

Reduced-Order Modeling of Complex Engineering and Geophysical Flows: Analysis and Computations

Zhu Wang

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

Doctor of Philosophy

in

Applied Mathematics

Traian Iliescu, Chairman

Jeff Borggaard

John A. Burns

Tao Lin

Lizette Zietsman

April 17, 2012

Blacksburg, Virginia

Keywords: Proper orthogonal decomposition, reduced-order modeling, turbulence, large eddy simulation, eddy viscosity, variational multiscale, dynamic subgrid-scale model, approximate deconvolution, two-level algorithm, finite elements, numerical analysis

Copyright 2012, Zhu Wang

Reduced-Order Modeling of Complex Engineering and Geophysical Flows: Analysis and Computations

Zhu Wang

(ABSTRACT)

Reduced-order models are frequently used in the simulation of complex flows to overcome the high computational cost of direct numerical simulations, especially for three-dimensional nonlinear problems. Proper orthogonal decomposition, as one of the most commonly used tools to generate reduced-order models, has been utilized in many engineering and scientific applications. Its original promise of computationally efficient, yet accurate approximation of coherent structures in high Reynolds number turbulent flows, however, still remains to be fulfilled. To balance the low computational cost required by reduced-order modeling and the complexity of the targeted flows, appropriate closure modeling strategies need to be employed.

In this dissertation, we put forth two new closure models for the proper orthogonal decomposition reduced-order modeling of structurally dominated turbulent flows: the dynamic subgrid-scale model and the variational multiscale model. These models, which are considered state-of-the-art in large eddy simulation, are carefully derived and numerically investigated.

Since modern closure models for turbulent flows generally have non-polynomial nonlinearities, their efficient numerical discretization within a proper orthogonal decomposition framework is challenging. This dissertation proposes a two-level method for an efficient and accurate numerical discretization of general nonlinear proper orthogonal decomposition closure models. This method computes the nonlinear terms of the reduced-order model on a coarse mesh. Compared with a brute force computational approach in which the nonlinear terms are evaluated on the fine mesh at each time step, the two-level method attains the

same level of accuracy while dramatically reducing the computational cost. We numerically illustrate these improvements in the two-level method by using it in three settings: the one-dimensional Burgers equation with a small diffusion parameter $\nu = 10^{-3}$, a two-dimensional flow past a cylinder at Reynolds number $\text{Re} = 200$, and a three-dimensional flow past a cylinder at Reynolds number $\text{Re} = 1000$.

With the help of the two-level algorithm, the new nonlinear proper orthogonal decomposition closure models (i.e., the dynamic subgrid-scale model and the variational multiscale model), together with the mixing length and the Smagorinsky closure models, are tested in the numerical simulation of a three-dimensional turbulent flow past a cylinder at $\text{Re} = 1000$. Five criteria are used to judge the performance of the proper orthogonal decomposition reduced-order models: the kinetic energy spectrum, the mean velocity, the Reynolds stresses, the root mean square values of the velocity fluctuations, and the time evolution of the proper orthogonal decomposition basis coefficients. All the numerical results are benchmarked against a direct numerical simulation. Based on these numerical results, we conclude that the dynamic subgrid-scale and the variational multiscale models are the most accurate.

We present a rigorous numerical analysis for the discretization of the new models. As a first step, we derive an error estimate for the time discretization of the Smagorinsky proper orthogonal decomposition reduced-order model for the Burgers equation with a small diffusion parameter. The theoretical analysis is numerically verified by two tests on problems displaying shock-like phenomena. We then present a thorough numerical analysis for the finite element discretization of the variational multiscale proper orthogonal decomposition reduced-order model for convection-dominated convection-diffusion-reaction equations. Numerical tests show the increased numerical accuracy over the standard reduced-order model and illustrate the theoretical convergence rates.

We also discuss the use of the new reduced-order models in realistic applications such as airflow simulation in energy efficient building design and control problems as well as numerical simulation of large-scale ocean motions in climate modeling. Several research directions that we plan to pursue in the future are outlined.

Dedication

To Yixuan and Ashlyn.

