u}}^{\min,r}$ with respect to $\boldsymbol{\theta}^{train}$ for $\delta = 0.1313$.

θ^{train}	$\mathcal{E}_{\boldsymbol{u}}^{\max,r}$
-200	6.39×10^{-2}
-275	7.58×10^{-2}
-350	8.74×10^{-2}
-425	9.20×10^{-2}
-500	9.95×10^{-2}

Table 5.2: Maximum relative L^2 errors for decreasing θ^{train} values and for $\delta = 0.1313$.

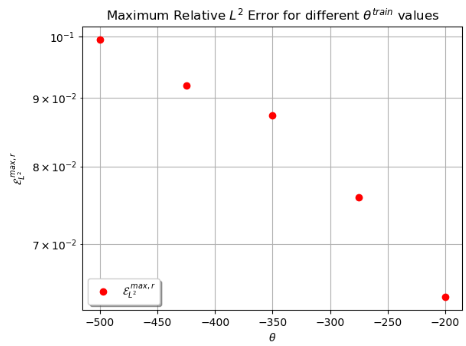

Figure 5.7: Behavior of $\mathcal{E}_{\boldsymbol{u}}^{\max,r}$ with respect to $\boldsymbol{\theta}^{train}$ for $\delta = 0.1313$.

θ^{train}	$\mathcal{E}_{\boldsymbol{u}}^{\text{ave},r}$
-200	3.99×10^{-2}
-275	4.62×10^{-2}
-350	5.05×10^{-2}
-425	5.36×10^{-2}
-500	5.60×10^{-2}

Table 5.3: Average relative L^2 errors for decreasing θ^{train} values and for $\delta = 0.1313$.

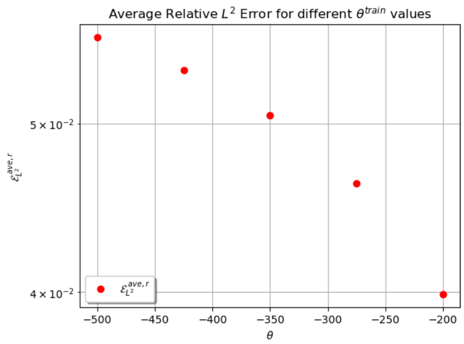

Figure 5.8: Behavior of $\mathcal{E}_{\boldsymbol{u}}^{\text{ave},r}$ with respect to $\boldsymbol{\theta}^{train}$ for $\delta = 0.1313$.

θ^{train}	$\mathcal{E}_{\boldsymbol{u}}^{T,r}$
-200	1.90×10^{-2}
-275	1.95×10^{-2}
-350	1.98×10^{-2}
-425	2.00×10^{-2}
-500	2.01×10^{-2}

Table 5.4: Final time relative L^2 errors for decreasing θ^{train} values and for $\delta = 0.1313$.

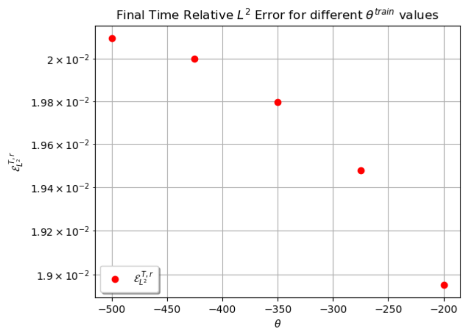

Figure 5.9: Behavior of $\mathcal{E}_{\boldsymbol{u}}^{T,r}$ with respect to $\boldsymbol{\theta}^{train}$ for $\delta = 0.1313$.

the relative L^2 errors associated with our analysis. Specifically, these tables report the minimum relative L^2 error, maximum relative L^2 error, average relative L^2 error, and the relative L^2 error at the final time for the five distinct values of the training parameter set $\boldsymbol{\theta}^{train}$.

By examining the data provided in these tables, we can observe a discernible trend: as the parameter θ decreases, which corresponds to a decrease in the viscosity, the relative L^2 errors exhibit a slight increase. This behavior aligns with our theoretical expectations, as a decrease in the viscosity leads to an increase in the Reynolds number. A higher Reynolds number, in turn, makes the problem more challenging to approximate accurately.

Despite this slight increase in the errors, the obtained values remain sufficiently small for all considered values of $\boldsymbol{\theta}^{train}$, as evidenced by the numerical results presented in the tables. Moreover, these observations are further supported by the plots in Figures 5.6, 5.7, 5.8, and 5.9. The consistency between the numerical data and the graphical visualizations reinforces our confidence in the accuracy of the ADL-ROM approach.

These results also validate our previous analyses and interpretations of the solution plots, confirming that the ADL-ROM solution provides an adequate approximation of the exact solution for all the values of $\boldsymbol{\theta}^{train}$ under consideration.

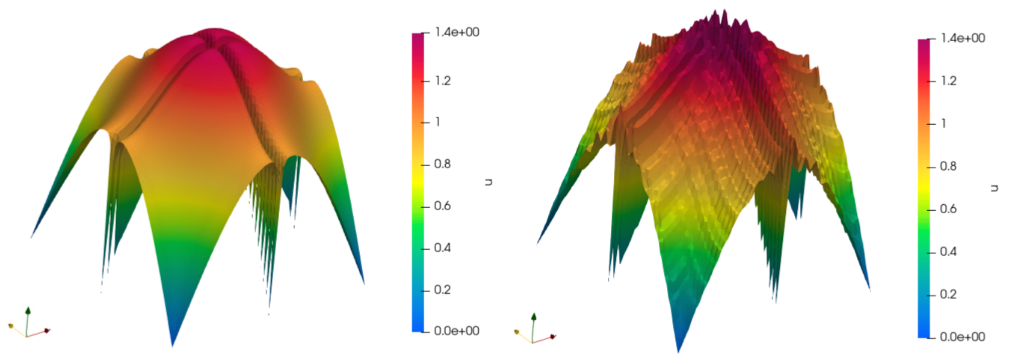

Figure 5.10: Exact solution (left) and ADL-ROM solution (right) at $t = 0.5$, for $\theta = -400$, $r = 50$, $\delta = 0.1313$, and $N = 5$. Color and position are determined by the magnitude of the velocity field.

Next, we extend our analysis to a different parameter setting, examining the numerical solutions for $\theta = -400$. Figures 5.10 and 5.11 depict the solutions, again in both 3D and 2D, at $t = 0.5$, for $\theta = -400$, i.e., the viscosity is set to 1.25×10^{-3}, and the same filter radius as before, $\delta = 0.1313$.

As expected, in this scenario, we observe that the primary challenge remains the accurate reconstruction of wave propagation, mirroring the diffi-

