Approximate Deconvolution Reduced Order Modeling

Xuping Xie

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

Masters of Science
in
Mathematics

Traian Iliescu, Chair
Serkan Gugercin
Tao Lin

01 May 2013
Blacksburg, Virginia

Keywords: Reduced Order Modeling, Large Eddy Simulation, Approximate Deconvolution, Inverse Problems, Regularization

Approximate Deconvolution Reduced Order Modeling

Xuping Xie

(ABSTRACT)

This thesis proposes a large eddy simulation reduced order model (LES-ROM) framework for the numerical simulation of realistic flows. In this LES-ROM framework, the proper orthogonal decomposition (POD) is used to define the ROM basis and a POD differential filter is used to define the large ROM structures. An approximate deconvolution (AD) approach is used to solve the ROM closure problem and develop a new AD-ROM. This AD-ROM is tested in the numerical simulation of the one-dimensional Burgers equation with a small diffusion coefficient ($\nu = 10^{-3}$).

To my parents

Acknowledgments

This thesis represents more than two years of my work in reduced order modeling. I would not have made it this far without the help of my academic family and the members of the Interdisciplinary Center for Applied Mathematics (ICAM) and the Department of Mathematics.

First, I would like to express my gratitude to my advisor, Dr. Traian Iliescu, who led me into this exciting research area and supported my research and study in many ways. And of course my âĂİacademic brothersâĂİ Dr. Zhu Wang and Dr. David Wells, for all their help in scientific computing and mathematical modeling.

I would like to thank my thesis committee members, Dr. Serkan Gugercin and Dr. Tao Lin, for their insightful, valuable discussions and suggestions.

I would also like to thank Dr. Boris Kramer for providing helpful resources for my research.

More generally, I would like to thank the Department of Mathematics and ICAM for making Blacksburg feel like home.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Turbulence Flow

Turbulence is one of the hardest problems in science and engineering. The modeling of realistic turbulent flows is very important in, e.g., chemical engineering, bioengineering, aerospace engineering, geophysics, and quantum mechanics, see [26, 25, 51, 64, 73]. The numerical simulation of turbulent flows is extremely challenging due to the enormous range of spatial and temporal scales that need to be approximated. According to Kolomogorov's theory, to capture all the energy containing scales requires $Re^{9/4}$ degrees of freedom, where Re is the Reynolds number [42]. Thus, a direct numerical simulation (DNS) is impossible for many practical turbulent flows for which Re is on the order of 10^5 or higher. Therefore, alternative numerical methods have been generally used for realistic turbulent flows over the past several decades.

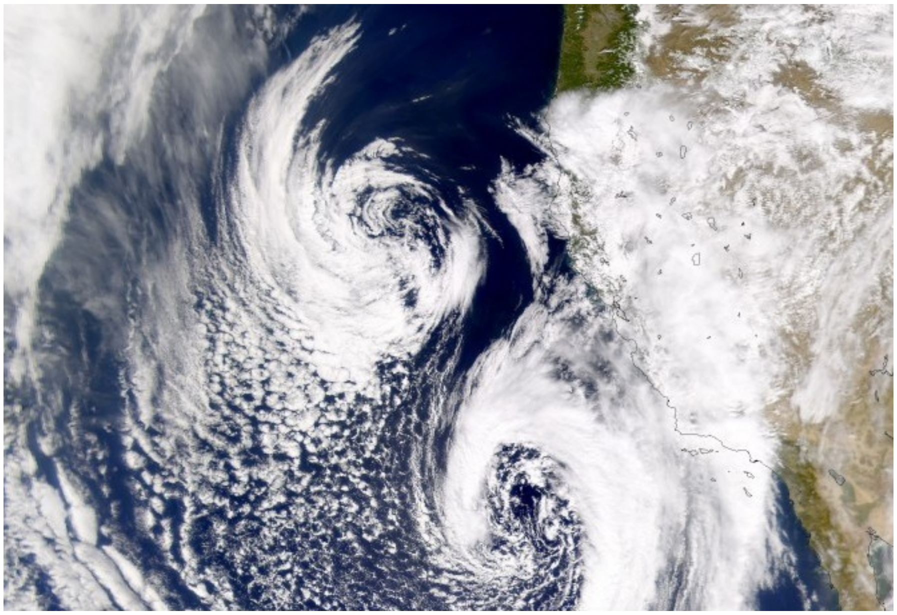


Figure 1.1: Atmospheric turbulence, satellite photo of the US West Coast, NASA, `http://www.wired.com/2009/08/hummingearth`

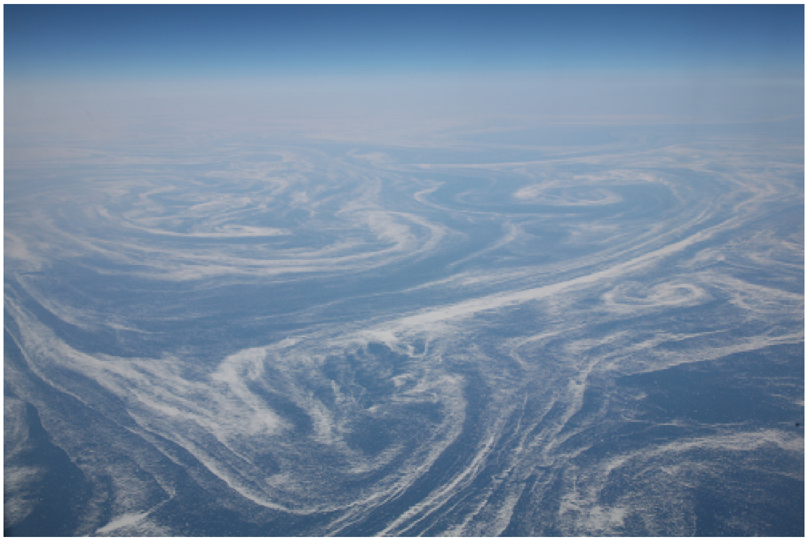


Figure 1.2: Ice fields in Labrador Current from a plane, Canada. `http://icyseas.org/2013/05/08/the-turbulence-of-van-gogh-and-the-labrador-shelf-current/`

1.2 Large Eddy Simulation (LES)

The distinguishing feature of a turbulent flow is that its velocity field appears to be random and varies unpredictably. The flow does, however, satisfy the Navier-Stokes differential equations, which are not random. This contrast is the source of much of what is interesting in turbulence theory...

One should keep in mind that a practical person is usually interested only in mean properties of a small number of functionals of the flow, and these could conceivably be obtained even when the details of the flow are unknown.

-A.J.Chorin, in Lectures on Turbulence Theory, Houston, 1975

Large eddy simulation (LES) is one of the most popular methods for the numerical simulation of realistic turbulence [13, 60]. It is motivated by two physical ideas: *1. Differing dynamics of the large and small eddies.* The velocity field under high Reynolds number is very sensitive, but it is not uniform. Those small eddies have a random character which make them more sensitive, and become universal features. So, their mean effects can be modelled by the large eddies. *2. The eddy-viscosity assumption.* On the other hand, the small eddies act to drain energy from the large eddies. And the mean fluid flowing turbulently behaves like a fluid of greatly increased and highly variable viscosity [46]. LES models aim at approximating the spatially filtered flow variables, which can be discretized on coarser meshes instead of approximating the exact flow variables.

However, as the Navier-Stokes equations (NSE) that model the flow are nonlinear, filtering them yields a system of equations that is not closed. So, how to close the filtered NSE is a well known issue in LES. There are many different LES closure models [13, 60], which could be divided into two categories: the first type is *functional closure models* (See chapters 5 and 6 in [60]). This approach amounts to adding an eddy viscosity to the molecular viscosity of the flow. A well known example in this class is the Smagorinsky model [66]. The second type of LES closure models is *structural closure models* (See chapter 7 in [60]). These closure models are different from the previous ones as they are derived through mathematical arguments, e.g., formal series expansions. The approximate deconvolution model is one of the most popular structural closure model [63, 69, 71, 72]. It uses the deconvolution idea developed in the image processing and inverse problems communities to reconstruct the original signal from a blurred filtered signal. The main goal of this thesis is to extend the approximate deconvolution approach to reduced order modeling.

1.3 Reduced Order Modeling

Another efficient numerical simulation of fluid flows is by using the reduced order models (ROMs), which can reduce the computational time of a DNS by orders of magnitude without significantly reducing its accuracy. The proper orthogonal decomposition (POD) is one of the most successful approaches for ROM development. It is a data driven technique, also known as principal component analysis (PCA), and singular value decomposition (SVD) [4, 64, 73, 75]. In POD, an accurate numerical simulation is used to extract the most dominant structures, i.e., POD modes, which are used later in a Galerkin approximation of the underlying dynamical system [35, 65]. In this thesis, POD will be exclusively used to construct the ROMs.

Standard ROMs are extremely efficient and relatively accurate for laminar flows [9, 16, 68]. They fail in the numerical simulation of realistic turbulent flows [6, 8, 9, 32, 54], however, because they encounter the same challenge faced in LES models: the closure problem. Indeed, we only use the first few POD modes in the ROM to achieve the low computational cost. However, for convection-dominated flows the corresponding low dimensional ROM generates bad numerical results in the form of numerical oscillations [32, 37, 78]. For example, in [32], the standard ROM of a two-dimensional (2D) convection-dominated flow was not stable. Similarly, in [78], for a three-dimensional (3D) flow past a cylinder at $Re = 10^3$, the standard ROM results were inaccurate: they over-predicted the coherent structures, kinetic energy content, and POD mode coefficient evolution. The results in [78] (see also[44, 49]) clearly show that the effect of the POD modes not used in the ROM needs to be modeled, i.e., *the ROM closure problem needs to be solved.*

Over the past several decades, a lot different ROM closure models have been developed (see, e.g., [3, 6, 7, 8, 9, 10, 12, 51, 56, 78]). These ROM closure models are considered to be of functional type as they have generally used some kind of stabilization procedure to model the effect of the discarded modes just as in LES models. A physical motivation for this popular approach is given in [24], where it is shown that the concept of energy cascade is also valid in a POD setting.

In this thesis, structural ROM closure models are put forth. To this end, the ROM spatial filtering proposed in [59, 78, 80] is used to develop an LES-ROM framwork. In this LES-ROM framework, we use a novel structural ROM closure model based on approximate deconvolution (AD). Specifically, given the approximation of the filtered ROM variables, we use AD to obtain an approximation of the original unfiltered ROM variables and solve the ROM closure problem. Since the AD problem is notoriously ill-posed [15, 47, 74], we use regularization methods from image processing and inverse problems to obtain stable AD approximations. Note that the resulting new AD-ROM is fundamentally different from the calibration ROMs used in [2, 22, 76, 79], which did not use an explicit ROM spatial filter.

Chapter 2

Proper Orthogonal Decomposition

2.1 Introduction

Proper orthogonal decomposition (POD) is a data analysis tool, often used for data reduction purposes. POD extracts the most dominant (”most pertinent”) structures from experimental or simulation data. The POD method, also known as Karhuenen-Loeve expansion [40, 48] principal component anaysis, Hotelling transformation [36], and singular value decomposition (SVD) [41], has a long rich history. POD has been employed to produce ROMs of Galerkin type, which are commonly used for simulation, design and control.

2.2 Mathematical Formulation

Now, we briefly describe the mathematical formulation of POD. One should refer to [43, 75] for more details.

Assume that $\mathbf{y} \in L^2(\mathcal{H}, (t_0{+}t_0{+}T))$ is the solution of a given dynamical system. Given the time instances, $t_1, ..., t_M \in [0, T]$, let Y be the ensemble of snapshots $Y = [\mathbf{y}(t_1), ..., \mathbf{y}(t_M)]$, with $\operatorname{rank}(Y) = d$. POD yields a set of basis functions that optimally represent the given solution data. This leads to an optimization problem with low-dimensional basis $\{\varphi_1, \varphi_2, ..., \varphi_r\}$ as follows:

$$\min \frac{1}{M} \sum_{j=1}^{M} \|\mathbf{y}(t_j) - \sum_{i=1}^{r} (\mathbf{y}(t_j), \varphi_i)_{\mathcal{H}} \varphi\|_{\mathcal{H}}^2, \tag{2.1}$$

such that $(\varphi_i, \varphi_j)_{\mathcal{H}} = \delta_{ij}$. Since the L^2 norm is the most common choice in literature, we consider the L^2 norm throughout this thesis. Note, however, that other norms (e.g., the H^1 norm) have also been used [76, 77].

Next, we provide the solution of the optimization problem (2.1). Considering orthogonality of POD basis, we have that for every i, j

$$\begin{aligned} 0 &\leq \|\mathbf{y}(t_j) - \sum_{i=1}^{r}(\mathbf{y}(t_j), \varphi_i)\varphi_i\|_2^2 \\ &= (\mathbf{y}(t_j) - \sum_{i=1}^{r}(\mathbf{y}(t_j), \varphi_i)\varphi_i, \mathbf{y}(t_j) - \sum_{i=1}^{r}(\mathbf{y}(t_j), \varphi_i)\varphi_i) \\ &= \|\mathbf{y}(t_j)\|_2^2 - 2\sum_{i=1}^{r}(\mathbf{y}(t_j), \varphi_i)^2 + \sum_{i=1}^{r}(\mathbf{y}(t_j), \varphi_i)^2 \\ &= \|\mathbf{y}(t_j)\|_2^2 - \sum_{i=1}^{r}(\mathbf{y}(t_j), \varphi_i)^2 \end{aligned} \tag{2.2}$$

It is clear that (2.1) is equivalent to the following optimization problem:

$$\max \sum_{j=1}^{M}\sum_{i=1}^{r}(\mathbf{y}(t_j), \varphi_i)^2 \tag{2.3}$$

Such that $(\varphi_i, \varphi_j) = \delta_{ij}$. We apply the Lagrange multiplier method to (2.3). Let λ be the auxiliary variable and set the lagrangian as

$$L(\varphi, \lambda) = \sum_{j=1}^{M}(\mathbf{y}(t_j), \varphi_i)^2 + \lambda[1 - (\varphi, \varphi)]. \tag{2.4}$$

Its derivative is

$$\sum_{j=1}^{M}(\mathbf{y}(t_j), \varphi_i)\mathbf{y}(t_j) = \lambda\varphi \tag{2.5}$$

Which we can reformulate as an eigenvalue problem,

$$YY^T\varphi = \lambda\varphi. \tag{2.6}$$

The function φ is called POD mode (or POD basis function) and the value λ is called the POD eigenvalue. The method of snapshots was introduced in [75] as an alternative to (2.6). It is the most common approach in generating the POD basis for large data sets. The method of snapshots is more efficient than the standard method in (2.6), since it performs the SVD of an $M \times M$ matrix instead of computing the SVD of an $n \times M$ matrix with $n \gg M$.

Chapter 3

Reduced Order Modeling

3.1 Introduction

ROMs attempt to replace the high rank model by low dimensional dynamical systems, which are valuable whenever the same problem must be solved repeatedly, e.g., for control applications and parameter estimation. ROMs reduce the dimension and computational complexity of a system, using a high resolution data set obtained by either physical experiment or numerical simulation. The primary goal is to construct surrogate models that are cheap to execute and still maintain the large scale features of the original system. There are several reduced order modeling strategies surveyed in, e.g., [5, 11, 34, 35, 49, 57, 56]. In this thesis, we exclusively consider ROMs that are based on POD.