Acknowledgments

First and foremost, I would like to express my most sincere gratitude to my advisor, Dr. Traian Iliescu, who led me into these exciting research areas and encouraged me to explore my research interests. I have always benefited from his knowledge, enthusiasm, patience, and constant support. I am so glad to have him as my advisor and enjoy working with him.

I am also deeply grateful to Dr. Jeff Borggaard for many valuable discussions and suggestions in my research. I would like to thank the rest of my PhD dissertation committee: Dr. John Burns, Dr. Tao Lin, and Dr. Lizette Zietsman, for their insightful comments on the dissertation.

I wish to express my heartfelt thanks to Dr. Mihai Anitescu and Dr. Oleg Roderick for offering me the wonderful summer internship opportunities at Argonne National Laboratory in two consecutive years and for introducing me to diverse projects in nuclear engineering. I also owe my warm thanks to Dr. Imran Akhtar, Dr. Anne Staples, and Dr. Omer San, for their help and collaborations in various engineering and geophysical applications.

Last but not the least, I would like to give my special thanks to my wife, Yixuan Lin. I will always appreciate her love and understanding. Without her support, this dissertation could not have been possible.

Contents

Abstract	ii
Dedication	iv
Acknowledgments	v
Contents	vi
List of Figures	ix
List of Tables	xiii
1 Introduction	1
1.1 Motivation	1
1.2 Proper Orthogonal Decomposition (POD)	7
1.3 POD Reduced-Order Model	8
1.3.1 Galerkin Projection	8
1.3.2 Mathematical Foundations	10
1.4 POD Closure Modeling	12
1.5 Efficient Numerical Discretizations	18
1.6 “Perfect” POD-ROM for Complex Flows	19
1.7 Outline of dissertation	21
2 POD Closure Models for Turbulent Flows	24
2.1 Introduction	24

2.2	Filter and Lengthscale	26
2.3	POD Closure Models	28
2.4	Summary	36
3	Two-Level Algorithms for Nonlinear Closure Models	37
3.1	Introduction	37
3.2	Two-Level Algorithms	39
3.3	Numerical Results	43
3.3.1	1D Burgers Equation with $\nu = 10^{-3}$	44
3.3.2	2D Flow Past a Circular Cylinder at $Re = 200$	47
3.3.3	3D Flow Past a Circular Cylinder at $Re = 1000$	56
3.4	Summary	62
4	Numerical Analysis	64
4.1	Introduction	64
4.2	Artificial Viscosity POD-ROM	65
4.2.1	Error Estimates	67
4.2.2	Numerical Results	72
4.3	Variational Multiscale POD-ROM	80
4.3.1	Error Estimates	82
4.3.2	Numerical Results	94
4.4	Summary	100
5	Numerical Experiments of POD Closure Models in Turbulence	103
5.1	Introduction	103
5.2	Numerical Methods and Parameter Choices	104
5.3	Numerical Results	109
5.4	Summary	115
6	Future Work	133
6.1	Introduction	133
6.2	Energy Efficient Building Airflow Simulations	134

6.3	Climate Modeling	139
6.4	Future Research Directions	144
	Bibliography	147

List of Figures

1.1	Schematic for the “perfect” POD-ROM for complex flows.	19
3.1	1D Burgers equation with $\nu = 10^{-3}$. POD basis functions: (left) φ_{10} ; (right) φ_{70}	41
3.2	1D Burgers equation with $\nu = 10^{-3}$. Coarsening and interpolation of φ_{10} for two rates of coarsening: 32 (left) and 256 (right).	46
3.3	(Figure continued on next page.) 1D Burgers equation with $\nu = 10^{-3}$. Temporal evolution of POD coefficients $a_1(\cdot)$ and $a_6(\cdot)$ over $[0, 1]$	48
3.4	Figure 3.3 continued. See figure caption on previous page.	49
3.5	(Figure continued on next page.) 2D flow past a cylinder at $\text{Re} = 200$. Temporal evolution of POD coefficients $a_1(\cdot)$ and $a_3(\cdot)$ over 100 vortex shedding cycles.	50
3.6	Figure 3.5 continued. See figure caption on previous page.	51
3.7	(Figure continued on next page.) 2D flow past a cylinder at $\text{Re} = 200$. 2D projections of the limit cycle on the phase planes (a_1, a_2) and (a_1, a_3) over 100 vortex shedding cycles. The red square denotes the initial position of $(a_1(0), a_i(0))$, $i = 2, 3$	52
3.8	Figure 3.7 continued. See figure caption on previous page.	53
3.9	2D flow past a cylinder at $\text{Re} = 200$: Mesh (left column), streamwise component of φ_1 (middle column), and streamwise component of φ_3 (right column). (For clarity, the POD modes are plotted only in a neighborhood of the cylinder.) Original fine mesh (top row), mesh coarsened by a factor of 2 (middle row), and mesh coarsened by a factor of 4 (bottom row).	54