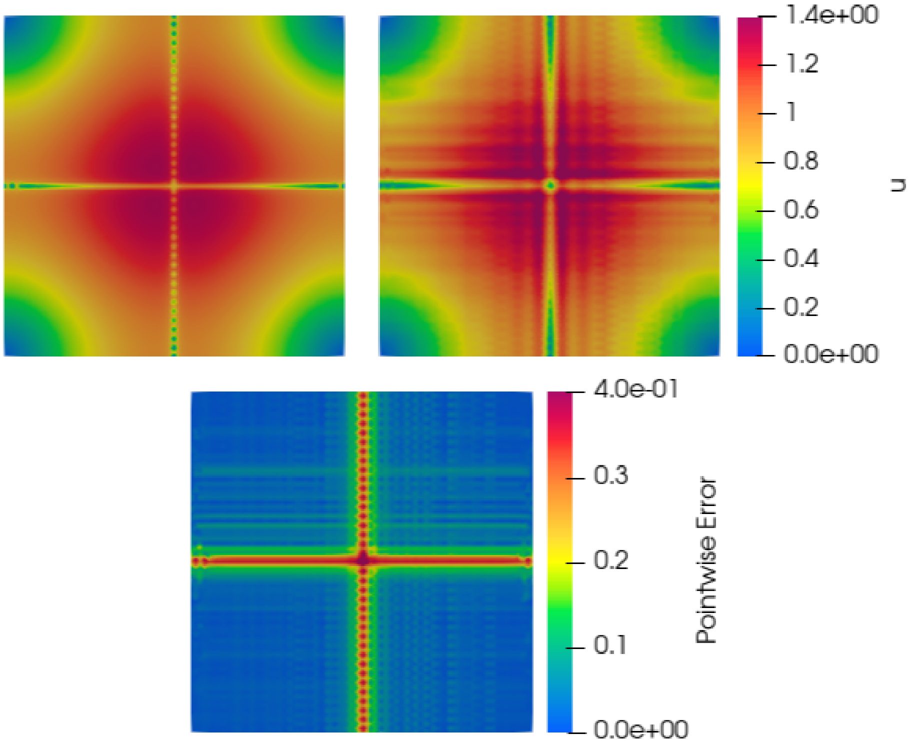

Figure 5.11: Exact solution (top left) and ADL-ROM solution (top right), and absolute pointwise error between the two solutions (bottom) at $t = 0.5$, for $\theta = -400$, $r = 50$, $\delta = 0.1313$, and $N = 5$.

culties encountered in the previous case, as seen in Figure 5.4. However, an important distinction is that the chosen parameter value, $\theta = -400$, belongs to the testing set rather than the training set. Consequently, the reconstruction of the flow dynamics becomes more challenging. As a result, the ADL-ROM solution in this case exhibits a slightly diminished accuracy compared to the previous case. Nevertheless, it still retains the dominant features of the flow, as evidenced by the bottom plot of Figure 5.11, where the absolute pointwise error does not increase excessively. In particular, the absolute pointwise error is less than 7×10^{-2} on about the 83% of the computational domain.

Furthermore, in Tables 5.5, 5.6, 5.7, and 5.8, we present the corresponding relative L^2 errors for the four values in the testing parameter set, $\boldsymbol{\theta}^{test}$. These tables include the minimum, maximum, average, and final time relative L^2 errors, offering a comprehensive assessment of the model performance on the testing parameter values.

By analyzing this additional data, we can confirm a similar trend as

θ^{test}	$\mathcal{E}_{\boldsymbol{u}}^{\min,r}$
-250	7.75×10^{-3}
-325	7.89×10^{-3}
-400	8.01×10^{-3}
-475	8.10×10^{-3}

Table 5.5: Minimum relative L^2 errors for decreasing θ^{test} values and for $\delta = 0.1313$.

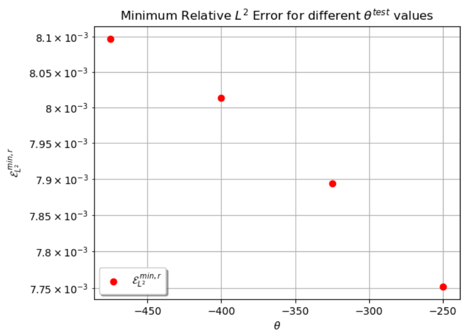

Figure 5.12: Behavior of $\mathcal{E}_{\boldsymbol{u}}^{\min,r}$ with respect to $\boldsymbol{\theta}^{test}$ for $\delta = 0.1313$.

θ^{test}	$\mathcal{E}_{\boldsymbol{u}}^{\max,r}$
-250	7.20×10^{-2}
-325	8.46×10^{-2}
-400	9.00×10^{-2}
-475	9.79×10^{-2}

Table 5.6: Maximum relative L^2 errors for decreasing θ^{test} values and for $\delta = 0.1313$.

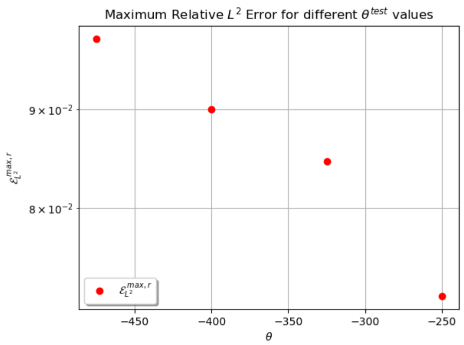

Figure 5.13: Behavior of $\mathcal{E}_{\boldsymbol{u}}^{\max,r}$ with respect to $\boldsymbol{\theta}^{test}$ for $\delta = 0.1313$.

θ^{test}	$\mathcal{E}_{\boldsymbol{u}}^{\text{ave},r}$
-250	4.44×10^{-2}
-325	4.92×10^{-2}
-400	5.27×10^{-2}
-475	5.53×10^{-2}

Table 5.7: Average relative L^2 errors for decreasing θ^{test} values and for $\delta = 0.1313$.

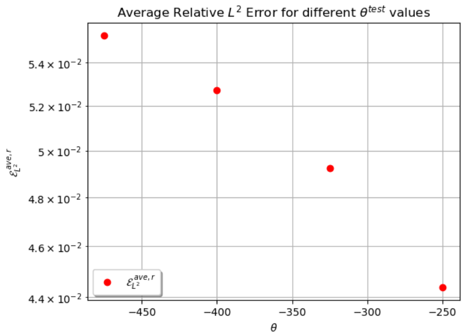

Figure 5.14: Behavior of $\mathcal{E}_{\boldsymbol{u}}^{\text{ave},r}$ with respect to $\boldsymbol{\theta}^{test}$ for $\delta = 0.1313$.

θ^{test}	$\mathcal{E}_{\boldsymbol{u}}^{T,r}$
-250	1.93×10^{-2}
-325	1.97×10^{-2}
-400	1.99×10^{-2}
-475	2.01×10^{-2}

Table 5.8: Final time relative L^2 errors for decreasing θ^{test} values and for $\delta = 0.1313$.

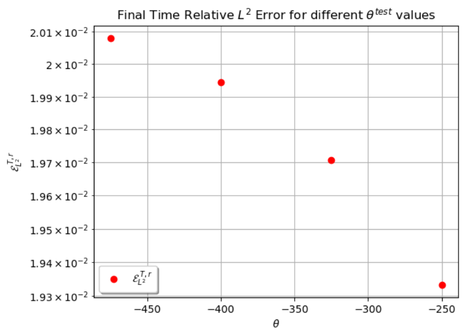

Figure 5.15: Behavior of $\mathcal{E}_{\boldsymbol{u}}^{T,r}$ with respect to $\boldsymbol{\theta}^{test}$ for $\delta = 0.1313$.

before: as the parameter θ decreases, the relative L^2 errors experience a slight increase. This observation is in line with our theoretical expectations and the behavior seen in the training set results. Moreover, as anticipated from both theoretical considerations and the solution plots, the errors associated with test parameter values, i.e., $\theta \in \boldsymbol{\theta}^{test}$, are slightly larger than those obtained for the training parameter values, i.e., $\theta \in \boldsymbol{\theta}^{train}$. This behavior is expected, as the model naturally performs slightly better on training values than on testing values. Nevertheless, it is important to note that the errors associated with the testing values of θ are still of the same order of magnitude as those associated with the training ones.

Thus, the errors for all considered $\boldsymbol{\theta}^{test}$ values remain sufficiently small, as clearly demonstrated in both the tables and the plots in Figures 5.12, 5.13, 5.14, and 5.15. This confirms that the ADL-ROM solution adequately approximates the exact solution for all the values in $\boldsymbol{\theta}^{test}$, albeit with a slightly lower accuracy compared to the training set.

This study represents an initial step in validating the effectiveness of the van Cittert-based approach within the ADL-ROM framework in scenarios where the model reduction is performed in both the time and parameter domains. These findings provide a foundation for further investigations into this topic.