3.2 Standard Galerkin ROM (G-ROM)

In this section, we develop the standard ROM for the NSE, which are used as mathematical model:

$$\frac{\partial \boldsymbol{u}}{\partial t} - Re^{-1}\Delta \boldsymbol{u} + \boldsymbol{u}\cdot\nabla\boldsymbol{u} + \nabla p = \mathbf{0}\,, \tag{3.1}$$

$$\nabla\cdot\boldsymbol{u} = 0\,, \tag{3.2}$$

where $\boldsymbol{u}$ is the velocity, p the pressure and Re the Reynolds number. The NSE (3.1)–(3.2) are supplemented with the initial condition $\boldsymbol{u}(\mathbf{x},0) = \boldsymbol{u}_0(\mathbf{x})$ and homogeneous Dirichlet boundary condition: $\boldsymbol{u}(\mathbf{x},t) = \mathbf{0}$ on the boundary.

By collecting snapshots of the velocity field and applying the POD method, the POD approximation of the velocity is

$$\boldsymbol{u}_r(\mathbf{x},t) \equiv \sum_{j=1}^{r} a_j(t)\boldsymbol{\varphi}_j(\mathbf{x}), \tag{3.3}$$

where $\{a_j(t)\}_{j=1}^{r}$ are the sought time-varying coefficients that represent the POD-Galerkin trajectories. Replacing the velocity $\boldsymbol{u}$ by $\boldsymbol{u}_r$ in the NSE (3.1), we obtain

$$\frac{\partial \boldsymbol{u}_r}{\partial t} - Re^{-1}\Delta \boldsymbol{u}_r + \boldsymbol{u}_r\cdot\nabla\boldsymbol{u}_r + \nabla p = 0\,, \tag{3.4}$$

Projecting the resulting equations (3.4) onto the subspace $\mathbf{X}^r$ spanned by the POD basis, $\mathbf{X}^r = span\{\varphi_1, ..., \varphi_r\}$. We obtain the *Galerkin ROM (G-ROM)*: $\forall\, k = 1, \ldots, r$,

$$\left(\frac{\partial \boldsymbol{u}_r}{\partial t}, \boldsymbol{\varphi}_k \right) + Re^{-1} \, (\nabla \boldsymbol{u}_r, \nabla \boldsymbol{\varphi}_k) + \Big((\boldsymbol{u}_r \cdot \nabla) \boldsymbol{u}_r, \boldsymbol{\varphi}_k \Big) = 0 \, . \tag{3.5}$$

Note that, for incompressible flows, the pressure terms can be generally ignored in (3.5) (see, however, [1, 20, 50]). Since r is usually small (e.g., $r = \mathcal{O}(10)$), the G-ROM provides an efficient surrogate model for simulating laminar flows. However, for high *Re* flows, the G-ROM is not a viable tool [6, 8, 9, 56, 78]. The main reason for the G-ROM's poor performance in high *Re* flows is that the POD modes $\{\boldsymbol{\varphi}_j\}_{j=r+1}^{R}$, which are not used in the G-ROM, play an important role in the physical evolution of the system. Thus, the *ROM closure problem* [78] needs to be addressed, i.e., the ROM needs to model the effect of the discarded POD modes $\{\boldsymbol{\varphi}_j\}_{j=r+1}^{R}$. If the ROM closure problem is not addressed, then the standard G-ROM generally yields inaccurate results for complex flows, often in the form of numerical oscillations.

3.3 ROM Closure

A lot of different ROM closure approaches have been studied over the years [6, 8, 12, 22, 23, 29, 28, 51, 55, 62, 64]. In this section, we outline the ROM closure models of LES type.

3.3.1 Mixing Length ROM

The first ROM closure model was the mixing length ROM in [6]. This model increases the viscosity coefficient ν by

$$\nu_{ML} = \alpha \nu_T = \alpha U_{ML} L_{ML}, \tag{3.6}$$

where U_{ML} and L_{ML} are characteristic velocity and length scales for the unresolved scales, and α is an $\mathcal{O}(1)$ non-dimensional parameter that characterizes the energy being dissipated. The parameter α is expected to vary in a real turbulent flow, and different valued of α may result in different dynamics of the flow [6]. There are also different ways to define ν_T in (3.6).

3.3.2 Smagorinsky ROM

This model is an improvement of the mixing length ROM which is obtained by replacing the constant ν_{ML} in (3.6) with a variable turbulent viscosity. This yields a ROM closure model in which the viscosity coefficient is increased by

$$\nu_S = 2 (C_S \delta)^2 \| \mathbb{D}(\boldsymbol{u}_r) \|, \tag{3.7}$$

where C_S is the Smagorinsky constant, δ is the length scale and $\| \mathbb{D}(\boldsymbol{u}_r) \| = \| (\nabla \boldsymbol{u}_r + \nabla \boldsymbol{u}_r^T)/2 \|$ is the Frobenius norm of the deformation tensor. The Smagorinsky ROM was investigated in [73, 77, 78]. It is more accurate than the mixing length ROM as it recomputes the coefficient at every time step, whereas the mixing length ROM utilizes a constant EV.

3.3.3 Variational Multiscale ROM

The variational multiscale ROM is based on the principle of locality of energy transfer, i.e., it uses the ansatz that energy is transfered mainly between neighboring scales [12, 37, 39, 78]. The derivation of the variational multiscale ROM closure model starts by decomposing the finite set of POD modes $\mathbf{X}^r$ into the large resolved POD modes $\mathbf{X}^r_L$ and small resolved modes $\mathbf{X}^r_S$:

$$\begin{aligned} \mathbf{X}^r_L &= span\{\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, ..., \boldsymbol{\varphi}_R\} && (3.8) \\ \mathbf{X}^r_S &= span\{\boldsymbol{\varphi}_{R+1}, \boldsymbol{\varphi}_{R+2}, ..., \boldsymbol{\varphi}_r\} && (3.9) \end{aligned}$$

Then, we decompose $\boldsymbol{u}_r$ into two components: the large resolved scales, $\boldsymbol{u}_r^L$, and small resolved scales, $\boldsymbol{u}_r^S$:

$$\boldsymbol{u}_r = \boldsymbol{u}_r^L + \boldsymbol{u}_r^S, \boldsymbol{u}_r^L = \boldsymbol{U} + \sum_{j=1}^{R} a_j \boldsymbol{\varphi}_j, \boldsymbol{u}_r^S = \sum_{j=R+1}^{r} a_j \boldsymbol{\varphi}_j. \tag{3.10}$$

Finally, one can get the variational multiscale ROM by substituting (3.10) into (3.5) and using a Smagorinsky ROM closure model only in the equations for $\boldsymbol{u}_r^S$. Thus, the variational multiscale ROM is different from the ROM closure models in Sections 3.3.1 and 3.3.3 since it only acts on the small resolved scales.

3.3.4 Dynamic Subgrid-Scale ROM

The dynamic subgrid-scale ROM was proposed in [78] and is an extension of the state-of-the-art LES dynamic subgrid-scale model. In [78] it was shown that the dynamic subgrid scale ROM performs better than the variational multiscale ROM (Section 3.3.3) and clearly outperforms the Smagorinsky ROM (Section 3.3.2) and the mixing length ROM (Section 3.3.1). The major improvement in the dynamic subgrid-scale ROM is that it computes the EV constant dynamically, at each time instance and each spatial point. This is achieved by considering a secondary spatial filter and using the Germano identity [14, 61].

Chapter 4

Large Eddy Simulation ROM

In this chapter, we introduce the LES-ROM framework (Section 4.2), which is centered around ROM spatial filtering (Section 4.1.

4.1 ROM Spatial Filtering

In standard reduced order modeling, spatial filtering is generally defined implicitly, by truncating the POD basis used in the Galerkin approximation. In this thesis, we use explicit ROM spatial filtering. In [80, 81] two different types of explicit ROM spatial filters were used: the ROM Projection Section 4.1.1 and ROM differential filter Section 4.1.2.

4.1.1 ROM Projection

Given $\boldsymbol{u}_r \in \mathbf{X}^r$, for a fixed $r_1 < r$, the POD projection attempts to find $P_r(\boldsymbol{u}_r) \in \mathbf{X}^{r_1}$ such that

$$(P_r(\boldsymbol{u}_r), \boldsymbol{\varphi}_j) = (\boldsymbol{u}_r, \boldsymbol{\varphi}_j), \forall j = 1, ..., r_1. \tag{4.1}$$

The effect of the POD projection on an input from the POD space $\mathbf{X}^r$ is the elimination of the components corresponding to the POD basis functions $\{\boldsymbol{\varphi}_j\}_{j=r_1+1}^{r}$ which generally correspond to small scales [77, 80]. The POD projection (4.1) was used as an explicit ROM spatial filter in [78, 80]. It was also used as a theoretical tool in the error analysis of the G-ROM [43].

4.1.2 ROM Differential Filter

The ROM differential filter is defined as follows: Let δ be the radius of the ROM differential filter, which determines the length scales modeled in the LES-ROM framework. For a given $\boldsymbol{u}^r \in \mathbf{X}^r$, the *ROM differential filter (DF)* seeks $\overline{\boldsymbol{u}}^r \in \mathbf{X}^r$ such that

$$\left(\left(I - \delta^2 \Delta \right) \overline{\boldsymbol{u}}^r, \boldsymbol{\varphi}_j \right) = (\boldsymbol{u}^r, \boldsymbol{\varphi}_j), \quad \forall\, j = 1, \ldots r\,. \tag{4.2}$$

The differential filter has been introduced by Germano in LES [30, 31]. It was also used in a ROM context to develop regularized ROMs: The DF was first used in [59] for the 1D Kuramoto–Sivashinsky equation in a periodic setting. The DF was subsequently used in [80] for the 3D NSE in a general non-periodic setting.

The DF (4.2) has several appealing properties [13, 47, 58, 60]: First, it acts as a spatial filter, since it eliminates the small scales (i.e., high frequencies) from the input. Indeed, the DF uses an elliptic operator to smooth the input variable. Second, the DF has an extremely low computational overhead, since it amounts to solving a linear system with a very small $r \times r$ matrix that is precomputed. Third, the DF preserves incompressibility in the NSE, since the POD basis functions are incompressible and the DF is a linear operator.

The algorithmic decisions needed in the implementation of the DF are carefully discussed in Section 3.4 in [80]. Next, we summarize these decisions. If the input variable is in $\mathbf{X}^r$, i.e., $\boldsymbol{u} = \sum_{j=1}^{r} U_j \, \boldsymbol{\varphi}_j$, then the following linear system needs to be solved:

$$\left(\boldsymbol{M} + \delta^2 \, \boldsymbol{S}\right) \overline{\boldsymbol{U}} = \boldsymbol{U} \, , \tag{4.3}$$

where $\boldsymbol{M} \in \mathbb{R}^{r \times r}$ is the POD mass matrix with entries $(\boldsymbol{S})_{ij} = (\boldsymbol{\varphi}_j, \boldsymbol{\varphi}_i)$, $1 \leq i, j \leq r$, $\boldsymbol{S} \in \mathbb{R}^{r \times r}$ is the POD stiffness matrix with entries $(\boldsymbol{S})_{ij} = (\nabla \boldsymbol{\varphi}_j, \nabla \boldsymbol{\varphi}_i)$, $1 \leq i, j \leq r$, $\overline{\boldsymbol{U}} \in \mathbb{R}^r$ is the vector of coefficients of the output filtered variable $\overline{\boldsymbol{u}}$ with entries $(\overline{\boldsymbol{U}})_i = \overline{U}_i$, $1 \leq i \leq r$, and $\boldsymbol{U} \in \mathbb{R}^r$ is the vector with entries $(\boldsymbol{U})_i = (\boldsymbol{u}, \boldsymbol{\varphi}_i)$, $1 \leq i \leq r$. If the input variable is not in $\mathbf{X}^r$ (e.g., the input variable is the centering trajectory), then the finite element version of the DF is used to prefilter the input variable (see Section 3.4 in [80] for details).

4.2 Large Eddy Simulation ROMs (LES-ROMs)

As we mentioned before in Section 3, numerous ROM closure modeling approaches have been proposed to address the inaccuracy (and numerical instability) of the standard G-ROM (see, e.g., [49, 56, 78] and references therein). In this thesis, we take a different approach based on LES methodology. The LES-ROM framework is centered around ROM spatial filtering, which is presented in detail in Section 4.1. To the best of my knowledge, the LES-ROM framework has only been used in Section 3.3.4 in [78], in the definition of the dynamic sub grid-scale ROM. Since the LES-ROM framework is essential to the AD-ROM development, we briefly present its main components next.

Using the fact that $\nabla \cdot \boldsymbol{u}_r = 0$ in (3.4), we get $(\boldsymbol{u}_r \cdot \nabla) \, \boldsymbol{u}_r = \nabla \cdot (\boldsymbol{u}_r \, \boldsymbol{u}_r)$. Thus, (3.4) can be rewritten as

$$\frac{\partial \boldsymbol{u}_r}{\partial t} - Re^{-1} \Delta \boldsymbol{u}_r + \nabla \cdot (\boldsymbol{u}_r \, \boldsymbol{u}_r) + \nabla p = 0. \tag{4.4}$$

Just as in LES, we filter all the terms in (4.4) with a ROM spatial filter (which will be defined explicitly in Section 4.1). We then use the fact that the ROM spatial filter is a linear operator and assume that differentiation and ROM filtering commute. We obtain

$$\frac{\partial \overline{\boldsymbol{u}}_r}{\partial t} - Re^{-1} \Delta \overline{\boldsymbol{u}}_r + \nabla \cdot (\overline{\boldsymbol{u}_r \, \boldsymbol{u}_r}) + \nabla \overline{p} = 0. \tag{4.5}$$

Remark 4.1 (ROM Commutation Error)**.** *If filtering and differentiation do not commute, one has to estimate the commutation error [13].*

Equation (4.5) can be rewritten as

$$\frac{\partial \overline{\boldsymbol{u}}_r}{\partial t} - Re^{-1} \Delta \overline{\boldsymbol{u}}_r + \nabla \cdot (\overline{\boldsymbol{u}}_r \, \overline{\boldsymbol{u}}_r) + \nabla \cdot \boldsymbol{\tau}_r + \nabla \overline{p} = 0, \tag{4.6}$$

where

$$\boldsymbol{\tau}_r = \overline{\boldsymbol{u}_r \, \boldsymbol{u}_r} - \overline{\boldsymbol{u}}_r \, \overline{\boldsymbol{u}}_r \tag{4.7}$$

is the *ROM subfilter-scale stress tensor*. Using a Galerkin projection of (4.7) onto $\mathbf{X}^r$ and the fact that $\nabla \cdot \boldsymbol{u}_r = 0$, the *spatially filtered G-ROM* is obtained: $\forall \, k = 1, \ldots, r$,

$$\left(\frac{\partial \overline{\boldsymbol{u}}_r}{\partial t}, \boldsymbol{\varphi}_k \right) + Re^{-1} \, (\nabla \overline{\boldsymbol{u}}_r, \nabla \boldsymbol{\varphi}_k) + \Big((\overline{\boldsymbol{u}}_r \cdot \nabla) \overline{\boldsymbol{u}}_r, \boldsymbol{\varphi}_k \Big) + (\boldsymbol{\tau}_r, \nabla \boldsymbol{\varphi}_k) = 0 \, . \tag{4.8}$$

The spatial structures in the spatially filtered G-ROM (4.8) are larger than the spatial structures in the G-ROM (3.5). Thus, it is expected that the spatially filtered G-ROM requires fewer POD modes than the G-ROM, which is advantageous from a computational point of view.