3.10	3D flow past a cylinder at $Re = 1000$. First streamwise POD mode (top left), first normal POD mode (top right), third streamwise POD mode (bottom left), and third normal POD mode (bottom right).	57
3.11	(Figure continued on next page.) 3D flow past a cylinder at $Re = 1000$. Temporal evolution of POD coefficients $a_1(\cdot)$ and $a_4(\cdot)$ over the time interval $[0, 75]$	58
3.12	Figure 3.11 continued. See figure caption on previous page.	59
4.1	DNS solution in Experiment 1.	74
4.2	POD-G-ROM (4.7) solution in Experiment 1.	74
4.3	AV-POD-ROM (4.9) solution in Experiment 1.	75
4.4	Snapshots of DNS, POD-G-ROM and AV-POD-ROM at the final time $T = 1$ in Experiment 1.	75
4.5	DNS solution in Experiment 2.	77
4.6	POD-G-ROM (4.7) solution in Experiment 2.	78
4.7	AV-POD-ROM (4.9) solution in Experiment 2.	78
4.8	Snapshots of DNS, POD-G-ROM and AV-POD-ROM at the final time $T = 1.5$ in Experiment 2.	79
4.9	Numerical solution at $t = 1$: DNS (top), POD-G-ROM (4.42) (middle), and P_R -VMS-POD-ROM (4.44) (bottom).	97
4.10	Linear regression of P_R -VMS-POD-ROM's average error with respect to e_3	98
5.1	(Continued on next page.) Snapshots of horizontal velocity at $t = 142.5$ s for: (a) DNS; (b) the POD-G-ROM (1.12); (c) the ML-POD-ROM (2.14)-(2.15); (d) the S-POD-ROM (2.17)-(2.18); (e) the new VMS-POD-ROM (2.25)-(2.34); and (f) the new DS-POD-ROM (2.52)-(2.53). Five iso-surfaces are plotted.	107
5.2	Kinetic energy spectrum of the DNS (blue) and the POD-ROMs (red) for $r = 6$: (a) the POD-G-ROM (1.12); (b) the ML-POD-ROM (2.14)-(2.15); (c) the S-POD-ROM (2.17)-(2.18); (d) the new VMS-POD-ROM (2.25)-(2.34); and (e) the new DS-POD-ROM (2.52)-(2.53).	117