Conclusions

The motivation behind the Approximate Deconvolution (AD) method aligns closely with the objectives of Large Eddy Simulation (LES), where the primary goal is to obtain accurate representations of dominant flow structures in under-resolved, convection-dominated fluid flow simulations while maintaining computational efficiency. In the Reduced Order Modeling (ROM) framework, resolving the finer flow scales in full detail is impractical. However, these small-scale structures significantly influence the accuracy of the solution. This challenge has been previously addressed using Leray-type models, which mitigate nonlinear effects to an extent sufficient for stabilizing convection-dominated flows [33, 37, 41, 47, 52, 85, 96, 99]. Nevertheless, a notable limitation of the Leray Reduced Order Model (L-ROM) is its strong dependence on the filter radius, δ, which is used to smooth out spurious numerical oscillations present in the standard Galerkin Reduced Order Model (G-ROM). Extensive numerical investigations on L-ROM have revealed that even slight deviations from the optimal δ values, i.e., those that yield the most accurate L-ROM results, can lead to significant inaccuracies. For instance, selecting an excessively large δ value to aggressively dampen the G-ROM oscillations often results in an overly diffusive L-ROM, reducing its predictive accuracy.

To counteract the overdiffusive nature of L-ROM while maintaining numerical stability, we propose our novel regularized ROM, the Approximate Deconvolution Leray ROM (ADL-ROM). This new framework incorporates approximate deconvolution techniques [16, 62] to enhance the accuracy of L-ROM without sacrificing its stabilizing properties. The principal innovation of ADL-ROM is the substitution of the L-ROM nonlinear term, $\overline{\boldsymbol{u}} \cdot \nabla \boldsymbol{u}$, with $D(\overline{\boldsymbol{u}}) \cdot \nabla \boldsymbol{u}$, where $D(\overline{\boldsymbol{u}})$ represents the approximate deconvolution of the filtered ROM velocity, $\overline{\boldsymbol{u}}$. Additionally, ADL-ROM introduces two novel types of approximate deconvolution operators at ROM level: the van Cittert AD and the Tikhonov AD.

Below, we summarize the main features emerged in this work from the numerical results of ADL-ROM in Section 3.3 of Chapter 3.

- Firstly, we validate our initial hypothesis: the application of AD operators yields results that are intermediary between the reduced solution and the filtered reduced solution. To verify this claim, we conduct numerical experiments on the 2D flow past a cylinder, a widely studied benchmark test case, in Section 3.3.2. By computing the G-ROM solution and subsequently applying the ROM Differential Filter (DF) and AD operators to it, we compare the results against the FEM reference solution. The numerical results confirm that AD solutions outperform both the reduced and filtered solutions, while the use of exact deconvolution proves to be computationally dangerous, other than theoretically useless.

- To further evaluate the effectiveness of ADL-ROM, we conduct comparative analyses of it with the standard L-ROM and G-ROM in simulations of convection-dominated flow past a backward-facing step, as discussed in Section 3.3.3. The findings of this numerical study lead to the following conclusions:

  1. When a large filter radius is used, ADL-ROM produces significantly more accurate results than both the classical G-ROM and standard L-ROM. Specifically, ADL-ROM introduces a minimal yet sufficient amount of numerical diffusion to stabilize the ROM simulation, whereas L-ROM suffers from excessive diffusion, resulting in reduced accuracy.
  2. ADL-ROM exhibits reduced sensitivity to variations in parameter values compared to L-ROM. Identifying optimal ADL-ROM parameters proves to be substantially more straightforward than determining the ideal L-ROM parameters. Consequently, ADL-ROM results demonstrate greater robustness concerning filtering parameters, and the differential filter concept facilitates the extension of this approach to broader regularization techniques beyond those examined in this work.
  3. When appropriately tuned, the three ROM approximate deconvolution strategies investigated, i.e., Lavrentiev AD, Tikhonov AD, and van Cittert AD, yield comparable performance, as elaborated in Section 3.3, Paragraph *AD Comparison*.

This thesis also presents the first comprehensive numerical analysis of the ROM approximate deconvolution operator (Section 4.1), which holds general applicability to all the AD ROM methodologies. This analysis establishes a crucial foundation for understanding the behavior and performance of ROM

AD operators in practical computational settings. More specifically, we rigorously derived an AD approximation bound, which is formally stated in Lemma 4.1.2. To further substantiate this theoretical result, we conducted an extensive numerical investigation, illustrating the corresponding rates of convergence in Section 4.3.2. These numerical experiments provide empirical validation of our theoretical findings and offer deeper insight into the accuracy of the approximate deconvolution process in ROM contexts.

Additionally, this thesis contains the first numerical analysis of the newly developed ADL-ROM (Section 4.2). In particular, we established an *a priori* error bound for ADL-ROM, which is rigorously formulated in Theorem 4.2.3. It is important to highlight that these newly obtained *a priori* error bounds offer a precise characterization of how the ADL-ROM error depends on both the ROM parameters and the FOM parameters.

To complement this theoretical result, we performed numerical experiments in Section 4.3.3, demonstrating the validity of the derived error bounds in practical computational scenarios. The convergence rates obtained in Section 4.3 not only confirmed our theoretical predictions but also provided valuable guidance regarding the impact of filter radius δ in both general AD methodologies and specific applications to the Navier-Stokes Equations (NSE). This is particularly relevant for practical computational settings, where trade-offs must be made between solution accuracy, resolved flow scales, and computational efficiency. In real-world scenarios, selecting an appropriate filter radius δ is often a challenging task that requires balancing numerical stability with accuracy. Without a rigorous mathematical framework describing the behavior of these filters, parameter tuning can only be performed heuristically on a case-by-case basis. As a result, ROMs tend to lack robustness and require recalibration whenever computational settings are modified.

To the best of our knowledge, the critical role played by the filter radius in ROM performance has primarily been investigated through numerical experiments. Relevant studies include the ROM-adaptive lengthscale method proposed in [70] and sensitivity analyses on δ conducted in [37, 94]. Chapter 4 of this work represents a significant advancement in this field by introducing, for the first time, a numerical analysis strategy for systematically assessing the influence of the filter radius on ROM accuracy and stability. Our approach provides a more rigorous alternative to previous heuristic tuning methods, thereby enhancing the robustness and predictive capabilities of ROM-based simulations (see [80] for a related yet distinct approach).

At the end of this work, we have also introduced the first extension of ADL-ROM to a parametric setting, marking an important step towards broadening its applicability. Specifically, in Chapter 5, we laid the groundwork for validating ADL-ROM in scenarios where space reduction is carried out not only in the time domain but also in the parameter space. This extension is crucial for ensuring that ADL-ROM remains effective in a wider range of practical applications where variations in physical or geometrical parameters play a significant role.

To this end, we conducted an initial numerical investigation aimed at evaluating the behavior of ADL-ROM under varying parameter values. The corresponding numerical results, presented in Section 5.3, demonstrate the ability of ADL-ROM to accurately capture the flow dynamics associated with an analytical reference solution, even in this more general setting. These experiments provide an encouraging first empirical validation, suggesting that the ADL-ROM framework can successfully accommodate parametric dependencies while maintaining its accuracy and stability.

Nevertheless, further numerical investigations are required to fully comprehend the implications of introducing parameter dependence into ADL-ROM. More extensive testing is needed to systematically analyze its performance across a broader range of parameter values and flow regimes. This additional research will allow for the refinement of the parametric ADL-ROM approach and enable a more comprehensive understanding of its behavior in practical applications.

In conclusion, this thesis has successfully integrated classical Approximate Deconvolution (AD) methodologies with Reduced Order Model (ROM) regularization techniques, representing a significant advancement in the field. The first-ever application of approximate deconvolution for constructing a regularized ROM, specifically the newly introduced ADL-ROM, has demonstrated promising results in improving the accuracy and stability of ROM simulations. Despite these achievements, several avenues for future research remain open and merit further exploration.