Of course, the spatially filtered G-ROM (4.8) is not closed. Thus, just as in LES, one needs to address the ROM closure problem, i.e., to model the ROM subfilter-scale stress tensor $\boldsymbol{\tau}_r$ in terms of the ROM filtered velocity $\overline{\boldsymbol{u}}_r$. Once the ROM closure problem is addressed, the resulting ROM is called *large eddy simulation ROM (LES-ROM)*.

Chapter 5

Approximate Deconvolution ROM

Using the LES-ROM terminology [60], the ROM closure models mentioned in Section 3.3 are of functional type, since they model the physical role played by the ROM subfilter-scale stress tensor $\boldsymbol{\tau}_r$ in (4.8). In this section we introduce a new type of ROM closure model, the approximate deconvolution ROM (AD-ROM). The AD-ROM is a structural ROM closure model, since it uses mathematical arguments to approximate the ROM subfilter-scale stress tensor $\boldsymbol{\tau}$. The AD modeling has been successfully used in LES [47, 69, 70, 71, 72]. Its use in ROM development is, to the best of my knowledge, new.

5.1 Approximate Deconvolution ROM (AD-ROM)

The deconvolution idea, which is central in *image processing* and *inverse problems* [15, 33, 74], is simple: Given an approximation of the filtered input signal, find an approximation of the input itself. Specifically, in our ROM setting, denoting the spatial filter operator by G, we assume that an approximation of the filtered flow variable

$$\overline{\boldsymbol{u}}_r := G\,\boldsymbol{u}_r \tag{5.1}$$

is available. The goal in the deconvolution problem is to find the original flow variable, $\boldsymbol{u}_r$. Since the DF presented in Section 4.1 is invertible, at first glance, one just has to use the inverse of the filter G in (5.1) to solve the deconvolution problem:

$$\boldsymbol{u}_r{}^{ED} = G^{-1}\,\overline{\boldsymbol{u}}_r\,. \tag{5.2}$$

In inverse problems, this is generally a bad idea because the inverse problem (5.1) is *ill-posed*: small changes in the data can lead to large changes in the solution. Indeed, inverting the operator G in (5.2) results in division by small coefficients of the high-frequency components of the operator G [15, 47, 74]. Thus, any changes in the input (5.1) translate into large nonphysical oscillations in the output (5.2) [15, 74]. The fact that (5.2) is useless in our LES-ROM framework is carefully shown in Section 5.2.

To find a useful approximation to the ROM deconvolution problem (5.1), we use some of the regularization methods developed in the image processing and inverse problems communities (which are outlined in Section 5.3) to obtain an AD approximation:

$$\boldsymbol{u}_r{}^{AD} \approx \boldsymbol{u}_r{}^{ED} = G^{-1}\,\overline{\boldsymbol{u}}_r\,. \tag{5.3}$$

The AD approximation (5.3) is then used to close the spatially filtered G-ROM (4.8):

$$\left(\frac{\partial \overline{\boldsymbol{u}}_r}{\partial t}, \boldsymbol{\varphi}_k\right) + Re^{-1}\,\left(\nabla \overline{\boldsymbol{u}}_r, \nabla \boldsymbol{\varphi}_k\right) + \left(\overline{\boldsymbol{u}_r^{AD}\,\boldsymbol{u}_r^{AD}}, \nabla \boldsymbol{\varphi}_k\right) = 0\,. \tag{5.4}$$

Thus, we obtain a new LES-ROM, the *approximate deconvolution ROM (AD-ROM)*:

$$\left(\frac{\partial \boldsymbol{w}_r}{\partial t}, \boldsymbol{\varphi}_k\right) + Re^{-1}\,\left(\nabla \boldsymbol{w}_r, \nabla \boldsymbol{\varphi}_k\right) + \left(\overline{\boldsymbol{w}_r^{AD}\,\boldsymbol{w}_r^{AD}}, \nabla \boldsymbol{\varphi}_k\right) = 0\,, \tag{5.5}$$

where $\boldsymbol{w}_r$ is an approximation of the spatially filtered flow variable $\overline{\boldsymbol{u}}_r$.

5.2 ROM Exact Deconvolution

In this section, we answer the following question: *Is the ROM exact deconvolution problem ill-posed?* That is, given a signal filtered with the DF (4.2), can the original signal be recovered by applying the inverse of the DF to the filtered signal? In our ROM setting, this would translate in investigating whether the ROM exact deconvolution problem (5.2) is both stable and accurate. We will present several different numerical tests to answer this question.

5.2.1 Is the ROM Exact Deconvolution Ill-Posed?

We emphasize that this is an important practical question. Indeed, if the ROM exact deconvolution problem (5.2) is well-posed, then we can simply use the ROM exact deconvolution in the AD-ROM (5.5). Otherwise, we need to use AD techniques, just as in LES.

Also note that the deconvolution problem was shown to be ill-posed in the LES context [47]. We emphasize, however, that since the LES and ROM settings are significantly different, (e.g., the problem sizes are vastly different – $\mathcal{O}(10^6)$ for LES and $\mathcal{O}(10)$ for ROM), we need to explicitly investigate whether the ROM exact deconvolution problem is ill-posed. To this end, we consider the following algorithm:

Algorithm 1

(1) Consider input signal u.

(2) Calculate the filtered input signal $\overline{u} := G\,u$, where G is the DF (4.2).

(3) *Exactly deconvolve (ED)* the input signal:

$$u^{ED} = G^{-1}\,\overline{u} = (\mathbf{M} + \delta^2\,\mathbf{S})\,\overline{u}\,. \tag{5.6}$$

(4) Compare the ED signal u^{ED} with the true signal u.

One of the following two cases can result from Algorithm 1: *Case 1*: u^{ED} is close to u. In this case, we use the ED method (5.6) in the AD-ROM (5.5), i.e., we solve small linear systems without any regularization. *Case 2*: u^{ED} is not close to u. In this case, we need to use the regularization methods presented in Section 5.3 in the AD-ROM (5.5).

Noisy Data In the numerical investigation in this section, the effect of *noise* on the filtered input signal will play a central role. Specifically, Algorithm 1 will be modified as follows.

Algorithm 2

(1) Consider input signal u.

(2) Calculate the filtered input signal $\overline{u} := G\, u$, where G is the DF (4.2).

(3) *Exactly deconvolve (ED)* the *noisy* input signal:

$$u^{ED} = G^{-1}\left(\overline{u} + \eta\right) = \left(\mathbf{M} + \delta^2\, \mathbf{S}\right)\left(\overline{u} + \eta\right), \tag{5.7}$$

where η is the noise.

(4) Compare the ED signal u^{ED} with the true signal u.

Adding noise to the filtered input signal is common practice in image processing and inverse problems [15, 33, 74]. This is also relevant to our AD-ROM (5.5), where noise could model, e.g., the inherent error in the numerical approximation of the filtered flow variable, $\overline{\boldsymbol{u}}_r$. Indeed, in the numerical discretization of the AD-ROM (5.5), the filtered flow variable $\overline{\boldsymbol{u}}_r$ is generally not available. Instead, a *numerical approximation* of the filtered flow variable is available. To further reinforce this point that the added noise models the AD-ROM numerical discretization error, the magnitude of the noise used in this section is within the magnitude of the AD-ROM's spatial discretization error (see Section **??**).

5.2.2 Condition Number of ROM Exact Deconvolution Operator

Another important role in ill-posed inverse problems is played by the condition number of the problem [15, 33, 74]. Indeed, if the condition number is large, then the noise in the input data will generally be highly amplified in the deconvolution process. Thus, in our setting, the condition number of the matrix $\mathbf{M} + \delta^2\, \mathbf{S}$ of the linear system (5.7) needs to be investigated.

From a theoretical point of view, estimating the condition number of $\mathbf{M}+\delta^2\, \mathbf{S}$ is generally impossible, since this matrix depends on the particular POD basis functions used, which in turn depend on the particular physical setting employed. To get some insight into this condition number, however, we use the approach proposed in [32, 38] and consider the case when the POD basis is actually the Fourier basis. (This is the case when the underlying problem is homogeneous [35].) As shown in [38], in this case, $\boldsymbol{M} = \frac{1}{2}\, \mathbf{I}$. Furthermore, since the POD basis functions are $\varphi_j(x) = \sin(j\, \pi\, x)$ (i.e., the Fourier basis functions), the matrix $\boldsymbol{S}$ is diagonal and its diagonal entries are given by $S_{jj} = \frac{1}{2}(j\, \pi)^2$. Finally, since the matrix $(\mathbf{M} + \delta^2\, \mathbf{S})$ is symmetric, its matrix 2-norm is given by the largest eigenvalue. Thus, the condition number of the ROM exact deconvolution, $\mathcal{K}^{ED}$, can be computed as follows:

$$\mathcal{K}^{ED} := \mathcal{K}\Big((\mathbf{M} + \delta^2\, \mathbf{S})^{-1}\Big) = \|(\mathbf{M} + \delta^2\, \mathbf{S})^{-1}\|_2\, \|(\mathbf{M} + \delta^2\, \mathbf{S})\|_2 = \frac{\frac{1}{2} + \frac{1}{2}\, \delta^2\, (r\, \pi)^2}{\frac{1}{2} + \frac{1}{2}\, \delta^2\, (1\, \pi)^2}\, . \tag{5.8}$$

The condition number in (5.8) scales asymptotically as follows:

$$\mathcal{K}^{ED} \sim \begin{cases} r^2 & \text{if } \delta^2 r^2 \text{ large} \\ 1 & \text{if } \delta^2 r^2 \text{ small}\,. \end{cases} \tag{5.9}$$

5.2.3 Numerical Illustion

To investigate whether the ROM exact deconvolution is ill-posed, we use Algorithm 1 (in which noise is not considered) and Algorithm 1' (in which noise is considered). The following parameter choices are made in these two algorithms: The input signal is

$$u(x) = \sin(2\,\pi\,x) + 0.1\sin(4\,\pi\,x) + 0.1\sin(16\,\pi\,x) \tag{5.10}$$

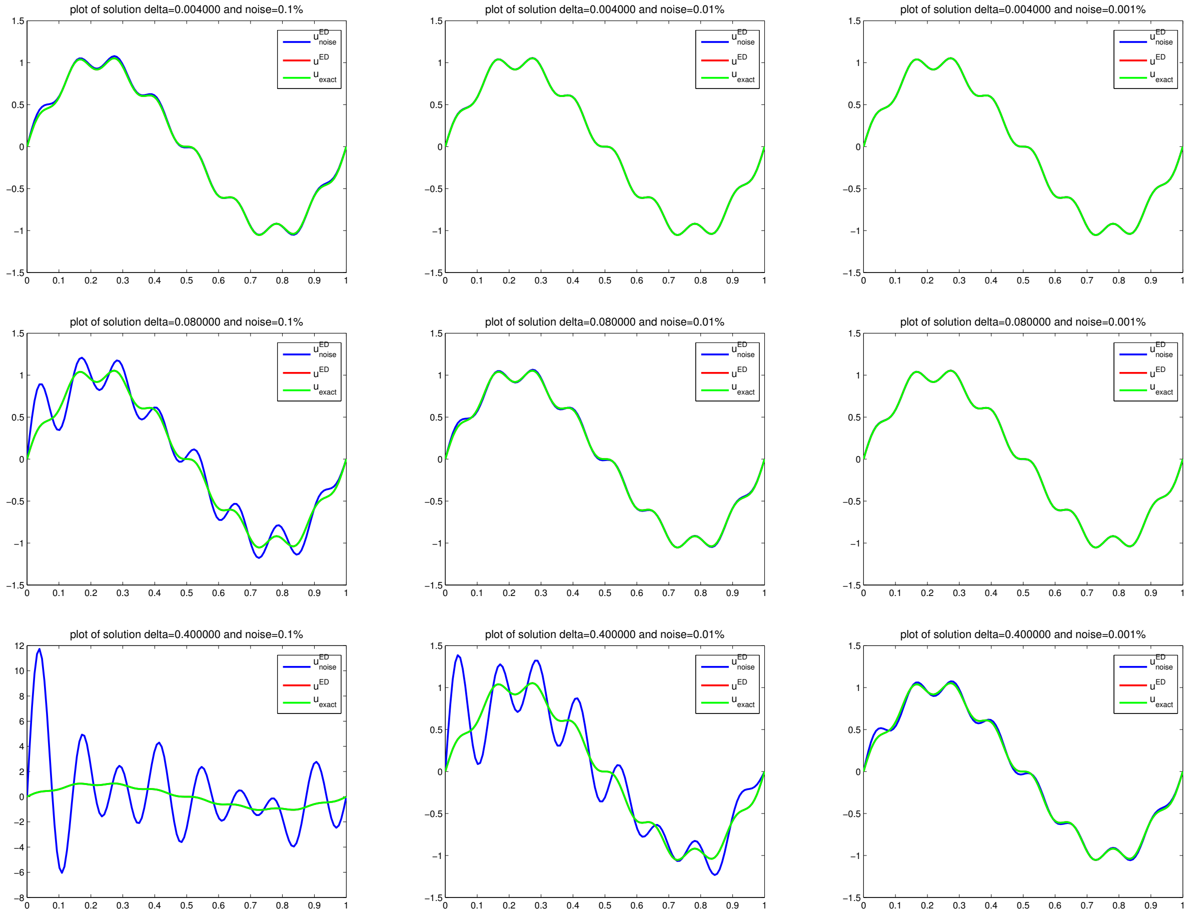
and the POD basis is the Fourier basis $\{\sin(\pi\,x), \sin(2\,\pi\,x), \ldots \sin(16\,\pi\,x)\}$. Furthermore, random noises of three different magnitude orders are considered: $\mathcal{O}(10^{-2})$, $\mathcal{O}(10^{-3})$, $\mathcal{O}(10^{-4})$; the resulting approximations use subscripts n_1, n_2 and n_3, respectively. Specifically, the Matlab command “rand” is first used to generate numbers between 0 and 1; these numbers are then divided by $10^2, 10^3$ and 10^4.

The ROM exact deconvolution approximation without noise (u^{ED}) and the ROM exact deconvolution with the three noise levels ($u^{ED}_{n_1}$, $u^{ED}_{n_2}$ and $u^{ED}_{n_3}$) are plotted in Fig. 5.1 for three representative δ values. As benchmark, the exact input signal (u) is also plotted. These plots yield the following conclusions: First, without noise, the ROM exact deconvolution approximation is accurate (almost indistinguishable from the exact input signal). Second, when noise is added, the ROM exact deconvolution is inaccurate, especially for large δ values and noise levels. This behavior is natural, since estimate (5.8) shows that, for the value $r = 16$ that was used in this section, the condition number of the ROM exact deconvolution problem increases with respect to δ (see sixth column of Table 5.1). These conclusions are reinforced by the errors of u^{ED}, $u^{ED}_{n_1}$, $u^{ED}_{n_2}$ and $u^{ED}_{n_3}$, which are listed in Table 5.1.