5.3	Mean velocity components for DNS and POD-ROMs when $r = 6$: (a) $\langle u \rangle$, (b) $\langle v \rangle$, and (c) $\langle w \rangle$, where $\langle \cdot \rangle = \langle \cdot \rangle_{tyz}$	118
5.4	Mean velocity components for DNS and POD-ROMs when $r = 6$: (a) $\langle u \rangle$, (b) $\langle v \rangle$, and (c) $\langle w \rangle$, where $\langle \cdot \rangle = \langle \cdot \rangle_{txz}$	119
5.5	Reynolds stresses for DNS and POD-ROMs when $r = 6$: (a) $\langle u - \langle u \rangle, v - \langle v \rangle \rangle$, (b) $\langle u - \langle u \rangle, w - \langle w \rangle \rangle$, and (c) $\langle v - \langle v \rangle, w - \langle w \rangle \rangle$, where $\langle \cdot \rangle = \langle \cdot \rangle_{tyz}$	120
5.6	Reynolds stresses for DNS and POD-ROMs when $r = 6$: (a) $\langle u - \langle u \rangle, v - \langle v \rangle \rangle$, (b) $\langle u - \langle u \rangle, w - \langle w \rangle \rangle$, and (c) $\langle v - \langle v \rangle, w - \langle w \rangle \rangle$, where $\langle \cdot \rangle = \langle \cdot \rangle_{txz}$	121
5.7	Rms values of the velocity fluctuations for DNS and POD-ROMs when $r = 6$: (a) $\langle u \rangle_{rms} = \langle u - \langle u \rangle, u - \langle u \rangle \rangle$, (b) $\langle v \rangle_{rms} = \langle v - \langle v \rangle, v - \langle v \rangle \rangle$, and (c) $\langle w \rangle_{rms} = \langle w - \langle w \rangle, w - \langle w \rangle \rangle$, where $\langle \cdot \rangle = \langle \cdot \rangle_{tyz}$	122
5.8	Rms values of the velocity fluctuations for DNS and POD-ROMs when $r = 6$: (a) $\langle u \rangle_{rms} = \langle u - \langle u \rangle, u - \langle u \rangle \rangle$, (b) $\langle v \rangle_{rms} = \langle v - \langle v \rangle, v - \langle v \rangle \rangle$, and (c) $\langle w \rangle_{rms} = \langle w - \langle w \rangle, w - \langle w \rangle \rangle$, where $\langle \cdot \rangle = \langle \cdot \rangle_{txz}$	123
5.9	Time evolutions of the POD basis coefficient a_1 of the DNS (blue) and the POD-ROMs (red) for $r = 6$: (a) the POD-G-ROM (1.12); (b) the ML-POD-ROM (2.14)-(2.15); (c) the S-POD-ROM (2.17)-(2.18); (d) the new VMS-POD-ROM (2.25)-(2.34); and (e) the new DS-POD-ROM (2.52)-(2.53).	124
5.10	Time evolutions of the POD basis coefficient a_4 of the DNS (blue) and the POD-ROMs (red) for $r = 6$: (a) the POD-G-ROM (1.12); (b) the ML-POD-ROM (2.14)-(2.15); (c) the S-POD-ROM (2.17)-(2.18); (d) the new VMS-POD-ROM (2.25)-(2.34); and (e) the new DS-POD-ROM (2.52)-(2.53).	125
5.11	Time evolutions of the POD basis coefficient a_4 of the DNS (green), the new VMS-POD-ROM (2.25)-(2.34) (blue), and the new DS-POD-ROM (2.52)-(2.53) (red) when $r = 6$	126
5.12	Kinetic energy spectrum of the DNS (blue) and the POD-ROMs (red) for $r = 4$	127
5.13	Mean velocity components for DNS and POD-ROMs when $r = 4$	128
5.14	Reynolds stresses for DNS and POD-ROMs when $r = 4$	129
5.15	Rms values of the velocity fluctuations for DNS and POD-ROMs when $r = 4$	130

5.16	Time evolutions of the POD basis coefficient a_1 of the DNS (blue) and the POD-ROMs (red) for $r = 4$	131
5.17	Time evolutions of the POD basis coefficient a_4 of the DNS (blue) and the POD-ROMs (red) for $r = 4$	132
6.1	Reduced-order modeling of airflow in the DOE energy-efficient buildings system design hub.	134
6.2	Mean temperature of airflow in a simple geometry.	134
6.3	The first nine POD modes $\{\psi_1(x), \dots, \psi_9(x)\}$ in the temperature field. . . .	137
6.4	Three temperature iso-surfaces (at $T=293.2, 293.65$ and 294.1) for the time instance $t = 10$. Top: DNS; Middle: POD-G-ROM; Bottom: ML-POD-ROM.	138
6.5	Ocean circulation traced by the GOCE gravity mission (source from http://earth.esa.int/GOCE/). 139	
6.6	Double-gyre oceanic circulation modeled by the QGE. Time-averaged streamfunction for $Re = 450$ and $Ro = 0.0036$: (a) DNS (512×256); (b) under-resolved QGE (32×16); (c) AD model (32×16). The new AD model yields accurate results on a much coarser mesh.	141
6.7	Mean streamfunction from POD-G-ROMs with $r=10, 5,$ and 3 (from left to right).	143
6.8	Energy evolution of POD-G-ROMs with $r = 10, 5,$ and 3 (from left to right).	144