A primary direction for future work involves assessing the performance of ADL-ROM in the numerical simulation of more complex and challenging convection-dominated flows. For instance, the turbulent channel flow problem [70] serves as an ideal test case to evaluate the robustness and effectiveness of ADL-ROM under highly turbulent and under-resolved flow conditions.

Another promising research trajectory lies in integrating Reg-ROMs with new AD methodologies. Indeed, many AD techniques originate from the field of image processing, which has recently undergone rapid transformation with the advent of advanced machine learning algorithms. However, these modern AD advancements have not yet been incorporated into ROM research. There is substantial potential to investigate the effectiveness of contemporary AD techniques in a ROM framework, including both methods reminiscent of classical AD approaches [13] and those leveraging deep learning architectures [73].

Additionally, while ROM regularization techniques provide an efficient and cost-effective means of improving model accuracy, alternative strategies such as ROM closure models [98] offer a complementary approach to addressing the issue of unresolved flow scales. Further research could explore how ROM closure models might benefit from recent advances in ROM AD methods, potentially leading to hybrid approaches that combine the strengths of both regularization and closure strategies.

Lastly, establishing rigorous numerical analysis foundations for future ROM AD methodologies remains a crucial research goal. In particular, proving *a priori* error bounds for new ROM AD techniques would be instrumental in enhancing the reliability and predictive capabilities of ROM simulations. Providing robust theoretical justifications for these novel strategies will not only solidify their mathematical underpinnings but also foster broader adoption in practical applications.

In summary, while this thesis has made substantial progress in bridging AD methods with ROM regularization, numerous research opportunities remain for further development. By exploring more complex flow scenarios, incorporating modern AD techniques, refining ROM closure models, and advancing numerical analysis frameworks, the field of AD-based ROMs can continue to evolve toward more accurate, stable, and computationally efficient solutions.

List of Figures

3.1 The computational domain, Ω. $\partial\Omega_D \doteq \partial\Omega_D^{in} \cup \partial\Omega_D^{\text{wall}}$, where the inlet boundary $\partial\Omega_D^{in}$ is represented by a dashed red line and the no-slip boundaries by a solid blue line. 50
3.2 The FE mesh. 50
3.3 Normalized singular values from POD procedure with $r_{max} = 50$. 51
3.4 At $t = 10$, 1st: FOM solution $\boldsymbol{u}^h$, 2nd: G-ROM solution $\boldsymbol{u}^r$ for $r = 50$, 3rd: absolute pointwise error between the solutions $\boldsymbol{u}^h$ and $\boldsymbol{u}^r$. 52
3.5 At the final time $T = 20$, 1st: FOM solution $\boldsymbol{u}^h$, 2nd: G-ROM solution $\boldsymbol{u}^r$ for $r = 50$, 3rd: absolute pointwise error between the solutions $\boldsymbol{u}^h$ and $\boldsymbol{u}^r$. 52
3.6 At the final time $T = 20$ for $r = 50$, 1st: G-ROM solution $\boldsymbol{u}^r$, 2nd: filtered G-ROM solution $F^r(\boldsymbol{u}^r)$ for $\delta = 0.1$, 3rd: AD filtered G-ROM solution $D_\mu^r\left(F^r(\boldsymbol{u}^r)\right)$ for $\delta = 0.1$ and $\mu = 0.03$. 53
3.7 At the final time $T = 20$ for $r = 50$ 1st: $\boldsymbol{u}^r$, 2nd: $F^r(\boldsymbol{u}^r)$ for $\delta = 0.1$, 3rd: little disturbed $F^r(\boldsymbol{u}^r)$, i.e., $\mathscr{F}^r(\boldsymbol{u}^r)$, for $\delta = 0.1$, 4th: $D_\mu^r\left(\mathscr{F}^r(\boldsymbol{u}^r)\right)$ for $\delta = 0.1$ and $\mu = 0.0$, 5th: $D_\mu^r\left(\mathscr{F}^r(\boldsymbol{u}^r)\right)$ for $\delta = 0.1$ and $\mu = 0.03$, 6th: $D_\mu^r\left(\mathscr{F}^r(\boldsymbol{u}^r)\right)$ for $\delta = 0.1$ and $\mu = 0.3$. 54
3.8 At the final time $T = 20$ for $r = 50$ 1st: $\boldsymbol{u}^r$, 2nd: $F^r(\boldsymbol{u}^r)$ for $\delta = 0.1$, 3rd: very little disturbed $F^r(\boldsymbol{u}^r)$, i.e., $\mathscr{F}^r(\boldsymbol{u}^r)$, for $\delta = 0.1$, 4th: $D_\mu^r\left(\mathscr{F}^r(\boldsymbol{u}^r)\right)$ for $\delta = 0.1$ and $\mu = 0.0$, 5th: $D_\mu^r\left(\mathscr{F}^r(\boldsymbol{u}^r)\right)$ for $\delta = 0.1$ and $\mu = 0.03$, 6th: $D_\mu^r\left(\mathscr{F}^r(\boldsymbol{u}^r)\right)$ for $\delta = 0.1$ and $\mu = 0.3$. 55
3.9 The computational domain, Ω. $\partial\Omega_D \doteq \partial\Omega_D^{in} \cup \partial\Omega_D^{\text{wall}}$, where the inlet boundary $\partial\Omega_D^{in}$ is represented by a dashed red line and the no-slip boundaries by a solid blue line. 57
3.10 The FE mesh. 57
3.11 Time evolution of FOM kinetic energy. 58
3.12 Normalized singular values from POD procedure with $r_{max} = 60$. 59

3.13 Pointwise difference between AD applied to G-ROM and G-ROM, $r = 10$, $\delta = 0.6$ and . 1^{st}: Lavrentiev with $\mu = 0.1$; 2^{nd}: Tikhonov with $\mu = 0.1$; 3^{rd}: van Cittert with $N = 10$. 59

3.14 Pointwise difference between AD applied to G-ROM and G-ROM, $r = 10$, $\delta = 0.6$. 1^{st}: Tikhonov with $\mu = 0.05$. 2^{nd}: van Cittert with $N = 12$. 60

3.15 Time evolution of the absolute and relative L^2 errors for G-ROM (red), L-ROM (green), and Lavrentiev ADL-ROM (blue). 62

3.16 Time evolution of the absolute and relative L^2 errors for G-ROM (red), L-ROM (green), and van Cittert ADL-ROM (blue). 63

3.17 Time evolution of the kinetic energy for FOM projection (black), G-ROM (red), L-ROM (green), and Lavrentiev ADL-ROM (blue). 63

3.18 Time evolution of the kinetic energy for FOM projection (black), G-ROM (red), L-ROM (green), and van Cittert ADL-ROM (blue). 64

3.19 At the final time $T = 150$, 1^{st}: FOM solution; 2^{nd}: G-ROM solution; 3^{rd}: L-ROM solution; and 4^{th}: Lavrentiev ADL-ROM solution for $r = 10$ with $\delta = 0.6$ and $\mu = 0.06$. 65

3.20 At the final time $T = 150$, 1^{st}: absolute pointwise error between FOM and G-ROM solutions; 2^{nd}: absolute pointwise error between FOM and L-ROM solutions; and 3^{rd}: absolute pointwise error between FOM and Lavrentiev ADL-ROM solutions for $r = 10$ with $\delta = 0.6$ and $\mu = 0.06$. 65

3.21 At the final time $T = 150$, 1^{st}: FOM solution; 2^{nd}: G-ROM solution; 3^{rd}: L-ROM solution; and 4^{th}: van Cittert ADL-ROM solution for $r = 10$ with $\delta = 0.6$ and $N = 8$. 66

3.22 At the final time $T = 150$, 1^{st}: absolute pointwise error between FOM and G-ROM solutions; 2^{nd}: absolute pointwise error between FOM and L-ROM solutions; and 3^{rd}: absolute pointwise error between FOM and van Cittert ADL-ROM solutions for $r = 10$ with $\delta = 0.6$ and $\mu = 0.06$. 66