Based on the results in Fig. 5.1 and Table 5.1, we conclude that, when noise is added to the filtered input signal, the ROM exact deconvolution increases its magnitude, which results in an inaccurate, oscillatory approximation of the original unfiltered signal. This suggests that the ROM exact deconvolution is ill-posed. We also emphasize that, in the simplified setting considered in this section, the condition number of the ROM exact deconvolution, $\mathcal{K}^{ED}$, is relatively low (see Table 5.1). We expect, however, that $\mathcal{K}^{ED}$ will be significantly larger in realistic fluid flow settings, which are the new AD-ROM's ultimate target. Thus, in these realistic settings, we expect the ill-posedness of the ROM exact deconvolution to be more clearly displayed.

5.3 Regularization Methods for Ill-Posed Inverse Problems

In Section 5.2 it was shown that the ROM exact deconvolution problem is ill-posed. In this section, we briefly describe several numerical methods that are commonly used for general ill-posed inverse problems. These methods will be tested on the ROM exact deconvolution problem (5.2) in Section 5.1.



plot of solution delta=0.004000 and noise=0.1%

plot of solution delta=0.004000 and noise=0.01%

plot of solution delta=0.004000 and noise=0.001%

plot of solution delta=0.080000 and noise=0.1%

plot of solution delta=0.080000 and noise=0.01%

plot of solution delta=0.080000 and noise=0.001%

plot of solution delta=0.400000 and noise=0.1%

plot of solution delta=0.400000 and noise=0.01%

plot of solution delta=0.400000 and noise=0.001%

Figure 5.1: ROM exact deconvolution. Plots of the approximations without noise, i.e., with Algorithm 1 (red curve) and with noise, i.e., with Algorithm 2 (blue curve). Three different δ values are used: $\delta = 0.004$ (top row), $\delta = 0.08$ (middle row) and $\delta = 0.4$ (bottom row). Three different noise magnitude levels are used: $\mathcal{O}(10^{-3})$ (left column), $\mathcal{O}(10^{-4})$ (middle column) and $\mathcal{O}(10^{-5})$ (right column). The exact solution (green curve) is also plotted for comparison purposes.

The ill-posedness of inverse problems is generally manifested in practice in the form of numerical oscillations (which were also observed in the ROM exact deconvolution in Section 5.2). Thus, different *regularization methods* have been proposed to alleviate the numerical oscillations displayed by ill-posed inverse problems. Next, we briefly summarize the most popular regularization methods. More details can be found in, e.g., [15, 33, 74].

There are two types of regularization methods: *variational* regularization methods, in which one solves a minimization problem (see Section 5.3.1), and *iterative* regularization methods, in which one uses a fixed-point iteration to approximate the solution (see Section 5.3.2).

		$\mathcal{O}(10^{-3})$ noise	$\mathcal{O}(10^{-4})$ noise	$\mathcal{O}(10^{-5})$ noise	
δ	$\|u - u^{ED}\|_{L^2}$	$\|u - u^{ED}_{n1}\|_{L^2}$	$\|u - u^{ED}_{n2}\|_{L^2}$	$\|u - u^{ED}_{n3}\|_{L^2}$	$\mathcal{K}^{ED}$
0.004	9.1451e-04	0.0167	0.0021	9.7247e-04	1.01
0.08	9.1451e-04	0.1377	0.0126	0.0018	15.93
0.4	9.1451e-04	3.0670	0.2743	0.0278	155.13

Table 5.1: ROM exact deconvolution. Errors without noise, i.e., with Algorithm 1 (second column) and with noise, i.e., with Algorithm 2 (third, fourth and fifth columns). Three different δ values and three different different noise magnitude levels are used. The condition number is also listed (sixth column).

5.3.1 Variational Regularization Methods

Among the variational regularization methods, the Tikhonov and Lavrentiev regularization methods are commonly used.

Tikhonov regularization method This is one of the most popular regularizations for ill-posed problems [15, 33, 74]. The Tikhonov regularization method minimizes the following functional (see equation (5.6) in [15]):

$$\Phi_\mu(u) := \|G\,u - \overline{u}\|^2 + \mu\,\|u\|^2\,, \tag{5.11}$$

which is a linear combination of the discrepancy functional $\|G\,u - \overline{u}\|^2$ and the energy functional $\|u\|^2$. Problem (5.11) aims at minimizing the discrepancy functional such that the solution has a prescribed energy (see Fig. 5.1 in [15]). The Tikhonov regularization (5.11) is a variational algorithm which is equivalent to the following linear system [15, 74]:

$$(G^*\,G\, + \mu\,I)\,u = G^*\,\overline{u}\,. \tag{5.12}$$

Thus, to solve the minimization problem in (5.11), one simply needs to compute

$$u = (G^*\,G\, + \mu\,I)^{-1} G^*\,\overline{u}\,. \tag{5.13}$$

The parameter μ in (5.11) is a free parameter. Several approaches for finding its optimal value are discussed in [15, 33, 74]. For large scale problems, instead of solving the linear system (5.12), one can instead use an iterative procedure, i.e., the Tikhonov iterative regularization.

Lavrentiev regularization method This is the Tikhonov regularization method, but without the adjoint operator G^*. Specifically, the Lavrentiev regularization method replaces the linear system in (5.12) with the following linear system (see (12.3) in [15]):

$$(G\, + \mu\,I)\,u = \overline{u}\,. \tag{5.14}$$

An improvement to the Lavrentiev regularization method is the *modified Lavrentiev regularization method* [47, 67]:

$$((1-\mu)\,G\, + \mu\,I)\,u = \overline{u}\,. \tag{5.15}$$

5.3.2 Iterative Regularization Methods

Among the iterative regularization methods, the Landweber and Van Cittert regularization methods are commonly used.

Landweber iterative method This method minimizes the following functional (see Appendix E in [15]):

$$\Phi(u) := \|G\, u - \overline{u}\|^2 \,, \tag{5.16}$$

i.e., it minimizes the discrepancy functional $\|G\, u - \overline{u}\|^2$, which is equivalent to the following linear system [15]:

$$(G^* \, G)\, u = G^* \, \overline{u} \,. \tag{5.17}$$

For large scale problems, one can instead use the Landweber iterative method

$$u^{n+1} = u^n + \tau \left[G^* \, \overline{u} - (G^* \, G)\, u^n \right] , \tag{5.18}$$

where τ is a relaxation parameter and the number of iterations plays the role of a regularization parameter.

Van Cittert iterative method This method is defined as follows [15]:

$$u^{n+1} = u^n + \tau \left(\overline{u} - G\, u^n \right) . \tag{5.19}$$

The Van Cittert iterative method is the most popular regularization method in AD-LES [69, 71, 72, 45, 47].

We emphasize that our LES-ROM framework is different from the standard LES setting: The dimension of the problem is small for the LES-ROM framework ($\mathcal{O}(10)$), but very large for the LES setting ($\mathcal{O}(10^6)$). Thus, it is not clear which regularization method is the most appropriate for our LES-ROM framework. This issue is discussed in Section 5.4.

5.4 ROM Approximate Deconvolution with Lavrentiev Regularization

In Section 5.2 it was shown that the ROM exact deconvolution problem (5.2) is ill-posed. Indeed, using Algorithm 2 showed that making small changes in (i.e., adding noise to) the input filtered signal yields nonphysical numerical oscillations. This behavior, which is typical to ill-posed inverse problems, is generally mitigated by using some of the regularization methods outlined in Section 5.3. In this section, we use these regularization methods to solve the ROM deconvolution problem. Following the LES terminology [13, 47], we call ROM approximate deconvolution the regularized methods applied to the ROM deconvolution problem. Just as is usually done in ill-posed inverse problems [15, 74], we investigate whether Algorithm 3 (presented below) yields more accurate results than Algorithm 2, i.e., whether the ROM approximate deconvolution is more accurate than the ROM exact deconvolution.

Algorithm 3

(1) Consider input signal u.

(2) Calculate the filtered input signal $\overline{u} := G\, u$, where G is the DF.

(3) Approximately deconvolve the noisy input signal $u^{AD} \approx G^{-1}\,(\overline{u} + \eta)$, where η is the noise.

(4) Compare the AD signal u^{AD} with the true signal u and the ED signal u^{ED} from Algorithm 2.

As in Section 5.2, the effect of noise on the filtered input signal is studied in Step 3 of Algorithm 3. As mentioned before, this is relevant to the numerical discretization of the new AD-ROM (5.5), where noise could model the numerical error in the approximation of the filtered flow variable, $\overline{\boldsymbol{u}}_r$.

Several regularization methods described in Section 5.3 (Lavrentiev, modified Lavrentiev, and Van Cittert) were investigated numerically in Step 3 of Algorithm 3. Since they yielded similar qualitative results in our preliminary tests, for clarity, we exclusively consider the Lavrentiev regularization method (5.14) to find u^{AD} in Step 3 of Algorithm 3:

$$(G + \mu\, I)\, u^{AD-L} = \overline{u}\,, \tag{5.20}$$

where u^{AD-L} is the AD approximation of u obtained with the Lavrentiev regularization method. Since G is the DF (4.2), equation (5.20) becomes

$$\left(\left(I - \delta^2\, \Delta \right)^{-1} + \mu\, I \right) u^{AD-L} = \overline{u}\,. \tag{5.21}$$

Multiplying (5.21) by $\left(I - \delta^2\, \Delta \right)$, we get

$$u^{AD-L} + \mu \left(I - \delta^2\, \Delta \right) u^{AD-L} = \left(I - \delta^2\, \Delta \right) \overline{u}\,, \tag{5.22}$$

which in matrix form can be written as

$$\left(\boldsymbol{M} + \mu\, \boldsymbol{M} + \mu\, \delta^2\, \boldsymbol{S} \right) u^{AD-L} = \left(\boldsymbol{M} + \delta^2\, \boldsymbol{S} \right) \overline{u}\,. \tag{5.23}$$

In Step 4 of Algorithm 3, the AD signal (u^{AD-L}) is compared with the true signal (u) and the ED signal (u^{ED}) from Step 3 of Algorithm 2. We expect that the regularization used in the computation of u^{AD-L} will make it more accurate than u^{ED}.

5.4.1 Numerical Results – ”Boxy” Components

In Section 5.2.3, we investigated whether the ROM approximate deconvolution is more accurate than the ROM exact deconvolution for exact Fourier components. In this section, we apply Algorithm 2 and Algorithm 3 to ”boxy” components, which are defined in (5.24).

$$u_{true}(x) = \begin{cases} 0 & x \in [0, \frac{1}{6}), [\frac{5}{6}, 1) \\ 1 & x \in [\frac{1}{6}, \frac{2}{6}), [\frac{3}{6}, \frac{4}{6}) \\ -1 & x \in [\frac{2}{6}, \frac{3}{6}), [\frac{4}{6}, \frac{5}{6}) \end{cases} \tag{5.24}$$

The results in Tables 5.2– 5.4 and Fig. 5.2 are consistent with the results in Section 5.2.3: the ROM approximate deconvolution is more accurate than the ROM exact deconvolution.

It is clear that for this problem, the ROM approximate deconvolution with Lavrentiev regularization in Algorithm 3 works better than the ROM exact deconvolution with noise. Also from the plot 5.2, the noise magnification is higher when δ is large.

	$\|u - u^{ED}\|_{L^2}$	$\|u - u_n^{ED}\|_{L^2}$	$\|u - u^{AD-L}\|$	$M_r + \delta^2 S_r$
$\delta = 0.004$	0.3122	0.3123	0.3123	1.00
$\delta = 0.08$	0.3122	0.3127	0.3129	4.73
$\delta = 0.4$	0.3122	0.8899	0.6159	39.45

Table 5.2: ROM exact deconvolution and ROM approximate deconvolution with Lavrentiev regularization for $r = 8$. Errors of u_{ED} without noise (second column), u_{ED} with random noise of magnitude level $\mathcal{O}(10^{-2})$(third column) and u_{AD-L} with the same random noise (fourth column). The condition numbers of the ROM exact deconvolution $\mathcal{K}^{AD-L}$ are also listed (fifth column). Three different δ values are used

.

	$\|u - u^{ED}\|_{L^2}$	$\|u - u_n^{ED}\|_{L^2}$	$\|u - u^{AD-L}\|$	$M_r + \delta^2 S_r$
$\delta = 0.004$	0.2944	0.2948	0.2947	1.01
$\delta = 0.08$	0.2944	0.3164	0.3133	15.93
$\delta = 0.4$	0.2944	2.8721	1.1135	155.13

Table 5.3: ROM exact deconvolution and ROM approximate deconvolution with Lavrentiev regularization for $r = 16$. Errors of u_{ED} without noise (second column), u_{ED} with random noise of magnitude level $\mathcal{O}(10^{-2})$(third column) and u_{AD-L} with the same random noise (fourth column). The condition numbers of the ROM exact deconvolution $\mathcal{K}^{AD-L}$ are also listed (fifth column). Three different δ values are used

.

	$\|u-u^{ED}\|_{L^2}$	$\|u-u_n^{ED}\|_{L^2}$	$\|u-u^{AD-L}\|$	$M_r+\delta^2 S_r$
$\delta=0.004$	0.1663	0.1667	0.1667	1.04
$\delta=0.08$	0.1663	0.4761	0.4049	51.96
$\delta=0.4$	0.1663	16.1452	2.6660	526.99

Table 5.4: ROM exact deconvolution and ROM approximate deconvolution with Lavrentiev regularization for $r=30$. Errors of u_{ED} without noise (second column), u_{ED} with random noise of magnitude level $\mathcal{O}(10^{-2})$(third column) and u_{AD-L} with the same random noise (fourth column). The condition numbers of the ROM exact deconvolution $\mathcal{K}^{AD-L}$ are also listed (fifth column). Three different δ values are used

.