List of Tables

3.1	1D Burgers equation with $\nu = 10^{-3}$. Speed-up factors S_f and relative errors for different coarsening factors R_c	47
3.2	2D flow past a cylinder at $\text{Re} = 200$. Speed-up factors S_f and relative errors for different coarsening factors R_c	55
3.3	3D flow past a cylinder at $\text{Re} = 1000$. Speed-up factors S_f and relative errors for different coarsening factors R_c	61
4.1	Average errors for POD-G-ROM (4.7) and AV-POD-ROM (4.9) in Experiment 1.	76
4.2	Average errors for POD-G-ROM (4.7) and AV-POD-ROM (4.9) in Experiment 2.	79
4.3	Average errors for the POD-G-ROM (4.42) with different values of r	95
4.4	P_R -VMS-POD-ROM's average error $e = \frac{1}{M} \sum_{k=1}^M \ u^k - u_r^k\ $ and its e_3 component for different values of R	96
4.5	P_R -VMS-POD-ROM's average error $e = \frac{1}{M} \sum_{k=1}^M \ u^k - u_r^k\ $ for different values of r and R , and $\alpha = 0.01 \alpha^*$, α^* , and $100 \alpha^*$	100
4.6	Average errors of DNS, POD-G-ROM and P_R -VMS-POD-ROM for different values of the diffusion coefficient ε	101
5.1	Speed-up factors of POD-ROMs.	113

Chapter 1

Introduction

1.1 Motivation

Due to the complexity of fluid flows in realistic engineering and geophysical problems, millions or even billions of degrees of freedom are often required in a direct numerical simulation (DNS). However, because of the requisite of repeated simulations in applications such as control, optimization, data assimilation, and uncertainty quantification, using the original system becomes prohibitive. Therefore, the model reduction methodology is widely used by engineers and researchers (e.g., [15, 27, 108, 178, 202]). The goal is to obtain a low-dimensional model, which is efficient in numerical simulations but contains the same characteristics as the original system, thus yielding accurate results.

Let $\mathbf{y}(x, t)$, with $x \in \Omega$ and $t \in (t_0, t_0 + T)$, be the state variable in the original system and \mathcal{H} a Hilbert space. The complex flow, usually *nonlinear* and *time-dependent*, is governed by a system of partial differential equations (PDEs) or a large system of ordinary differential equations (ODEs). The PDE system, which comprises an infinite number of degrees of freedom, reads, e.g., to find $\mathbf{y}(\cdot, t) \in \mathcal{H}$ satisfying

$$\begin{cases} \dot{\mathbf{y}}(x, t) = \mathbf{f}(t, \mathbf{y}(x, t)) \\ \mathbf{y}(x, t_0) = \mathbf{y}_0(x). \end{cases} \quad (1.1)$$

The ODE system is usually an approximation of (1.1) by well established numerical methods such as finite difference or finite element methods with large number of degrees of freedom.

It reads, e.g., to find $\mathbf{y}(\cdot, t) \in \mathbb{R}^N$ satisfying

$$\begin{cases} \dot{\mathbf{y}}(x, t) = \mathbf{f}(t, \mathbf{y}(x, t)) \\ \mathbf{y}(x, t_0) = \mathbf{y}_0(x). \end{cases} \quad (1.2)$$

The basic premise of model reduction is to approximate a full-order model, (1.1) or (1.2), by only a handful of degrees of freedom. The resulting low-dimensional model becomes a system of ODEs with a dramatically reduced dimension r ($r \ll N$).