4.1 The computational domain, Ω. 87

4.2 The FE mesh. 88

4.3 Normalized singular values from POD procedure with $\nu = 10^{-3}$ and $r_{max} = 100$. 89

4.4 Normalized singular values from POD procedure with $\nu = 5 \times 10^{-4}$ and $r_{max} = 100$. 89

4.5 Exact solution (left) and ADL-ROM solution (right) at $t = 0.5$, for $\nu = 10^{-3}$, $r = 99$, $\delta = 6.25 \times 10^{-2}$, and $N = 5$. Color and position are determined by the magnitude of the velocity field. 90
4.6 Exact solution (left) and ADL-ROM solution (right) at $t = 0.5$, for $\nu = 10^{-3}$, $r = 99$, $\delta = 6.25 \times 10^{-2}$, and $N = 5$. Color and position are determined by the x-component of the velocity field. 91
4.7 Exact solution (left) and ADL-ROM solution (right) at $t = 0.5$, for $\nu = 10^{-3}$, $r = 99$, $\delta = 6.25 \times 10^{-2}$, and $N = 5$. Color and position are determined by the y-component of the velocity field. 91
4.8 Exact solution (left) and ADL-ROM solution (right) at $t = 0.5$, for $\nu = 5 \times 10^{-4}$, $r = 99$, $\delta = 1.25 \times 10^{-1}$, and $N = 5$. Color and position are determined by the magnitude of the velocity field. 92
4.9 Exact solution (left) and ADL-ROM solution (right) at $t = 0.5$, for $\nu = 5 \times 10^{-4}$, $r = 99$, $\delta = 1.25 \times 10^{-1}$, and $N = 5$. Color and position are determined by the x-component of the velocity field. 92
4.10 Exact solution (left) and ADL-ROM solution (right) at $t = 0.5$, for $\nu = 5 \times 10^{-4}$, $r = 99$, $\delta = 1.25 \times 10^{-1}$, and $N = 5$. Color and position are determined by the y-component of the velocity field. 93
4.11 At $t = 0.5$, for $N = 5$ and $r = 99$ Pointwise absolute error between the exact and the ADL-ROM solutions for $\nu = 10^{-3}$ and $\delta = 6.25 \times 10^{-2}$ (left) and for $\nu = 5 \times 10^{-4}$ and $\delta = 1.25 \times 10^{-1}$ (right). 94
4.12 Linear regression of $E^{AD}_{L^2}$ with respect to δ for $\nu = 10^{-3}$. . . . 95
4.13 Linear regression of $E^{AD}_{L^2}$ with respect to δ for $\nu = 5 \times 10^{-4}$. . 96
4.14 Linear regression of $E^{ADL-ROM}_{L^2}$ with respect to δ for $\nu = 10^{-3}$. 97
4.15 Linear regression of $E^{ADL-ROM}_{L^2}$ with respect to $\Lambda^r_{H^1}$ for $\nu = 10^{-3}$. 98
4.16 Linear regression of $E^{ADL-ROM}_{L^2}$ with respect to δ for $\nu = 5 \times 10^{-4}$. 99
4.17 Linear regression of $E^{ADL-ROM}_{L^2}$ with respect to $\Lambda^r_{H^1}$ for $\nu = 5 \times 10^{-4}$. 99

5.1 The computational domain, Ω. 105
5.2 The FE mesh. 107
5.3 Normalized singular values from POD procedure with $\nu \in \left[10^{-3}, 2.5 \times 10^{-3}\right]$ and $r_{max} = 105$. 107

5.4 Exact solution (left) and ADL-ROM solution (right) at $t = 0.5$, for $\theta = -200$, $r = 50$, $\delta = 0.1313$, and $N = 5$. Color and position are determined by the magnitude of the velocity field. 110
5.5 Exact solution (top left) and ADL-ROM solution (top right), and absolute pointwise error between the two solutions (bottom) at $t = 0.5$, for $\theta = -200$, $r = 50$, $\delta = 0.1313$, and $N = 5$. 111
5.6 Behavior of $\mathcal{E}_{\boldsymbol{u}}^{\min,r}$ with respect to $\boldsymbol{\theta}^{train}$ for $\delta = 0.1313$. 112
5.7 Behavior of $\mathcal{E}_{\boldsymbol{u}}^{\max,r}$ with respect to $\boldsymbol{\theta}^{train}$ for $\delta = 0.1313$. 112
5.8 Behavior of $\mathcal{E}_{\boldsymbol{u}}^{\mathrm{ave},r}$ with respect to $\boldsymbol{\theta}^{train}$ for $\delta = 0.1313$. 113
5.9 Behavior of $\mathcal{E}_{\boldsymbol{u}}^{T,r}$ with respect to $\boldsymbol{\theta}^{train}$ for $\delta = 0.1313$. 113
5.10 Exact solution (left) and ADL-ROM solution (right) at $t = 0.5$, for $\theta = -400$, $r = 50$, $\delta = 0.1313$, and $N = 5$. Color and position are determined by the magnitude of the velocity field. 114
5.11 Exact solution (top left) and ADL-ROM solution (top right), and absolute pointwise error between the two solutions (bottom) at $t = 0.5$, for $\theta = -400$, $r = 50$, $\delta = 0.1313$, and $N = 5$. 115
5.12 Behavior of $\mathcal{E}_{\boldsymbol{u}}^{\min,r}$ with respect to $\boldsymbol{\theta}^{test}$ for $\delta = 0.1313$. 116
5.13 Behavior of $\mathcal{E}_{\boldsymbol{u}}^{\max,r}$ with respect to $\boldsymbol{\theta}^{test}$ for $\delta = 0.1313$. 116
5.14 Behavior of $\mathcal{E}_{\boldsymbol{u}}^{\mathrm{ave},r}$ with respect to $\boldsymbol{\theta}^{test}$ for $\delta = 0.1313$. 117
5.15 Behavior of $\mathcal{E}_{\boldsymbol{u}}^{T,r}$ with respect to $\boldsymbol{\theta}^{test}$ for $\delta = 0.1313$. 117

List of Tables

3.1 Average absolute and relative L^2 errors for G-ROM, L-ROM, and ADL-ROM with two different AD operators and different N values. 62
3.2 L-ROM and van Cittert ADL-ROM average kinetic energies for various filter radii (δ) and van Cittert orders (N). FOM $\mathscr{K}(L^2)$ is 9.0417, G-ROM $\mathscr{K}(L^2)$ is 9.0703. 64

4.1 AD approximation error $E_{L^2}^{AD}$ for decreasing δ values and for $\nu = 10^{-3}$. 95
4.2 AD approximation error $E_{L^2}^{AD}$ for decreasing δ values and for $\nu = 5 \times 10^{-4}$. 96
4.3 ADL-ROM approximation error $E_{L^2}^{ADL-ROM}$ for decreasing δ values and for $\nu = 10^{-3}$. 97
4.4 ADL-ROM approximation error $E_{L^2}^{ADL-ROM}$ for increasing r values and for $\nu = 10^{-3}$. 98
4.5 ADL-ROM approximation error $E_{L^2}^{ADL-ROM}$ for decreasing δ values and for $\nu = 5 \times 10^{-4}$. 99
4.6 ADL-ROM approximation error $E_{L^2}^{ADL-ROM}$ for increasing r values and for $\nu = 5 \times 10^{-4}$. 99

5.1 Minimum relative L^2 errors for decreasing θ^{train} values and for $\delta = 0.1313$. 112
5.2 Maximum relative L^2 errors for decreasing θ^{train} values and for $\delta = 0.1313$. 112
5.3 Average relative L^2 errors for decreasing θ^{train} values and for $\delta = 0.1313$. 113
5.4 Final time relative L^2 errors for decreasing θ^{train} values and for $\delta = 0.1313$. 113
5.5 Minimum relative L^2 errors for decreasing θ^{test} values and for $\delta = 0.1313$. 116

5.6 Maximum relative L^2 errors for decreasing θ^{test} values and for $\delta = 0.1313$. 116
5.7 Average relative L^2 errors for decreasing θ^{test} values and for $\delta = 0.1313$. 117
5.8 Final time relative L^2 errors for decreasing θ^{test} values and for $\delta = 0.1313$. 117

Bibliography

[1] S. Ali, F. Ballarin, and G. Rozza. “A Reduced basis stabilization for the unsteady Stokes and Navier-Stokes equations”. In: *Advances in Computational Science and Engineering* 1.2 (2023), pp. 180–201. DOI: 10.3934/acse.2023008.