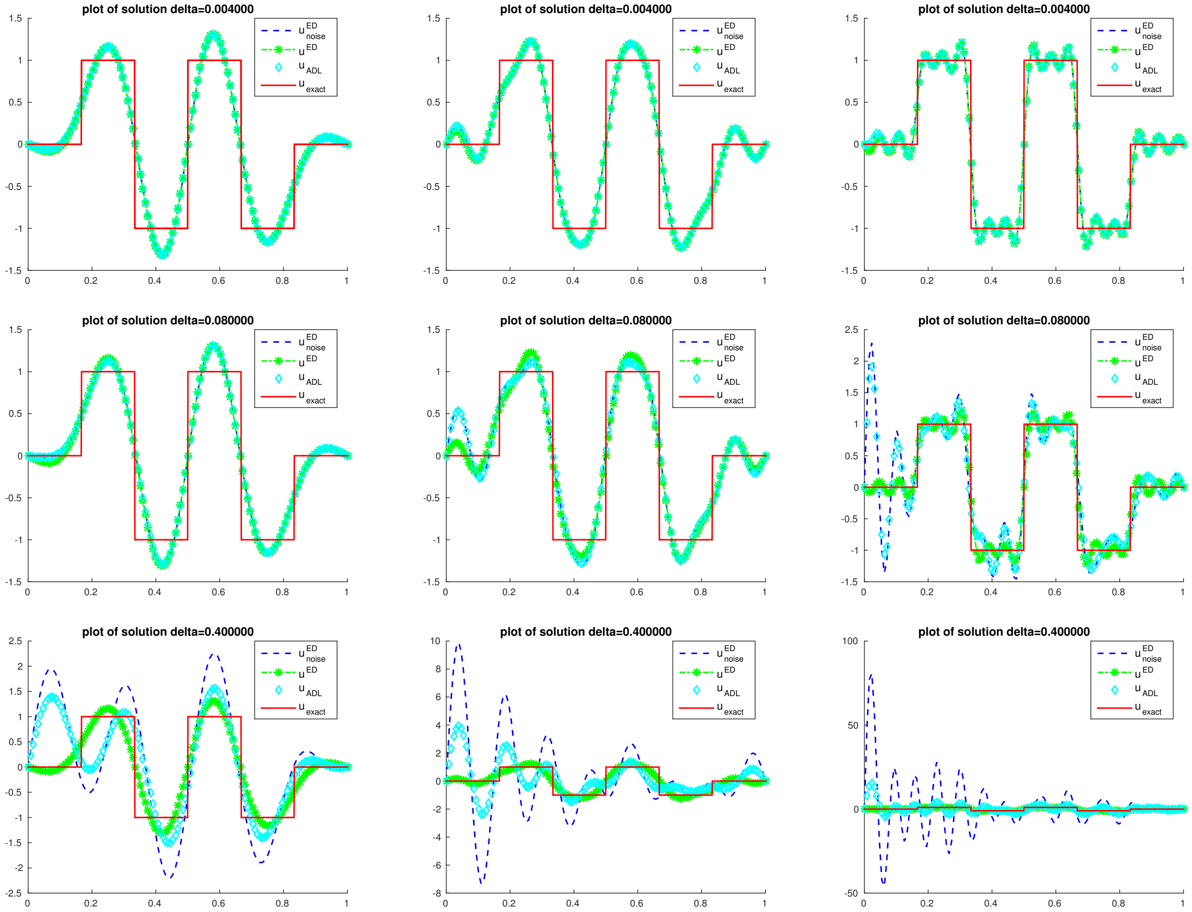


Figure 5.2: ROM exact deconvolution (u_{ED}, blue curve) and ROM approximate deconvolution with Lavrentiev regularization (u_{AD-L}, red curve) of a noisy filtered signal. A random noise with magnitude level $\mathcal{O}(10^{-2})$ is used. Three different δ values and three different r values are used: $\delta=0.004$ (top row), $\delta=0.08$ (middle row), $\delta=0.04$ (bottom row), $r=8$ (left column), $r=16$ (middle column) and $r=30$ (right column). The exact solution (u, green curve) is also plotted for comparison purposes

.

5.4.2 Numerical Results – Fourier Components

As in Section 5.2, Algorithm 2 is used with the following parameter choices: The input signal is the same as in (5.10) and the POD basis is $\{\sin(\pi\, x), \sin(2\,\pi\, x), \ldots \sin(16\,\pi\, x)\}$, i.e., the Fourier basis. A random noise with magnitude order $\mathcal{O}(10^{-2})$ is considered in Step 3 of Algorithm 2. (For clarity, results for the other two noise levels considered in Section 5.2 are not presented in this section.) The ROM exact deconvolution approximation (u^{ED}) and the ROM approximate deconvolution approximation with Lavrentiev regularization (u^{AD-L}) are plotted in Fig. 5.3 for three representative δ values and three parameter μ values in the Lavrentiev regularization. As benchmark, the exact input signal (u) is also plotted. The plots in Fig. 5.3 show that, as δ increases, u^{AD-L} becomes significantly more accurate than u^{ED}. Furthermore, larger μ values (i.e., larger regularization levels in the Lavrentiev regularization) increase the u^{AD-L} accuracy. These conclusions are reinforced by the errors of u^{ED} and u^{AD-L} listed in Table 5.5.

To explain the increase in accuracy of u^{AD-L} over u^{ED}, the ROM exact deconvolution condition number ($\mathcal{K}^{ED}$ defined in (5.8))and ROM approximate deconvolution with Lavrentiev regularization condition number $\mathcal{K}^{AD-L} := \mathcal{K}(\boldsymbol{M} + \mu\, \boldsymbol{M} + \mu\, \delta^2\, \boldsymbol{S})$ are listed in Table 5.6. As δ increases, $\mathcal{K}^{AD-L}$ gets smaller than $\mathcal{K}^{ED}$. Thus, the noise in the input filter signal is amplified more in the ROM exact deconvolution than in the ROM approximate deconvolution.

Given the scaling in (5.9), these numerical results are natural: an increase in δ results in an increase in $\mathcal{K}^{ED}$, which in turn triggers an amplification of the input noise in the ROM exact deconvolution. The same scaling in (5.9) suggests that increasing r will have an effect similar to that of increasing δ. To test this conjecture, we consider the signal

$$u(x) = \sin(\pi\, x) + 0.1 \sin(50\,\pi\, x) + 0.1 \sin(100\,\pi\, x) \tag{5.25}$$

and the POD basis is taken to be the Fourier basis $\{\sin(\pi\, x), \sin(2\,\pi\, x), \ldots \sin(100\,\pi\, x)\}$. Thus, we replaced $r = 16$ with $r = 100$ in our analytical example. In this case, $\mathcal{K}^{ED} = 350$, whereas $\mathcal{K}^{AD-L} = 34.02$. Furthermore, the u^{ED} error is 11.0582, which is more than 40 times larger than the u^{AD-L} error, which is 0.3113. These results (see also Fig. 5.4) support our conjecture that increasing r will have an effect similar to that of increasing δ.

Based on the results in Fig. 5.3, Table 5.5 and Table 5.6, we conclude that AD-L is more accurate than ED.

		$\mu = 0.1$	$\mu = 0.01$	$\mu = 0.001$
	$\Vert u - u^{ED}\Vert_{L^2}$	$\Vert u - u^{AD-L}\Vert_{L^2}$	$\Vert u - u^{AD-L}\Vert_{L^2}$	$\Vert u - u^{AD-L}\Vert_{L^2}$
$\delta = 0.004$	0.0167	0.0598	0.0148	0.0164
$\delta = 0.08$	0.1214	0.0924	0.1035	0.1193
$\delta = 0.4$	2.7389	0.2999	0.7749	2.1214

Table 5.5: ROM exact deconvolution and ROM approximate deconvolution with Lavrentiev regularization. A random noise with magnitude level $\mathcal{O}(10^{-2})$ is used. Errors of u^{ED} (second column) and u^{AD-L} for three different μ values: $\mu = 0.1$ (third column), $\mu = 0.01$ (fourth column) and $\mu = 0.001$ (fifth column). Three different δ values are used.

	$\mathcal{K}^{ED}$	$\mathcal{K}^{AD-L}$, $\mu = 0.1$	$\mathcal{K}^{AD-L}$, $\mu = 0.01$	$\mathcal{K}^{AD-L}$, $\mu = 0.001$
$\delta = 0.004$	1.01	1.02	1.02	1.02
$\delta = 0.08$	15.93	2.41	1.13	1.00
$\delta = 0.4$	155.13	32.58	4.85	1.31

Table 5.6: Condition numbers for the ROM exact deconvolution ($\mathcal{K}^{ED}$) and ROM approximate deconvolution with Lavrentiev regularization ($\mathcal{K}^{AD-L}$). A random noise with magnitude level $\mathcal{O}(10^{-2})$ is used. Three δ values and three μ values are used.

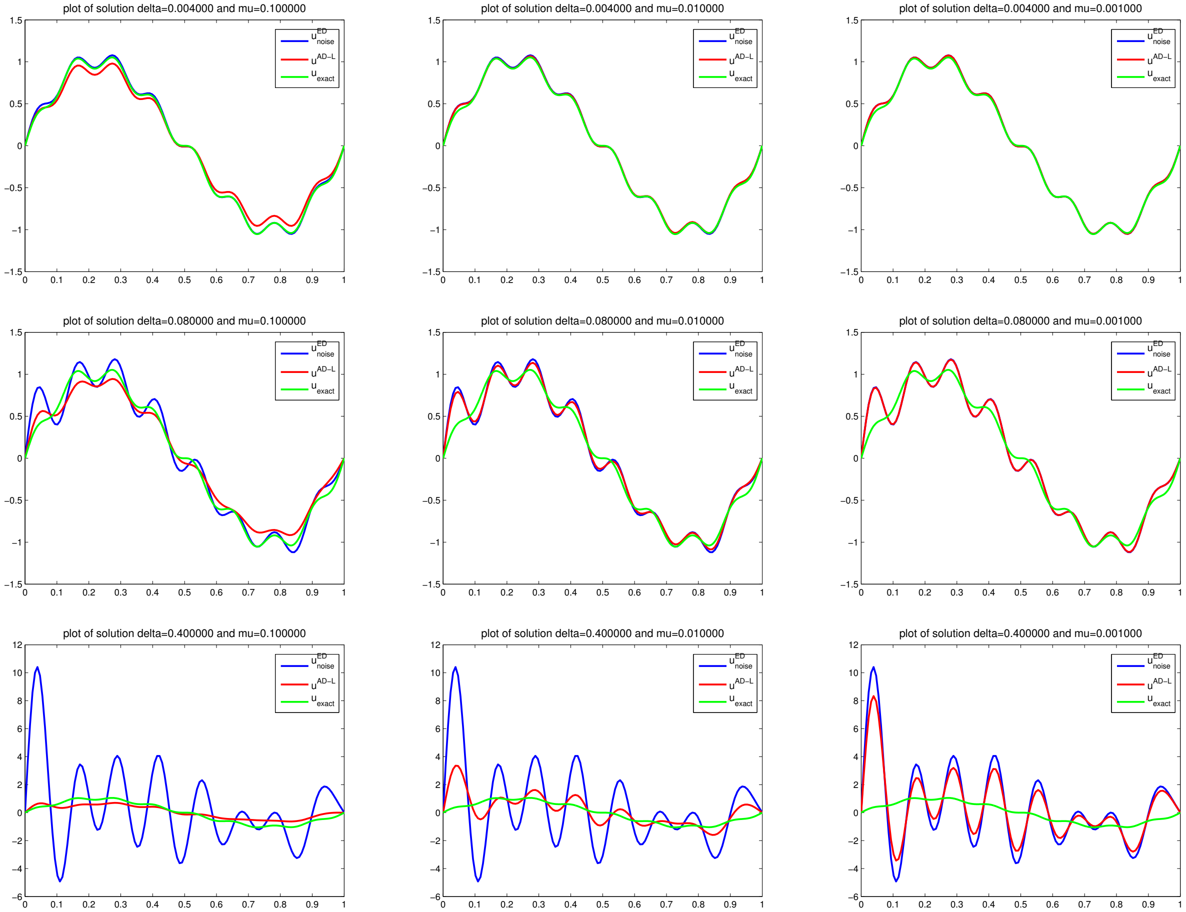


Figure 5.3: ROM exact deconvolution (u^{ED}, blue curve) and ROM approximate deconvolution with Lavrentiev regularization (u^{AD-L}, red curve) of a noisy filtered signal. A random noise with magnitude level $\mathcal{O}(10^{-2})$ is used. Three different δ values and three different μ values are used: $\delta = 0.004$ (top row), $\delta = 0.08$ (middle row), $\delta = 0.04$ (bottom row), $\mu = 0.1$ (left column), $\mu = 0.01$ (middle column) and $\mu = 0.001$ (right column). The exact solution (u, green curve) is also plotted for comparison purposes.