In this dissertation, we focus on the *proper orthogonal decomposition (POD)*-based reduced-order modeling approach, which is one of the most significant *projection-based* model reduction methods for nonlinear dynamical systems. The POD method provides an optimal basis (or modes) to represent the dynamical system. This method is also known as the Karhunen-Loève expansion [130, 151] and principal component analysis (PCA) [111] in statistics, singular value decomposition (SVD) [33, 134] in matrix theory and signal analysis, and empirical orthogonal functions (EOF) [152] in meteorology and geophysical fluid dynamics. For a systematic proof of the equivalence connection among the first three methods, the reader is referred to [147].

The POD method has had a long and rich history in the reduced-order modeling of complex systems. Its use as a simulation tool is a tremendously active research area. With the risk of being subjective, we list some of these applications below. Arguably the most dynamic research area in which the POD method has been used as a reduced-order modeling technique is fluid flow control [3, 4, 7, 22, 23, 30, 39, 47, 57, 91, 107, 119, 120, 139, 146, 188]. Other applications include climate modeling and data assimilation of atmospheric and oceanic flows [48, 49, 56, 61, 62, 63, 76, 77, 156, 157, 231], low-dimensional dynamics modeling [14, 31, 66, 108, 160, 161, 172], stochastic PDEs [46, 70, 220], surrogate models for optimization [17, 44, 51], image processing [108], pattern recognition [209], gappy data reconstruction [100, 168, 219, 230], the complex Ginzburg-Landau equations [197, 210], groundwater models [50], control texture in the growth of polycrystals [1], pectoral fin kinematics [40], nonlinear structural systems [132, 141, 201], structural vibration [80, 81, 90, 133], vehicle interior acoustics [222], aircraft configuration [148, 149], structural health monitoring [142], ecosystems [143], and cell population models [232]. Several of these works cite the need to explicitly account for the influence of truncation or slow separation in the POD spectrum,

e.g., [30, 201, 205, 232]. The importance of reduced-order modeling in all these applications cannot be overemphasized. Indeed, the POD-based *reduced-order models (ROMs)* (which we will denote in the sequel POD-ROMs) dramatically reduce the computational cost of a brute force numerical simulation, which for some of the above applications (e.g., climate modeling) is clearly unfeasible.

The POD method was introduced in the field of *turbulence* by Lumley [153] to identify the coherent structures in the flow and examine their stability [108]. Later, Sirovich [206, 207, 208] introduced the *method of snapshots* to study the dynamics of turbulent flows based on a set of instantaneous flow field solutions, so-called *snapshots*, obtained from either experiments or numerical simulations. This method allows a fast generation of the POD basis from the snapshots. Such POD basis functions are optimal in the sense that they capture the dominant energy contents, thus become a suitable candidate for the ROM. The resulting POD-ROM permits an analytical insight into the physical phenomenon and enables applications of dynamical systems theory and control methods.

The most common approach for generating POD-ROMs is first to simulate the *full-order* system (1.1) or (1.2) and find a set of “representative” snapshots (state variable vectors), then process the data set by the POD method to find an optimal basis $\{\boldsymbol{\varphi}_1(x), \boldsymbol{\varphi}_2(x), \dots, \boldsymbol{\varphi}_r(x)\}$, and finally employ the Galerkin projection to obtain a *low-order* dynamical system for the basis coefficients $\{\mathbf{a}_1(t), \mathbf{a}_2(t), \dots, \mathbf{a}_r(t)\}$. The resulting low-order model is named *POD Galerkin reduced-order model (POD-G-ROM)*. The basic procedure is outlined in Algorithm 1.

The POD method can be viewed as a method of information compression. As a consequence, the ability of the POD modes to approximate any state of a complex system is totally dependent on the information originally contained in the snapshot set used to generate the POD basis functions. That is, a POD basis cannot contain more information than that contained in the snapshot set. Thus, the choice of snapshots can have a strong influence on the general applicability of the ROM in applications such as parametrized problems in flow control and optimization, bifurcation analysis and uncertainty quantification. For example, in [65], the authors noted that although POD-ROM performs well at a specific Reynolds number at which snapshots are taken, “the accuracy of the model predictions

Algorithm 1: POD-G-ROM Generation

Given $\mathbf{y}(\cdot, t)$ resulting from the complex system (1.1) or (1.2) for $t \in (t_0, t_0 + T)$