[2] S. Ali, F. Ballarin, and G. Rozza. “Stabilized reduced basis methods for parametrized steady Stokes and Navier-Stokes equations”. In: *Computers & Mathematics with Applications* 80.11 (2020), pp. 2399–2416. DOI: 10.1016/j.camwa.2020.03.019. eprint: 2001.00820.

[3] M. S. Alnæs, A. Logg, K. B. Ølgaard, M. E. Rognes, and G. N. Wells. “Unified Form Language: A domain-specific language for weak formulations of partial differential equations”. In: *ACM Transactions on Mathematical Software* 40.2 (2014), pp. 231–244. DOI: https://doi.org/10.1145/2566630.

[4] D. N. Arnold and A. Logg. “Periodic table of the finite elements”. In: *SIAM News* 47.9 (2014).

[5] C. Audouze, F. De Vuyst, and P. B. Nair. “Reduced-order modeling of parameterized PDEs using time–space-parameter principal component analysis”. In: *International Journal for Numerical Methods in Engineering* 80.8 (2009), pp. 1025–1057. DOI: https://doi.org/10.1002/nme.2540. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.2540.

[6] J. Baiges, R. Codina, and S. Idelsohn. “Reduced-order subscales for POD models”. In: *Computer Methods in Applied Mechanics and Engineering* 291 (2015), pp. 173–196.

[7] M. Balajewicz and E. H. Dowell. “Stabilization of projection-based reduced order models of the Navier–Stokes”. In: *Nonlinear Dynamics* 70 (2012), pp. 1619–1632.

[8] 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". In: *J. Comput. Phys.* 315 (2016), pp. 609–628.

[9] F. Ballarin and T. Iliescu. "*A priori* Error Bounds for POD-ROMs for Fluids: A Brief Survey". In: *arXiv preprint* *`http://arxiv.org/abs/2409.00621`* (2024).

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

[11] I. A. Baratta et al. "DOLFINx: The next generation FEniCS problem solving environment". In: *Zenodo preprint* (2023). DOI: `https://doi.org/10.5281/zenodo.10447666`.

[12] M. F. Barone, I. Kalashnikova, D. J. Segalman, and H. K. Thornquist. "Stable Galerkin reduced order models for linearized compressible flow". In: *Journal of Computational Physics* 228.6 (2009), pp. 1932–1946.

[13] M. Benning and M. Burger. "Modern regularization methods for inverse problems". In: *Acta Numerica* 27 (2018), pp. 1–111. DOI: `10.1017/S0962492918000016`.

[14] M. Bergmann, C. H. Bruneau, and A. Iollo. "Enablers for robust POD models". In: *J. Comput. Phys.* 228.2 (2009), pp. 516–538.

[15] L. C. Berselli, T. Iliescu, and W. J. Layton. *Mathematics of Large Eddy Simulation of Turbulent Flows*. Scientific Computation. Berlin: Springer-Verlag, 2006, pp. xviii+348. ISBN: 978-3-540-26316-6; 3-540-26316-0.

[16] M. Bertero and P. Boccacci. *Introduction to Inverse Problems in Imaging*. Bristol: Institute of Physics Publishing, 1998, pp. xii+351. ISBN: 0-7503-0439-1; 0-7503-0435-9.

[17] B. Brands, J. Mergheim, and P. Steinmann. "Reduced-order modelling for linear heat conduction with parametrised moving heat sources". In: *GAMM-Mitteilungen* 39.2 (2016), pp. 170–188. DOI: `https://doi.org/10.1002/gamm.201610011`. URL: `https://onlinelibrary.wiley.com/doi/abs/10.1002/gamm.201610011`.

[18] S. L. Brunton and J. N. Kutz. *Data-driven science and engineering: Machine learning, dynamical systems, and control.* Cambridge University Press, 2019.

[19] K. Carlberg, C. Bou-Mosleh, and C. Farhat. "Efficient non-linear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximations". In: *International Journal for Numerical Methods in Engineering* 86.2 (2011), pp. 155–181.

[20] D. Chapelle, A. Gariah, and J. Sainte-Marie. "Galerkin approximation with Proper Orthogonal Decomposition: new error estimates and illustrative examples". In: *ESAIM: Math. Model. Numer. Anal.* 46 (2012), pp. 731–757.

[21] M. Couplet, P. Sagaut, and C. Basdevant. "Intermodal energy transfers in a proper orthogonal decomposition–Galerkin representation of a turbulent separated flow". In: *J. Fluid Mech.* 491 (2003), pp. 275–284. ISSN: 0022-1120.

[22] V. DeCaria, T. Iliescu, W. Layton, M. McLaughlin, and M. Schneier. "An Artificial Compression Reduced Order Model". In: *SIAM J. Numer. Anal.* 58.1 (2020), pp. 565–589.

[23] A. A. Dunca. "Numerical analysis and testing of a stable and convergent finite element scheme for approximate deconvolution turbulence models". In: *Comput. Math. Appl.* 75.2 (2018), pp. 690–702. ISSN: 0898-1221. DOI: `https://doi.org/10.1016/j.camwa.2017.09.035`. URL: `https://www.sciencedirect.com/science/article/pii/S0898122117306156`.

[24] S. Eskew and J. R. Singler. "A new approach to proper orthogonal decomposition with difference quotients". In: *Adv. Comput. Math.* 49.2 (2023), p. 13.

[25] *FEniCS Project.* URL: `https://fenicsproject.org/`.

[26] L. Fick, Y. Maday, A. T. Patera, and T. Taddei. "A stabilized POD model for turbulent flows over a range of Reynolds numbers: Optimal parameter sampling and constrained projection". In: *Journal of Computational Physics* 371 (2018), pp. 214–243.

[27] C. Foias, D. D. Holm, and E. S. Titi. "The Navier–Stokes-alpha model of fluid turbulence". In: *Physica D: Nonlinear Phenomena* 152 (2001), pp. 505–519.

[28] B. García-Archilla, V. John, and J. Novo. "Second order error bounds for POD-ROM methods based on first order divided differences". In: *Appl. Math. Lett.* 146 (2023), p. 108836.

[29] B. García-Archilla and J. Novo. "Pointwise error bounds in POD methods without difference quotients". In: *arXiv preprint, http://arxiv.org/abs/2407.17159* (2024).

[30] M. Germano. "Differential filters for the large eddy numerical simulation of turbulent flows". In: *Phys. Fluids* 29.6 (1986), pp. 1755–1757. ISSN: 0031-9171.

[31] M. Germano. "Differential filters of elliptic type". In: *Phys. Fluids* 29.6 (1986), pp. 1757–1758. ISSN: 0031-9171.

[32] B. J. Geurts and D. D. Holm. "Regularization modeling for large-eddy simulation". In: *Physics of Fluids* 15.1 (2003), pp. L13–L16.

[33] M. Girfoglio, A. Quaini, and G. Rozza. "A Finite Volume approximation of the Navier-Stokes equations with nonlinear filtering stabilization". In: *Comput. & Fluids* 187 (2019), pp. 27–45. DOI: 10.1016/j.compfluid.2019.05.001.