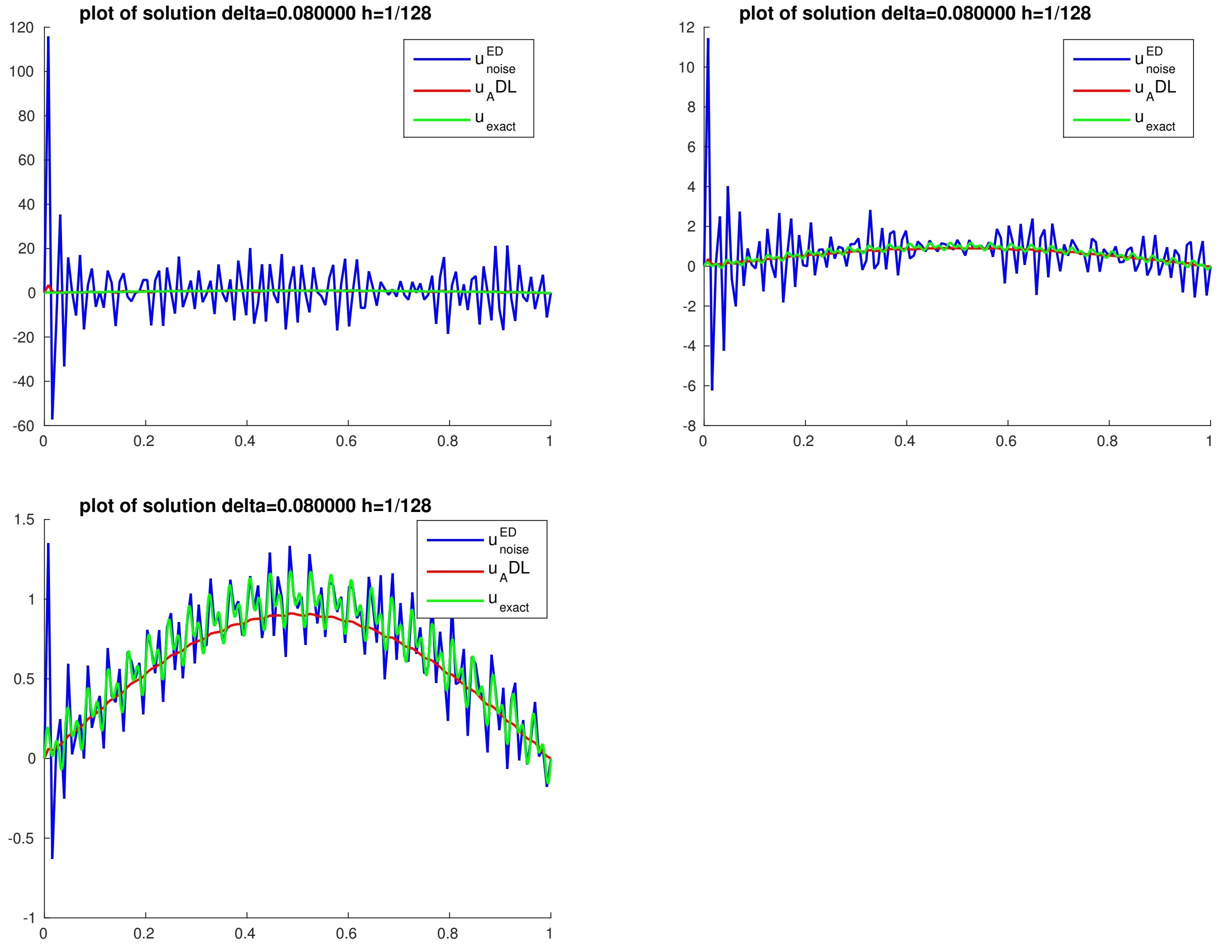


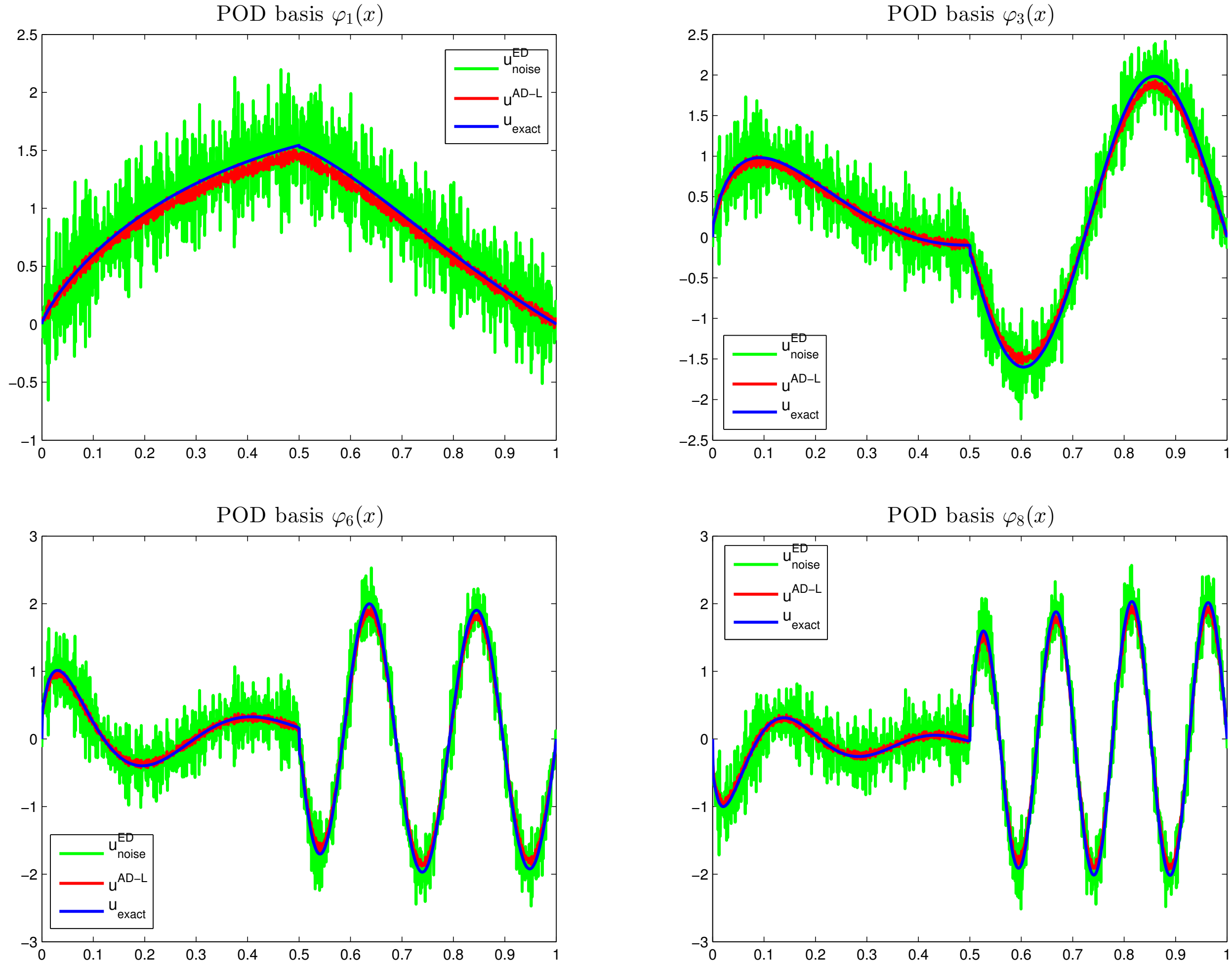
Figure 5.4: ROM exact deconvolution (u^{ED}, blue curve) and ROM approximate deconvolution with Lavrentiev regularization (u^{AD-L}, red curve) of a noisy filtered signal (5.25). $r = 100$, $\delta = 0.08$ are used. Three different level of random noise are used: $\mathcal{O}(10^{-2})$ (top left), $\mathcal{O}(10^{-2})$ (top right) and $\mathcal{O}(10^{-4})$ (bottom left). The exact solution (u, green curve) is also plotted for comparison purposes.

5.4.3 Numerical Results – POD basis

In the previous sections, we investigated whether the ROM approximate deconvolution is more accurate than the ROM exact deconvolution for exact Fourier components (Section 5.4.2) and ”boxy” components (Section 5.4.1). Note, however, that to develop the AD-ROM, one needs to apply the ROM deconvolution to the POD basis. Thus, in this section, we apply Algorithm 2 and Algorithm 3 to POD basis functions generated by the Burgers equation. The results in Table 5.7, Fig. 5.5 and Fig. 5.6 are consistent with the results in the previous sections: the ROM approximate deconvolution is more accurate than the ROM exact deconvolution.

μ		0.006	0.06	0.6
	$\|u-u_n^{ED}\|_{L^2}$	$\|u-u^{AD-L}\|_{L^2}$	$\|u-u^{AD-L}\|_{L^2}$	$\|u-u^{AD-L}\|_{L^2}$
$\delta = 0.0004$	0.0062	0.0088	0.0572	0.3752
$\delta = 0.004$	0.1814	0.0995	0.0627	0.3788
$\delta = 0.04$	17.98	0.3276	0.1405	0.5814

Table 5.7: ROM exact deconvolution and ROM approximate deconvolution with Lavrentiev regularization for POD mode $\varphi_6(x)$. A random noise with magnitude level $\mathcal{O}(10^{-2})$ is used. Errors of u^{ED} (second column) and u^{AD-L} for three different μ values: $\mu = 0.006$ (third column), $\mu = 0.06$ (fourth column) and $\mu = 0.6$ (fifth column). Three different δ values are used: $\delta = 0.0004$ (third row), $\delta = 0.004$ (fourth row), $\delta = 0.04$ (fifth row).


POD basis $\varphi_1(x)$

POD basis $\varphi_3(x)$

POD basis $\varphi_6(x)$

POD basis $\varphi_8(x)$

Figure 5.5: ROM exact deconvolution (u_{ED}, green curve) and ROM approximate deconvolution with Lavrentiev regularization (u_{AD-L}, red curve) of a noisy filtered POD basis functions. A random noise with magnitude level $\mathcal{O}(10^{-2})$ is used. Four different POD basis functions are tested: φ_1 (top left), φ_3 (top right), φ_6 (bottom left), φ_8 (bottom right). The The exact solution (u, blue curve) is also plotted for comparison purposes.

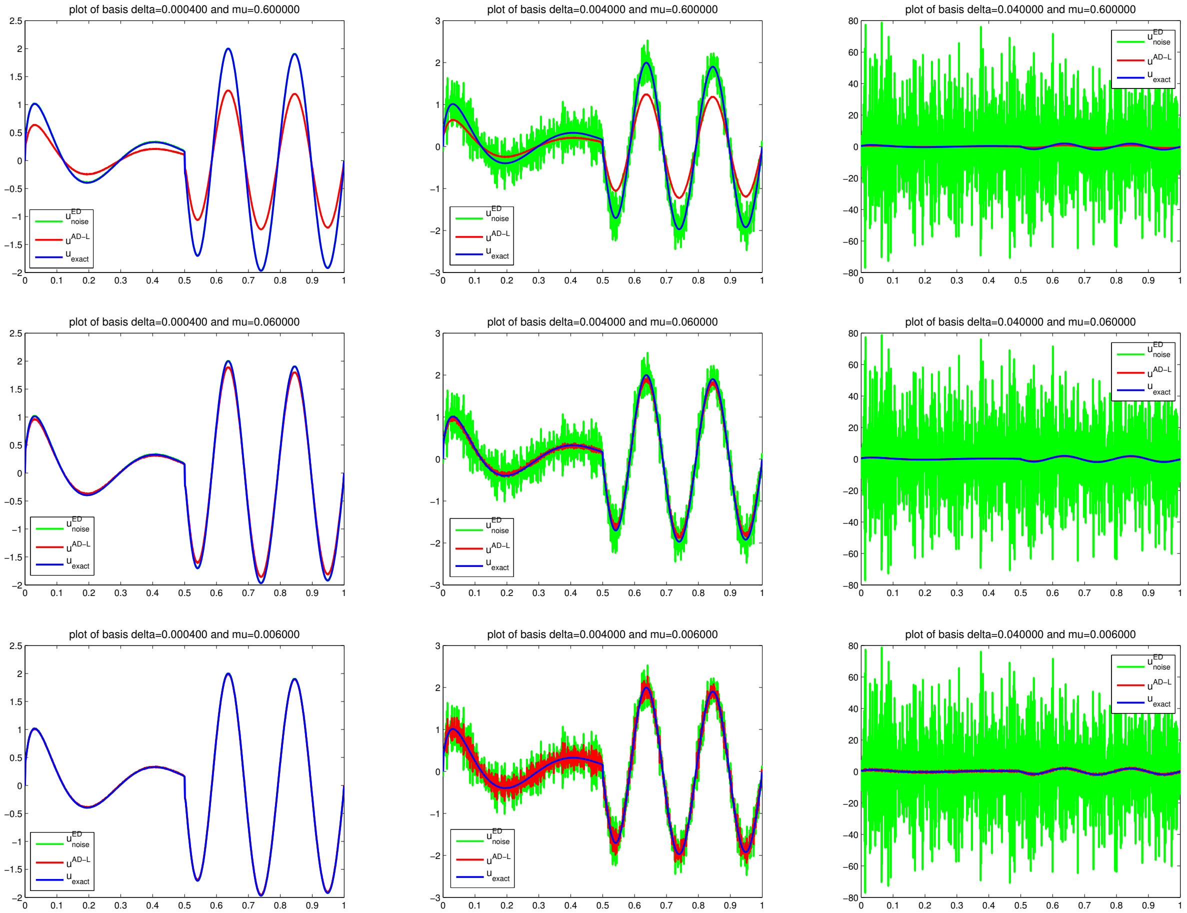


Figure 5.6: ROM exact deconvolution (u_{ED}, green curve) and ROM approximate deconvolution with Lavrentiev regularization (u_{AD-L}, red curve) of a noisy filtered POD basis functions $\varphi_6(x)$. A random noise with magnitude level $\mathcal{O}(10^{-2})$ is used in the middle and right columns. Three different δ values and three different μ values are used: $\delta = 0.0004$ (left column), $\delta = 0.004$ (middle column), $\delta = 0.04$ (right column), $\mu = 0.6$ (top row), $\mu = 0.06$ (middle row) and $\mu = 0.006$ (bottom row). The exact solution (u, blue curve) is also plotted for comparison purposes.

5.5 AD-ROM Discretization

For a given input signal, we showed that the ROM exact deconvolution is an ill-posed problem that yields highly oscillatory results (Section 5.2), whereas the ROM approximate deconvolution with Lavrentiev regularization yields accurate results (Section 5.4). In this section, we compare the ROM exact deconvolution and the ROM approximate deconvolution in the numerical discretization of the actual AD-ROM (5.5). To this end, the deconvolution variable $\boldsymbol{w}_r^{AD}$ in the AD-ROM is taken to be (i) the ROM exact deconvolution in Algorithm 2, and (ii) the ROM approximate deconvolution in Algorithm 3. By convention, in the remainder of the thesis, the AD-ROM (5.5) is called ED-ROM in case (i) and AD-ROM in case (ii). Specifically, for the forward Euler time discretization used in this thesis, the following algorithm needs to be used at each time step for ED-ROM and AD-ROM: Given $\boldsymbol{w}_r^n$, which is the approximation of $\overline{\boldsymbol{u}}_r$ at the current time step n, find $\boldsymbol{w}_r^{n+1}$, which is the approximation of $\overline{\boldsymbol{u}}_r$ at the next time step $n+1$ as follows:

Algorithm 4

(1)

$$\boldsymbol{w}_r^{D,n} = \begin{cases} \boldsymbol{w}_r^{ED,n} = (\boldsymbol{M} + \delta^2\, \boldsymbol{S})\,(\boldsymbol{w}_r^n + \boldsymbol{\eta}) & \text{for ED-ROM} \\ \boldsymbol{w}_r^{AD-L,n} = (\boldsymbol{M} + \mu\, \boldsymbol{M} + \mu\, \delta^2\, \boldsymbol{S})^{-1}(\boldsymbol{M} + \delta^2\, \boldsymbol{S})\,(\boldsymbol{w}_r^n + \boldsymbol{\eta}) & \text{for AD-ROM}\,. \end{cases} \tag{5.26}$$

(2)

$$\left(\frac{\boldsymbol{w}_r^{n+1} - \boldsymbol{w}_r^n}{\Delta t}, \boldsymbol{\varphi}_k\right) + Re^{-1}\,(\nabla \boldsymbol{w}_r, \nabla \boldsymbol{\varphi}_k) + \left(\overline{\boldsymbol{w}_r^{D,n}\,\boldsymbol{w}_r^{D,n}}, \nabla \boldsymbol{\varphi}_k\right) = 0\,, \tag{5.27}$$

where $\boldsymbol{\eta}$ is the noise and Δt is the time step.

For both the ED-ROM and AD-ROM, we need to filter the initial condition: $\boldsymbol{w}_r(\mathbf{x}, 0) = \overline{\boldsymbol{u}_0}(\mathbf{x})$. We also need to filter the boundary conditions. Since we are using the DF (4.2), the boundary conditions remain homogeneous Dirichlet: $\boldsymbol{w}_r(\mathbf{x}, t) = \mathbf{0}$ on the boundary.

To compare the AD-ROM with the ED-ROM, as in Sections 5.2 and 5.4, noise of two different magnitudes ($\mathcal{O}(10^{-2})$ and $\mathcal{O}(10^{-3})$) is used in Algorithm 4.

5.5.1 Noise Modeling AD-ROM Discretization Error

We emphasize that the addition of noise to Algorithm 4 is relevant to ROM of *realistic* flows, e.g., structure dominated turbulent flows [35, 52, 53, 55], which represent the ultimate target of the new AD-ROM. In these realistic settings, noise could model inaccuracies in the data (e.g., forcing term or boundary conditions) or numerical inaccuracies. In this thesis, we exclusively consider that noise models numerical inaccuracies (although the other scenario is equally important). We note, however, that the numerical investigation in Section **??** is carried out for the 1D Burgers equation. In this simple setting, we can afford to use a fine spatial and temporal resolution and a relatively high number of POD modes. Thus, the numerical error is small in this case. However, to get some insight into the new AD-ROM's performance in realistic flows, in which the numerical error is large, we add noise to Algorithm 4. We emphasize, however, that we ensure that the magnitude of the noise added to Algorithm 4 is of the same (or lower) order as the magnitude of the numerical discretization error in the AD-ROM.

Indeed, the total discretization error in the AD-ROM has three main components [37, 38, 39]: the time discretization error (which, for the forward Euler method used in this section, is $\mathcal{O}(\Delta t) = \mathcal{O}(10^{-4})$), the spatial discretization error (which, for the linear FE method used in this section, is $\mathcal{O}(h^2) = \mathcal{O}(10^{-6})$), and the POD truncation error (which, for the r values used in this section, is $\mathcal{O}(\sqrt{\sum_{i=r+1}^{d} \lambda_i}) = \mathcal{O}(10^{-2})$ or lower). Thus, the magnitude of AD-ROM's total discretization error is $\mathcal{O}(10^{-2})$, which is of the same order as (or higher than) the magnitude of the noise used in Algorithm 4.

Chapter 6

Numerical Experiments for AD-ROM

In this the section, we compare the AD-ROM with the ED-ROM. For comparison purposes, results for the G-ROM and DNS are also included. Numerical results are presented for the Burgers equation (Section 6.1).