(1) Compute a POD basis

$$\{\varphi_1(x), \varphi_2(x), \dots, \varphi_r(x)\}$$

such that

$$\mathbf{X}^r = \text{span} \{\varphi_1(\cdot), \varphi_2(\cdot), \dots, \varphi_r(\cdot)\} \quad (1.3)$$

is a good approximation for the data space

$$\{\mathbf{y}(\cdot, t)\}_{t \in (t_0, t_0 + T)}.$$

Define the reduced-order approximation

$$\mathbf{y}_r(\cdot, t) = \sum_{j=1}^r \varphi_j(\cdot) a_j(t) \in \mathbf{X}^r, \quad (1.4)$$

where $\{\mathbf{a}_j(t)\}_{j=1}^r$ are the sought time-varying POD basis coefficient functions.

(2) Substitute the POD approximation (1.4) into the full-order system (1.1) or (1.2), and apply the Galerkin procedure.

$$\begin{aligned} \left\langle \sum_{j=1}^r \varphi_j(\cdot) \dot{a}_j(t), \varphi_i(\cdot) \right\rangle &= \left\langle \mathbf{f} \left(t, \sum_{j=1}^r \varphi_j(\cdot) a_j(t) \right), \varphi_i(\cdot) \right\rangle, \\ \left\langle \sum_{j=1}^r \varphi_j(\cdot) a_j(0), \varphi_i(\cdot) \right\rangle &= \langle \mathbf{y}_0, \varphi_i(\cdot) \rangle \text{ for } i = 1, \dots, r, \end{aligned}$$

which gives the POD-G-ROM, i.e., a dynamical system for $\{\mathbf{a}_i(t)\}_{i=1}^r$:

$$\dot{a}_i(t) = \left\langle \mathbf{f} \left(t, \sum_{j=1}^r \varphi_j(\cdot) a_j(t) \right), \varphi_i(\cdot) \right\rangle \quad (1.5)$$

with

$$a_i(0) = \langle \mathbf{y}_0, \varphi_i(\cdot) \rangle \text{ for } i = 1, \dots, r.$$

rapidly deteriorates as we move away from the decomposition value.” Several ideas have been proposed to *improve the selection of snapshots*. For example, with bifurcation information at hand, snapshots were taken at only a few parameter samples in [161] to derive a POD-G-ROM that is able to capture the 3D bifurcation of the wake flow. Without any *a priori* knowledge such as the bifurcation information, in [128], snapshots were taken at samples uniformly distributed in the parameter space and the sequential proper orthogonal decomposition (SPOD) was introduced. A more intelligent parameter sampling method, the centroidal Voronoi tessellation (CVT) method was introduced in [45]. An adaptive method in updating snapshots during the optimization process was developed in [187, 189]. These approaches were discussed in [30]. A different adaptive method is the trust-region proper orthogonal decomposition (TRPOD). This approach, introduced in [16, 75], benefits from the trust-region philosophy [177]. Rigorous global convergence results were proved for the TRPOD of the *Navier-Stokes equations (NSE)*. It was further investigated in [56] within a dual weighting data assimilation system for the shallow-water equations. For parametrized problems, the reduced basis method (RBM), which assumes a smooth low-dimensional manifold induced by the parameter dependence, is one of the most powerful model reduction techniques. Its rigorous *a posteriori* error estimators have been proposed in [184, 198]. The sampling procedure for snapshot selections was discussed in [101, 221]. Furthermore, a greedy algorithm was used for an optimal selection of parameter values [34, 42, 97, 162, 221]. The offline-online computational decoupling, which is based on the empirical interpolation method (EIM) [24, 96], was discussed in [171, 198].

The performance of POD-ROMs relies on how much information is retained in the reduced system, which is directly influenced by the number of POD modes kept in the ROM. Using more POD modes in the ROM keeps more information, however, greatly increases the computational cost. Therefore, an alternative way for improving POD-ROMs is to *improve the basis functions*. Next, we briefly mention some of the approaches used to develop improved POD basis functions. In [172], shift modes were incorporated in the POD basis. This method significantly improved the resolution of the transient dynamics from the onset of vortex shedding to the periodic von Kármán vortex street. In the same paper, it was shown that the inclusion of stability eigenmodes further enhances the accuracy of fluctuation dynamics.