[34] M. Girfoglio, A. Quaini, and G. Rozza. "A hybrid projection/data-driven reduced order model for the Navier-Stokes equations with non-linear filtering stabilization". In: *Journal of Computational Physics* 486 (2023), p. 112127. DOI: 10.1016/j.jcp.2023.112127.

[35] M. Girfoglio, A. Quaini, and G. Rozza. "A linear filter regularization for POD-based reduced-order models of the quasi-geostrophic equations". In: *Comptes Rendus. Mécanique* (2023). DOI: 10.5802/crmeca.183.

[36] M. Girfoglio, A. Quaini, and G. Rozza. "A novel Large Eddy Simulation model for the Quasi-Geostrophic equations in a Finite Volume setting". In: *Journal of Computational and Applied Mathematics* 418 (2023), p. 114656. ISSN: 0377-0427. DOI: 10.1016/j.cam.2022.114656.

[37] M. Girfoglio, A. Quaini, and G. Rozza. "A POD-Galerkin reduced order model for a LES filtering approach". In: *J. Comput. Phys.* 436 (2021), p. 110260. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2021.110260.

[38] M. Girfoglio, A. Quaini, and G. Rozza. "Pressure Stabilization Strategies for a LES Filtering Reduced Order Model". In: *Fluids* 6.9 (2021). DOI: 10.3390/fluids6090302.

[39] S. Grimberg, C. Farhat, and N. Youkilis. "On the stability of projection-based model order reduction for convection-dominated laminar and turbulent flows". In: *Journal of Computational Physics* 419 (2020), p. 109681.

[40] M. Gunzburger, T. Iliescu, M. Mohebujjaman, and M. Schneier. "An evolve-filter-relax stabilized reduced order stochastic collocation method for the time-dependent Navier-Stokes equations". In: *SIAM / ASA Journal on Uncertainty Quantification* 7.4 (2019), pp. 1162–1184.

[41] M. Gunzburger, T. Iliescu, and M. Schneier. "A Leray regularized ensemble-proper orthogonal decomposition method for parameterized convection-dominated flows". In: *IMA J. Numer. Anal.* 40.2 (2020), pp. 886–913.

[42] M. Gunzburger, N. Jiang, and M. Schneier. "An Ensemble-Proper Orthogonal Decomposition Method for the Nonstationary Navier-Stokes Equations". In: *SIAM Journal on Numerical Analysis* 55.1 (2017), pp. 286–304.

[43] P. C. Hansen. *Discrete Inverse Problems: Insight and Algorithms*. Vol. 7. Society for Industrial and Applied Mathematics, 2010.

[44] J. S. Hesthaven, G. Rozza, and B. Stamm. *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*. Springer, 2015.

[45] C. Himpe, T. Leibner, and S. Rave. "Hierarchical Approximate Proper Orthogonal Decomposition". In: *SIAM Journal on Scientific Computing* 40.5 (2018), A3267–A3292. DOI: `10.1137/16M1085413`. URL: `https://doi.org/10.1137/16M1085413`.

[46] C. Huang, C. R. Wentland, K. Duraisamy, and C. Merkle. "Model reduction for multi-scale transport problems using model-form preserving least-squares projections with variable transformation". In: *J. Comput. Phys.* 448 (2022), p. 110742.

[47] T. Iliescu, H. Liu, and X. Xie. "Regularized Reduced Order Models for a Stochastic Burgers Equation". In: *International Journal of Numerical Analysis and Modeling* 15.4–5 (2018), pp. 594–607.

[48] T. Iliescu and Z. Wang. "Are the Snapshot Difference Quotients Needed in the Proper Orthogonal Decomposition?" In: *SIAM J. Sci. Comput.* 36.3 (2014), A1221–A1250.

[49] T. Iliescu and Z. Wang. "Variational Multiscale Proper Orthogonal Decomposition: Navier-Stokes Equations". In: *Num. Meth. P.D.E.s* 30.2 (2014), pp. 641–663.

[50] T. Kadeethum, F. Ballarin, and N. Bouklas. "Data-driven reduced order modeling of poroelasticity of heterogeneous media based on a discontinuous Galerkin approximation". English. In: *GEM* 12 (2021), 12–N/A. ISSN: 1869-2672. DOI: `10.1007/s13137-021-00180-4`.

[51] 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”. In: *International Journal for Numerical Methods in Engineering* 83.10 (2010), pp. 1345–1375.

[52] K. Kaneko, P.-H. Tsai, and P. Fischer. “Towards model order reduction for fluid-thermal analysis”. In: *Nucl. Eng. Des.* 370 (2020), p. 110866.

[53] S. Kaya and C. C. Manica. “Convergence analysis of the finite element method for a fundamental model in turbulence”. In: *Math. Models Methods Appl. Sci.* 22.11 (2012), p. 1250033.

[54] R. C. Kirby and A. Logg. “A compiler for variational forms”. In: *ACM Transactions on Mathematical Software* 32.6 (2006), pp. 417–444. DOI: `http://doi.org/10.1145/1163641.1163644`.

[55] B. Koc, S. Rubino, M. Schneier, J. R. Singler, and T. Iliescu. “On Optimal Pointwise in Time Error Bounds and Difference Quotients for the Proper Orthogonal Decomposition”. In: *SIAM J. Numer. Anal.* 59.4 (2021), pp. 2163–2196.

[56] 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”. In: *arXiv preprint,http://arxiv.org/abs/1501.02004* (2015).

[57] K. Kunisch and S. Volkwein. “Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics”. In: *SIAM J. Numer. Anal.* 40.2 (2002), 492–515 (electronic). ISSN: 0036-1429.

[58] K. Kunisch and S. Volkwein. “Galerkin proper orthogonal decomposition methods for parabolic problems”. In: *Numer. Math.* 90.1 (2001), pp. 117–148. ISSN: 0029-599X.

[59] 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”. In: *Num. Meth. P.D.E.s* 24.2 (2008), pp. 555–582.

[60] W. J. Layton. *Introduction to the numerical analysis of incompressible viscous flows.* Vol. 6. Society for Industrial and Applied Mathematics (SIAM), 2008.

[61] W. J. Layton and R. Lewandowski. “A high accuracy Leray-deconvolution model of turbulence and its limiting behavior”. In: *Anal. Appl.* 6.01 (2008), pp. 23–49.

[62] W. J. Layton and L. G. Rebholz. *Approximate Deconvolution Models of Turbulence: Analysis, Phenomenology and Numerical Analysis*. Vol. 2042. Springer Berlin Heidelberg, 2012.

[63] J. Leray. "Sur le mouvement d'un fluide visqueux emplissant l'espace". In: *Acta Mathematica* 63 (1934), pp. 193–248.

[64] A. Logg, K.-A. Mardal, and G. N. Wells. *Automated Solution of Partial Differential Equations by the Finite Element Method*. Verlag Berlin Heidelberg: Springer, 2012. DOI: `http://doi.org/10.1007/978-3-642-23099-8`.

[65] 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". In: *SIAM J. Numer. Anal.* 47.1 (2008), pp. 1–19. ISSN: 0036-1429.

[66] M. Mohebujjaman, L. G. Rebholz, X. Xie, and T. Iliescu. "Energy balance and mass conservation in reduced order models of fluid flows". In: *J. Comput. Phys.* 346 (2017), pp. 262–277.

[67] I. Moore. "Filter Based Stabilization Methods for Reduced Order Models of Convection-Dominated Systems". MA thesis. Virginia Tech, 2023. URL: `http://hdl.handle.net/10919/115065`.

[68] I. Moore, A. Sanfilippo, F. Ballarin, and T. Iliescu. "*A Priori* Error Bounds for the Approximate Deconvolution Leray Reduced Order Model". In: *arXiv preprint, https://arxiv.org/abs/2410.02673* (2024).