6.1 Burgers Equation

The mathematical model used in this section is the (1D) Burgers equation

$$\begin{cases} u_t - \nu\, u_{xx} + u\, u_x = 0 & x \in [0,1], \\ u(x,0) = u_0(x) & x \in [0,1], \\ u(0,t) = u(1,t) = 0\,, \end{cases} \tag{6.1}$$

where $\nu = 10^{-3}$ is the diffusion parameter and $t \in [0,1]$. The initial condition is $u_0(x) = 1$ for $x \in (0, \frac{1}{2}]$ and $u_0(x) = 0$ for $x \in (\frac{1}{2}, 1)$. This computational setting is similar to that used in [43, 80].

For the DNS, a uniform mesh with $h = 1/1024$ and a linear FE were used for the spatial discretization and the forward Euler method with a time step $\Delta t = 10^{-4}$ were used for the time discretization. For ROMs, the forward Euler method with a time step $\Delta t = 10^{-4}$ were used for the time discretization. A total of 101 snapshots were collected and the following r values were used: $r = 5, 10, 15$ and 20.

For all the parameters used in the numerical investigation, the CPU times of the ED-ROM and AD-ROM were significantly lower that the CPU time of the corresponding DNS. The DNS took about 200 seconds to run. In contrast, the CPU times of the G-ROM and the ED-ROM and AD-ROM were on the order of 10 seconds.

6.2 AD-ROM with Lavrentiev Regularization

The ED-ROM and AD-ROM errors are listed in Table 6.1 for four r values, two δ values and two noise magnitudes. The optimal μ values were used in the AD-ROM. For comparison purposes, G-ROM errors are also listed. To compute the errors, the ED-ROM and AD-ROM results are compared with the filtered DNS data, whereas the G-ROM results are compared with the unfiltered DNS data. The results in Table 6.1 show that the AD-ROM is consistently more accurate than the ED-ROM. In particular, for large r values, large δ values and large noise levels, the AD-ROM error is as much as five times lower than the ED-ROM error. This is due to the fact that $\mathcal{K}^{AD-L}$ is significantly lower than $\mathcal{K}^{ED}$, especially for high δ and r values (see Table 6.2).

We note that the G-ROM is less accurate than the ED-ROM and AD-ROM for $r = 5$ and $r = 10$, but more accurate for the larger r values. We emphasize, however, that the G-ROM yields accurate results for the relatively simple Burgers equation considered in this section. For realistic flows, significantly higher r values are generally needed [53, 55] and, thus, we expect the ED-ROM and AD-ROM to perform significantly better than G-ROM [78].

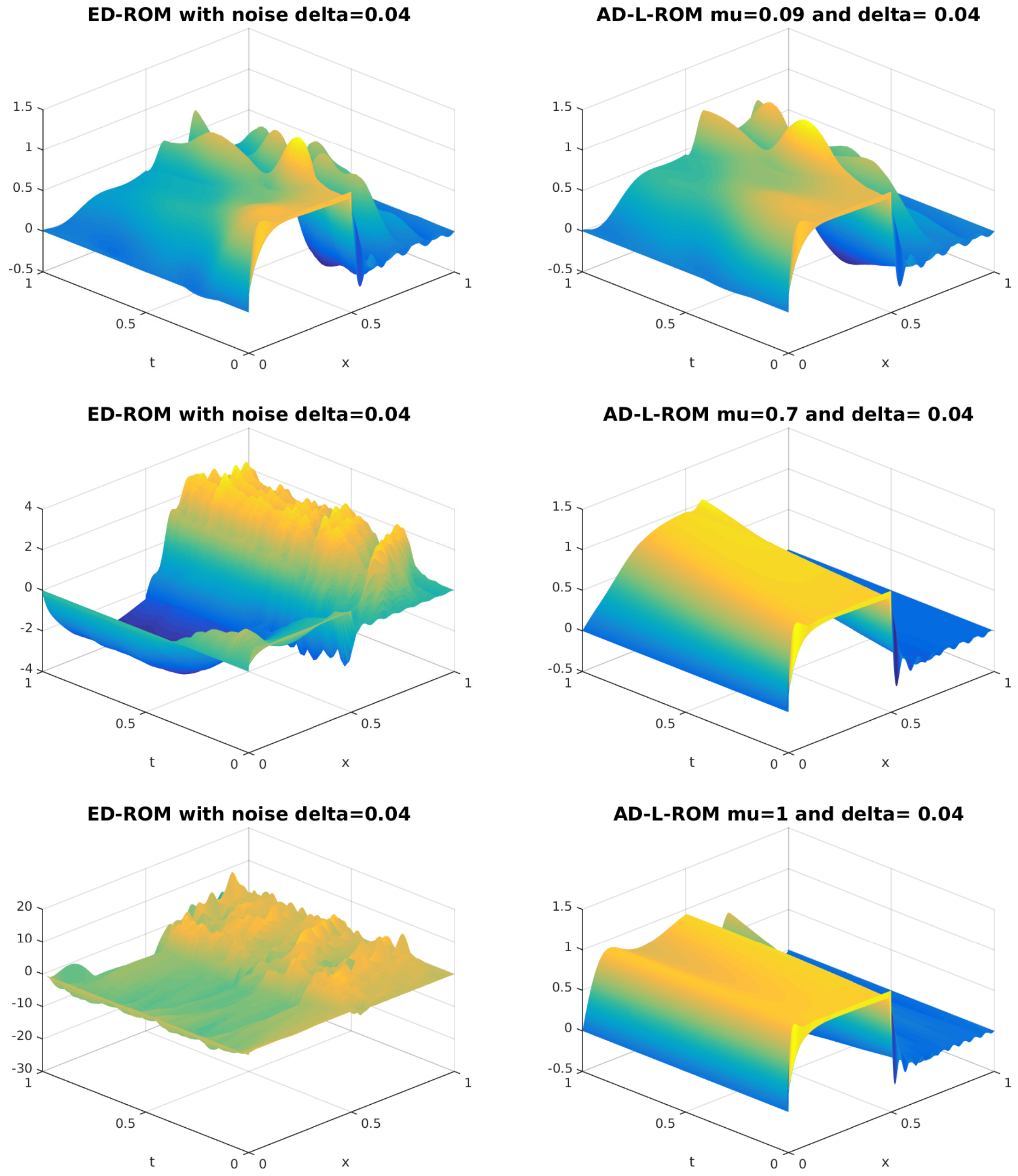
The results in Table 6.1 are supported by the plots in Fig. 6.1.

r	δ	G-ROM		ED-ROM		AD-ROM	
		$\mathcal{O}(10^{-2})$	$\mathcal{O}(10^{-3})$	$\mathcal{O}(10^{-2})$	$\mathcal{O}(10^{-3})$	$\mathcal{O}(10^{-2})$	$\mathcal{O}(10^{-3})$
$r = 5$	$\delta = 0.04$	0.2248	0.2007	0.1526	0.1233	0.1474	0.1167
	$\delta = 0.004$	0.2248	0.2007	0.2219	0.1964	0.1871	0.1841
$r = 10$	$\delta = 0.04$	0.2784	0.1679	0.3435	0.0771	0.2759	0.0718
	$\delta = 0.004$	0.2784	0.1679	0.2789	0.1604	0.2105	0.1487
$r = 15$	$\delta = 0.04$	0.2245	0.0905	2.0121	0.1513	0.4208	0.1483
	$\delta = 0.004$	0.2245	0.0905	0.2214	0.0800	0.2117	0.0721
$r = 20$	$\delta = 0.04$	0.2589	0.0610	NA	0.2813	0.4108	0.1284
	$\delta = 0.004$	0.2589	0.0610	0.2634	0.0562	0.2361	0.0472

Table 6.1: Burgers equation, ED-ROM and AD-ROM errors for four r values, two δ values and two noise magnitude levels. G-ROM errors are also listed for comparison purposes.

r	δ	$\mathcal{K}^{ED}$	$\mathcal{K}^{AD-L}$
$r = 5$	$\delta = 0.04$	2.80	1.01
	$\delta = 0.004$	1.01	1.00
$r = 10$	$\delta = 0.04$	14.10	1.74
	$\delta = 0.004$	1.13	1.01
$r = 15$	$\delta = 0.04$	43.61	12.55
	$\delta = 0.004$	1.50	1.01
$r = 20$	$\delta = 0.04$	101.59	33.83
	$\delta = 0.004$	2.15	1.04

Table 6.2: Burgers equation, ED-ROM and AD-ROM condition numbers for four r values and two δ values.


ED-ROM with noise delta=0.04

AD-L-ROM mu=0.09 and delta= 0.04

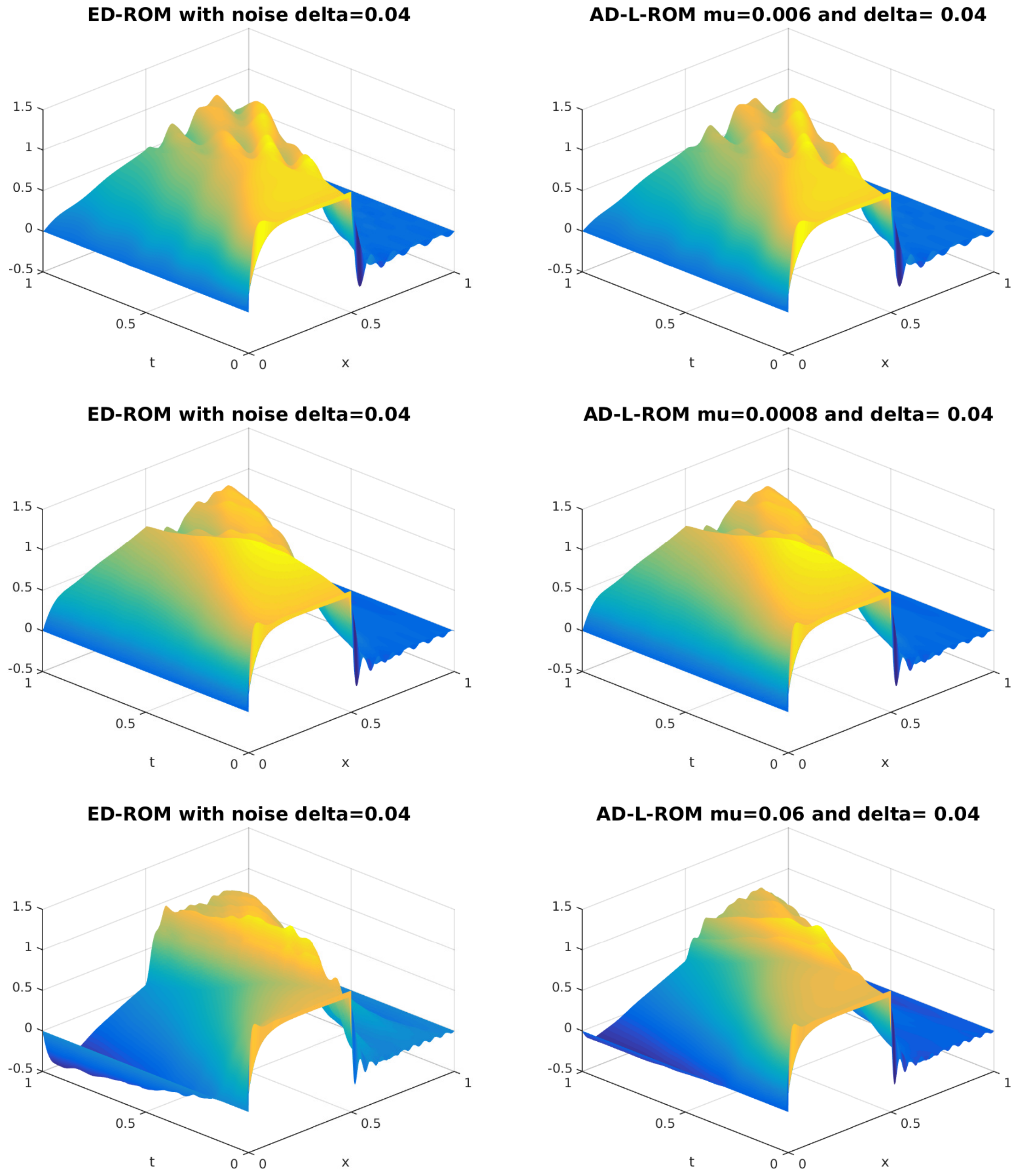
ED-ROM with noise delta=0.04

AD-L-ROM mu=0.7 and delta= 0.04

ED-ROM with noise delta=0.04

AD-L-ROM mu=1 and delta= 0.04

Figure 6.1: Burgers equation, ED-ROM (left column) and AD-ROM (right column) plots for $\delta = 0.04$, noise magnitude 10^{-2}, and $r = 10$ (top row), $r = 15$ (middle row) and $r = 20$ (bottom row).


ED-ROM with noise delta=0.04

AD-L-ROM mu=0.006 and delta= 0.04

ED-ROM with noise delta=0.04

AD-L-ROM mu=0.0008 and delta= 0.04

ED-ROM with noise delta=0.04

AD-L-ROM mu=0.06 and delta= 0.04

Figure 6.2: Burgers equation, ED-ROM (left column) and AD-ROM (right column) plots for $\delta = 0.04$, noise magnitude 10^{-3}, and $r = 10$ (top row), $r = 15$ (middle row) and $r = 20$ (bottom row).

6.3 AD-ROM with Modified Lavrentiev Regularization

In this section, we investigate the AD-ROM with the modified Lavrentiev regularization. Thus, instead of solving eqn. (5.20), we solve the following equation:

$$((1-\mu)G + \mu I)\, u^{AD-L} = \overline{u}\,, \tag{6.2}$$

Therefore, instead of solving eqn. (5.23) for the AD-ROM with Lavrentiev regularization, we solve the following linear system in Algorithm 4 in Section 5.5:

$$\Big((1-\mu)\boldsymbol{M} + \mu\,\boldsymbol{M} + \mu\,\delta^2\,\boldsymbol{S}\Big)\, u^{AD-L} = \Big(\boldsymbol{M} + \delta^2\,\boldsymbol{S}\Big)\,\overline{u}\,. \tag{6.3}$$

Table 6.3 shows that the AD-ROM with Lavrentiev regularization and AD-ROM with modified Lavrentiev regularization yield similar qualitative results.

	G-ROM	AD-ROM-L	AD-ROM-ML
$r=6$	0.2208	0.1113	0.1184
$r=10$	0.1589	0.0633	0.0646
$r=15$	0.0848	0.0532	0.0486
$r=20$	0.0538	0.0529	0.0468

Table 6.3: Burgers equation, errors for the G-ROM (second column), the AD-ROM with Lavrentiev regularization (third column) and the AD-ROM with modified Lavrentiev regularization (fourth column). Four r values and noise of magnitude $\mathcal{O}(10^{-2})$ are used

Chapter 7

Conclusions and Future Work

7.1 Conclusions

Explicit ROM spatial filtering was used to develop a large eddy simulation ROM (LES-ROM) framework. Within this LES-ROM framework, an approximate deconvolution ROM (AD-ROM) was proposed. The AD-ROM was assessed in the numerical simulation of the Burgers equation with a small diffusion coefficient ($\nu = 10^{-3}$). Noise of different magnitude levels was used to model the numerical error. Different values for the ROM spatial filter radius δ were used. The L^2-norm of the error was the criterion used in the numerical assessment of the AD-ROM. The numerical investigation showed that the AD-ROM was significantly more accurate than the exact deconvolution ROM (ED-ROM), in which an exact filter inversion was employed. Furthermore, the AD-ROM was more accurate than the standard G-ROM for low r values, which were appropriate for the test problems employed in this thesis. It should be emphasized that the CPU time of the AD-ROM was orders of magnitude lower than the CPU time of the DNS.

7.2 Future Work

- **Van Cittert and Tikhonov Regularizations** We mentioned these two regularization methods in Section 5.3, but in our numerical tests, we just focused on Lavrentiev regularization. Comparing the three regularization methods appears as a promising research direction.

- **3D Turbulence** As we mentioned at the beginning of this thesis, our ultimate goal is to develop efficient and accurate ROMs for turbulent flows. In the numerical experiments, however, we simply tested the one dimensional Burgers equation. We plan to test the new AD-ROM in realistic three dimensional turbulent flows.

- **Compressive Sensing** Note that in Section 5.3.1, we used the L^2 norm in the minimization problem. We can use the L^1 norm in the minimization problem, which yields the following functional:

$$\Phi_\mu(u) := \|G\,u - \overline{u}\|_1^2 + \mu\,\|u\|_1^2\,, \tag{7.1}$$

This leads to *compressive sensing*, which is an active research topic. Compressed sensing seeks sampling methods that allow the reconstruction of unknown signals given far fewer samples of these signals than would be required to directly acquire the signals. A sparse recovery problem deals with recovering an unknown signal that is sparse in some domain. Numerous relevant results have been obtained in [17, 18, 19, 21, 27]. We will investigate whether the compressive sensing approach can be applied in the deconvolution process of the new AD-ROM.

Bibliography

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

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

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

[4] D. Amsallem, M. J. Zahr, and C. Farhat. Nonlinear model order reduction based on local reduced-order bases. *Int. J. Num. Meth. Eng.*, 92(10):891–916, 2012.

[5] Athanasios C Antoulas. *Approximation of large-scale dynamical systems*, volume 6. Siam, 2005.

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

[7] J. Baiges, R. Codina, and S. Idelsohn. Explicit reduced order models for the stabilized finite element approximation of the incompressible Navier-Stokes equations. *Int. J. Num. Meth. Fluids*, 72(12):1219–1243, 2013.

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

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

[10] M. F. Barone, I. Kalashnikova, D. J. Segalman, and H. K. Thornquist. Stable Galerkin reduced order models for linearized compressible flow. *J. Comput. Phys.*, 228(6):1932–1946, 2009.

[11] Peter Benner, Serkan Gugercin, and Karen Willcox. A survey of model reduction methods for parametric systems. 2013.

[12] M. Bergmann, C. H. Bruneau, and A. Iollo. Enablers for robust POD models. *J. Comput. Phys.*, 228(2):516–538, 2009.

[13] L. C. Berselli, T. Iliescu, and W. J. Layton. *Mathematics of large eddy simulation of turbulent flows*. Scientific Computation. Springer-Verlag, Berlin, 2006.

[14] Luigi Berselli, Traian Iliescu, and William J Layton. *Mathematics of large eddy simulation of turbulent flows*. Springer Science & Business Media, 2005.

[15] M. Bertero and P. Boccacci. *Introduction to inverse problems in imaging*. Institute of Physics Publishing, Bristol, 1998.

[16] DA Bistrian and IM Navon. An improved algorithm for the shallow water equations model reduction: Dynamic mode decomposition vs pod. *International Journal for Numerical Methods in Fluids*, 2015.

[17] B.Kramer. *Model and data reduction for control, identification and compressed sensing.* PhD thesis, Virginia Tech, 2015.

[18] Ido Bright, Guang Lin, and J Nathan Kutz. Compressive sensing based machine learning strategy for characterizing the flow around a cylinder with limited pressure measurements. *Physics of Fluids (1994-present)*, 25(12):127102, 2013.

[19] Steven L Brunton, Jonathan H Tu, Ido Bright, and J Nathan Kutz. Compressive sensing and low-rank libraries for classification of bifurcation regimes in nonlinear dynamical systems. *SIAM Journal on Applied Dynamical Systems*, 13(4):1716–1732, 2014.

[20] A. Caiazzo, T. Iliescu, V. John, and S. Schyschlowa. A numerical investigation of velocity-pressure reduced order models for incompressible flows. *J. Comput. Phys.*, 259:598–616, 2014.

[21] Emmanuel J Candès et al. Compressive sampling. In *Proceedings of the international congress of mathematicians*, volume 3, pages 1433–1452. Madrid, Spain, 2006.

[22] L. Cordier, B. Abou El Majd, and J. Favier. Calibration of POD reduced-order models using Tikhonov regularization. *Int. J. Num. Meth. Fluids*, 63(2):269–296, 2010.

[23] M. Couplet, C. Basdevant, and P. Sagaut. Calibrated reduced-order POD-Galerkin system for fluid flow modelling. *J. Comput. Phys.*, 207(1):192–220, 2005.

[24] M. Couplet, P. Sagaut, and C. Basdevant. Intermodal energy transfers in a proper orthogonal decomposition–Galerkin representation of a turbulent separated flow. *J. Fluid Mech.*, 491:275–284, 2003.

[25] D. N. Daescu and I. M. Navon. A dual-weighted approach to order reduction in 4DVAR data assimilation. *Mon. Weather Rev.*, 136(3):1026–1041, 2008.

[26] G. DE STEFANO and OV VASILYEV. Stochastic coherent adaptive large eddy simulation of forced isotropic turbulence. *Journal of Fluid Mechanics*, 646(1):453–470, 2010.

[27] David L Donoho. Compressed sensing. *Information Theory, IEEE Transactions on*, 52(4):1289–1306, 2006.

[28] B. Galletti, C. H. Bruneau, L. Zannetti, and A. Iollo. Low-order modelling of laminar flow regimes past a confined square cylinder. *J. Fluid Mech.*, 503:161–170, 2004.

[29] Bernardo Galletti, Alessandro Bottaro, Charles-Henri Bruneau, and Angelo Iollo. Accurate model reduction of transient and forced wakes. *European Journal of Mechanics-B/Fluids*, 26(3):354–366, 2007.

[30] M. Germano. Differential filters for the large eddy numerical simulation of turbulent flows. *Phys. Fluids*, 29(6):1755–1757, 1986.

[31] M. Germano. Differential filters of elliptic type. *Phys. Fluids*, 29(6):1757–1758, 1986.

[32] S. Giere, T. Iliescu, V. John, and D. Wells. SUPG reduced order models for convection-dominated convection-diffusion-reaction equations. *Comput. Methods Appl. Mech. Engrg.*, 289:454–474, 2015.

[33] P. C. Hansen. *Discrete inverse problems: insight and algorithms*, volume 7. Society for Industrial and Applied Mathematics, 2010.

[34] Jan S Hesthaven, Gianluigi Rozza, and Benjamin Stamm. Certified reduced basis methods for parametrized partial differential equations. *SpringerBriefs in Mathematics*, 2015.

[35] P. Holmes, J. L. Lumley, and G. Berkooz. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry.* Cambridge, 1996.

[36] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.

[37] T. Iliescu and Z. Wang. Variational multiscale proper orthogonal decomposition: Convection-dominated convection-diffusion-reaction equations. *Math. Comput.*, 82(283):1357–1378, 2013.

[38] T. Iliescu and Z. Wang. Are the snapshot difference quotients needed in the proper orthogonal decomposition? *SIAM J. Sci. Comput.*, 36(3):A1221âĂŞA1250, 2014.

[39] T. Iliescu and Z. Wang. Variational multiscale proper orthogonal decomposition: Navier-Stokes equations. *Num. Meth. P.D.E.s*, 30(2):641–663, 2014.

[40] Kari Karhunen. *Zur spektraltheorie stochastischer prozesse.* Suomalainen tiedeakatemia, 1946.

[41] Virginia C Klema and Alan J Laub. The singular value decomposition: Its computation and some applications. *Automatic Control, IEEE Transactions on*, 25(2):164–176, 1980.

[42] Andrey Nikolaevich Kolmogorov. Dissipation of energy in the locally isotropic turbulence. *Proceedings: Mathematical and Physical Sciences*, pages 15–17, 1991.

[43] K. Kunisch and S. Volkwein. Galerkin proper orthogonal decomposition methods for parabolic problems. *Numer. Math.*, 90(1):117–148, 2001.

[44] T. Lassila, A. Manzoni, A. Quarteroni, and G. Rozza. Model order reduction in fluid dynamics: challenges and perspectives. In *Reduced Order Methods for modeling and computational reduction*, pages 235–273. Springer, 2014.

[45] W. Layton and I. Stanculescu. Chebychev optimized approximate deconvolution models of turbulence. *Appl. Math. Comput.*, 208(1):106–118, 2009.

[46] W. J. Layton. *Introduction to the numerical analysis of incompressible viscous flows*, volume 6. Society for Industrial and Applied Mathematics (SIAM), 2008.

[47] W. J. Layton and L. G. Rebholz. *Approximate deconvolution models of turbulence: analysis, phenomenology and numerical analysis*, volume 2042. Springer, 2012.

[48] Michel Moise Loeve. *Probability Theory: 2d Ed.* D. Van Nostrand, 1960.

[49] B. R. Noack, M. Morzynski, and G. Tadmor. *Reduced-Order Modelling for Flow Control*, volume 528. Springer Verlag, 2011.

[50] B. R. Noack, P. Papas, and P. A. Monkewitz. The need for a pressure-term representation in empirical Galerkin models of incompressible shear flows. *J. Fluid Mech.*, 523:339–365, 2005.

[51] B. R. Noack, M. Schlegel, B. Ahlborn, G. Mutschke, M. Morzynski, P. Comte, and G. Tadmor. A finite-time thermodynamics of unsteady fluid flows. *J. Non-Equil. Thermody.*, 33(2):103–148, 2008.

[52] J. Östh. *Unsteady Numerical Simulations and Reduced-Order Modelling of Flows around Vehicles.* PhD thesis, Chalmers University of Technology, 2014.

[53] J. Östh, B. R Noack, S. Krajnović, D. Barros, and J. Borée. On the need for a nonlinear subscale turbulence term in POD models as exemplified for a high-Reynolds-number flow over an Ahmed body. *J. Fluid Mech.*, 747:518–544, 2014.

[54] P. Pacciarini and G. Rozza. Stabilized reduced basis method for parametrized advection–diffusion PDEs. *Comput. Meth. Appl. Mech. Eng.*, 274:1–18, 2014.

[55] B. Protas, B. R. Noack, and J. Östh. Optimal nonlinear eddy viscosity in Galerkin models of turbulent flows. *J. Fluid Mech.*, 766:337–367, 2015.

[56] A. Quarteroni, G. Rozza, and A. Manzoni. Certified reduced basis approximation for parametrized partial differential equations and applications. *Journal of Mathematics in Industry*, 1(1):1–49, 2011.

[57] Alfio Quarteroni, Andrea Manzoni, and Federico Negri. *Reduced Basis Methods for Partial Differential Equations: An Introduction*, volume 92. Springer, 2015.

[58] T. Chacón Rebollo and R. Lewandowski. *Mathematical and numerical foundations of turbulence models and applications.* Springer, 2014.

[59] F. Sabetghadam and A. Jafarpour. α regularization of the POD-Galerkin dynamical systems of the Kuramoto–Sivashinsky equation. *Appl. Math. Comput.*, 218(10):6012–6026, 2012.

[60] P. Sagaut. *Large eddy simulation for incompressible flows.* Scientific Computation. Springer-Verlag, Berlin, 2001. An introduction, With an introduction by Marcel Lesieur, Translated from the 1998 French original by the author.

[61] Pierre Sagaut. *Large eddy simulation for incompressible flows: an introduction.* Springer Science & Business Media, 2006.

[62] Omer San and Traian Iliescu. A stabilized proper orthogonal decomposition reduced-order model for large scale quasigeostrophic ocean circulation. *Advances in Computational Mathematics*, pages 1–31, 2014.

[63] A. Scotti and S. Mitran. An approximated method for the solution of elliptic problems in thin domains: Application to nonlinear internal waves. *Ocean Modelling*, 25:144–153, 2008.

[64] S. Sirisup and G. E. Karniadakis. A spectral viscosity method for correcting the long-term behavior of POD models. *J. Comput. Phys.*, 194(1):92–116, 2004.

[65] L. Sirovich. Turbulence and the dynamics of coherent structures. Parts I–III. *Quart. Appl. Math.*, 45(3):561–590, 1987.

[66] J. S. Smagorinsky. General circulation experiments with the primitive equations. *Mon. Weather Rev.*, 91:99–164, 1963.

[67] I. Stanculescu and C. C. Manica. Numerical analysis of Leray–Tikhonov deconvolution models of fluid motion. *Comput. Math. Appl.*, 60(5):1440–1456, 2010.

[68] Răzvan Ştefănescu, Adrian Sandu, and Ionel M Navon. Comparison of pod reduced order strategies for the nonlinear 2d shallow water equations. *International Journal for Numerical Methods in Fluids*, 76(8):497–521, 2014.

[69] S. Stolz and N.A Adams. An approximate deconvolution procedure for large-eddy simulation. *Phys. Fluids*, 11(7):1699–1701, 1999.

[70] S. Stolz and N.A Adams. Large-eddy simulation of high-reynolds-number supersonic boundary layers using the approximate deconvolution model and a rescaling and recycling technique. *Phys. Fluids*, 15(8):2398–2412, 2003.

[71] S. Stolz, N.A. Adams, and L. Kleiser. An approximate deconvolution model for large-eddy simulation with application to incompressible wall-bounded flows. *Phys. Fluids*, 13(4):997–1015, 2001.

[72] S. Stolz, N.A. Adams, and L. Kleiser. The approximate deconvolution model for large-eddy simulations of compressible flows and its application to shock-turbulent-boundary-layer interaction. *Phys. Fluids*, 13(10):2985–3001, 2001.

[73] S. Ullmann and J. Lang. A POD-Galerkin reduced model with updated coefficients for Smagorinsky LES. In J. C. F. Pereira and A. Sequeira, editors, *V European Conference on Computational Fluid Dynamics, ECCOMAS CFD 2010*, Lisbon, Portugal, June 2010.

[74] C. R. Vogel. *Computational methods for inverse problems*, volume 23. Society for Industrial and Applied Mathematics (SIAM), 2002.

[75] S. Volkwein. Model reduction using proper orthogonal decomposition. *Lecture Notes, Faculty of Mathematics and Statistics, University of Konstanz*, 2011.

[76] Y. Wang, I. M. Navon, X. Wang, and Y. Chengd. 2D Burgers equations with large Reynolds number using POD/DEIM and calibration. 2015. preprint.

[77] Z. Wang, I. Akhtar, J. Borggaard, and T. Iliescu. Two-level discretizations of nonlinear closure models for proper orthogonal decomposition. *J. Comput. Phys.*, 230:126–146, 2011.

[78] Z. Wang, I. Akhtar, J. Borggaard, and T. Iliescu. Proper orthogonal decomposition closure models for turbulent flows: A numerical comparison. *Comput. Meth. Appl. Mech. Eng.*, 237-240:10–26, 2012.

[79] J. Weller, E. Lombardi, and A. Iollo. Robust model identification of actuated vortex wakes. *Phys. D*, 238(4):416–427, 2009.

[80] D. Wells, Z. Wang, X. Xie, and T. Iliescu. Regularized reduced order models. *arXiv preprint arXiv:1506.07555*, 2015.

[81] David Wells. *Stabilization of pod-rom.* PhD thesis, Virginia Tech, 2015.