[69] C. Mou, B. Koc, O. San, L. G. Rebholz, and T. Iliescu. "Data-Driven Variational Multiscale Reduced Order Models". In: *Computer Methods in Applied Mechanics and Engineering* 373 (2021), p. 113470.

[70] C. Mou, E. Merzari, O. San, and T. Iliescu. "An Energy-Based Lengthscale for Reduced Order Models of Turbulent Flows". In: *Nucl. Eng. Des.* 412 (2023), p. 112454.

[71] B. R. Noack, P. Papas, and P. A. Monkewitz. "The need for a pressure-term representation in empirical Galerkin models of incompressible shear flows". In: *J. Fluid Mech.* 523 (2005), pp. 339–365.

[72] J. Novo and S. Rubino. "Error analysis of proper orthogonal decomposition stabilized methods for incompressible flows". In: *SIAM J. Numer. Anal.* 59.1 (2021), pp. 334–369.

[73] G. Ongie, A. Jalal, C. A. Metzler, R. G. Baraniuk, A. G. Dimakis, and R. Willett. "Deep Learning Techniques for Inverse Problems in Imaging". In: *IEEE J. Sel. Areas Inf. Theory* 1.1 (2020), pp. 39–56. DOI: 10.1109/JSAIT.2020.2991563.

[74] P. Pacciarini and G. Rozza. "Stabilized reduced basis method for parametrized advection–diffusion PDEs". In: *Computer Methods in Applied Mechanics and Engineering* 274 (2014), pp. 1–18.

[75] E. J. Parish, C. Wentland, and K. Duraisamy. "The adjoint Petrov–Galerkin method for non-linear model reduction". In: *Comput. Meth. Appl. Mech. Eng.* 365 (2020), p. 112991.

[76] E. J. Parish, M. Yano, I. Tezaur, and T. Iliescu. "Residual-based stabilized reduced-order models of the transient convection-diffusion-reaction equation obtained through discrete and continuous projection". In: *arXiv preprint, http://arxiv.org/abs/2302.09355* (2023).

[77] A. Quarteroni, A. Manzoni, and F. Negri. *Reduced Basis Methods for Partial Differential Equations: An Introduction.* Vol. 92. Springer, 2015.

[78] *RBniCS Project.* URL: https://www.rbnicsproject.org/.

[79] *RBniCSx Project.* URL: https://github.com/RBniCS/RBniCSx.

[80] J. Reyes, P. H. Tsai, J. Novo, and T. Iliescu. "*A priori* Error Bounds and Parameter Scalings for the Time Relaxation Reduced Order Model". In: *arXiv preprint, http://arxiv.org/abs/2411.08986* (2024).

[81] R. Reyes and R. Codina. "Projection-based reduced order models for flow problems: A variational multiscale approach". In: *Computer Methods in Applied Mechanics and Engineering* 363 (2020), p. 112844.

[82] G. Rozza and K. Veroy. "On the stability of the reduced basis method for Stokes equations in parametrized domains". In: *Comput. Methods Appl. Mech. Engrg.* 196.7 (2007), pp. 1244–1260.

[83] S. Rubino. "Numerical analysis of a projection-based stabilized POD-ROM for incompressible flows". In: *SIAM J. Numer. Anal.* 58.4 (2020), pp. 2019–2058.

[84] S.B. Pope. *Turbulent flows.* Cambridge: Cambridge University Press, 2000, pp. xxxiv+771. ISBN: 0-521-59886-9.

[85] F. Sabetghadam and A. Jafarpour. "α regularization of the POD-Galerkin dynamical systems of the Kuramoto–Sivashinsky equation". In: *Appl. Math. Comput.* 218.10 (2012), pp. 6012–6026.

[86] A. Sanfilippo, I. Moore, F. Ballarin, and T. Iliescu. "Approximate deconvolution Leray reduced order model for convection-dominated flows". In: *Finite Elements in Analysis and Design* 226 (2023), p. 104021. ISSN: 0168-874X. DOI: `https://doi.org/10.1016/j.finel.2023.104021`. URL: `https://www.sciencedirect.com/science/article/pii/S0168874X23001142`.

[87] M. W. Scroggs, I. A. Baratta, C. N. Richardson, and G. N. Wells. "Basix: a runtime finite element basis evaluation library". In: *Journal of Open Source Software* 7.73 (2022), p. 3982. DOI: `10.21105/joss.03982`. URL: `https://doi.org/10.21105/joss.03982`.

[88] M. W. Scroggs, J. S. Dokken, C. N. Richardson, and G. N. Wells. "Construction of Arbitrary Order Finite Element Degree-of-Freedom Maps on Polygonal and Polyhedral Cell Meshes". In: *ACM Trans. Math. Softw.* 48.2 (May 2022). ISSN: 0098-3500. DOI: `10.1145/3524456`. URL: `https://doi.org/10.1145/3524456`.

[89] J. R. Singler. "New POD error expressions, error bounds, and asymptotic results for reduced order models of parabolic PDEs". In: *SIAM J. Numer. Anal.* 52.2 (2014), pp. 852–876.

[90] G. Stabile and G. Rozza. "Finite volume POD-Galerkin stabilised reduced order methods for the parametrised incompressible Navier-Stokes equations". In: *Comput. & Fluids* 173 (2018), pp. 273–284.

[91] M. Strazzullo, M. Girfoglio, F. Ballarin, T. Iliescu, and G. Rozza. "Consistency of the full and reduced order models for evolve-filter-relax regularization of convection-dominated, marginally-resolved flows". In: *International Journal for Numerical Methods in Engineering* 123.14 (2022), pp. 3148–3178. DOI: `10.1002/nme.6942`. eprint: `2110.05093`.

[92] R. Temam. *Navier-Stokes equations: Theory and numerical analysis*. Vol. 2. American Mathematical Society, 2001.

[93] D. Torlo, F. Ballarin, and G. Rozza. "Stabilized Weighted Reduced Basis Methods for Parametrized Advection Dominated Problems with Random Inputs". In: *SIAM/ASA Journal on Uncertainty Quantification* 6.4 (2018), pp. 1475–1502. DOI: `10.1137/17M1163517`.

[94] P. H. Tsai, P. Fischer, and T. Iliescu. "A Time-Relaxation Reduced Order Model for the Turbulent Channel Flow". In: *arXiv preprint, http://arxiv.org/abs/arXiv:2312.13272* (2023).

[95] D. Wells. "Stabilization of POD-ROMs". PhD thesis. `http://vtechworks.lib.vt.edu/bitstream/handle/10919/52960/Wells_DR_D_2015.pdf?sequence=1&isAllowed=y`: Virginia Tech, 2015.

[96] D. Wells, Z. Wang, X. Xie, and T. Iliescu. "An evolve-then-filter regularized reduced order model for convection-dominated flows". In: *Int. J. Num. Meth. Fluids* 84 (2017), pp. 598–615.

[97] X. Xie. "Approximate Deconvolution Reduced Order Modeling". MA thesis. http://theses.lib.vt.edu/theses/available/etd-11232015-161835/unrestricted/Xie_X_T_2015.pdf: Virginia Tech, 2015.

[98] X. Xie, D. Wells, Z. Wang, and T. Iliescu. "Approximate Deconvolution Reduced Order Modeling". In: *Comput. Methods Appl. Mech. Engrg.* 313 (2017), pp. 512–534.

[99] X. Xie, D. Wells, Z. Wang, and T. Iliescu. "Numerical Analysis of the Leray Reduced Order Model". In: *J. Comput. Appl. Math.* 328 (2018), pp. 12–29.

[100] F. Zoccolan, M. Strazzullo, and G. Rozza. "A streamline upwind Petrov-Galerkin reduced order method for advection-dominated partial differential equations under optimal control". In: *arXiv preprint, http://arxiv.org/abs/arXiv:2301.01973* (2023).