In [36], the principal interval decomposition (PID) was used to extract dominant POD basis functions over time subintervals. The POD mode interpolation methods were also used to increase the robustness of the ROM with respect to parameter changes [146, 148, 149, 167]. In [12], an approach based on an interpolation on a Grassmann manifold was proposed to adapt precomputed POD bases to new parameter values. This method was further investigated in [10, 11, 13, 67]. An alternative approach was studied in [104, 105], where the sensitivity of the POD modes with respect to the problem parameters was calculated and associated sensitivity modes were included in the ROM. Since *a priori* the POD modes do not contain any information about the time evolution or the dynamical structure of the system, in [140], the author extended the general principal interaction patterns (PIPs) method [103]. In PIPs, the optimal basis is obtained from a nonlinear minimization procedure based on a dynamical optimality criterion involving higher-order correlation tensors of both the state variables and their time derivatives.

However, when complex flows are considered, it was recognized early on that a simple Galerkin truncation of the POD basis will generally produce inaccurate results, even if the retained POD modes capture most of the system's energy. In other words, choosing an appropriate POD basis will not generally be sufficient for developing an accurate POD-ROM of complex flows. To overcome this issue, various *closure* methods have been proposed to provide a more precise low-order system for the coefficients $\mathbf{a}(t)$. This is the main topic of this dissertation. We will first summarize the relevant approaches in Section 1.4. Then, in the next chapters, we will propose two novel closure POD-ROMs, develop new numerical discretization algorithms for nonlinear closure models, investigate the numerical behavior and build solid mathematical foundations for these models.

The organization of the rest of this chapter is as follows: Section 1.2 presents the standard POD method. In Section 1.3, the conventional POD-G-ROM is introduced and the associated error analysis is presented. Section 1.4 summarizes the closure methods that improve the POD-G-ROM. Several efficient numerical discretization algorithms for POD-ROMs are presented in Section 1.5. After overviewing the current POD reduced-order modeling techniques, we describe the “perfect” POD-ROM in Section 1.6. The dissertation is outlined in Section 1.7.

1.2 Proper Orthogonal Decomposition (POD)

In this section, we briefly describe the proper orthogonal decomposition method, following [137]. For a detailed presentation, the reader is referred to [108, 206, 207, 208].

Assume that $\mathbf{y}(x, t) \in L^2(\mathcal{H}, (t_0, t_0 + T))$, i.e., $\int_{t_0}^{t_0+T} |\mathbf{y}(\cdot, t)|^2 dt < \infty$. Given the time instances $t_1, \dots, t_M \in [0, T]$, we consider the ensemble of snapshots

$$\mathcal{R} := \text{span} \{\mathbf{y}(\cdot, t_1), \dots, \mathbf{y}(\cdot, t_M)\},$$

with $\dim \mathcal{R} = d$. The POD method seeks a low-dimensional (r) basis $\{\varphi_1, \dots, \varphi_r\}$ that optimally approximates the input collection in the sense that

$$\min \frac{1}{M} \sum_{\ell=1}^M \left\| \mathbf{y}(\cdot, t_\ell) - \sum_{j=1}^r (\mathbf{y}(\cdot, t_\ell), \varphi_j(\cdot))_{\mathcal{H}} \varphi_j(\cdot) \right\|_{\mathcal{H}}^2 \quad (1.6)$$

subject to the conditions that $(\varphi_i, \varphi_j)_{\mathcal{H}} = \delta_{ij}$, $1 \leq i, j \leq r \leq d$, where δ_{ij} is the Kronecker delta. In order to solve (1.6), we consider the eigenvalue problem

$$K v = \lambda v, \quad (1.7)$$

where $K \in \mathbb{R}^{M \times M}$, with $K_{k\ell} = \frac{1}{M} (\mathbf{y}(\cdot, t_\ell), \mathbf{y}(\cdot, t_k))_{\mathcal{H}}$, is the snapshot correlation matrix, v_j , $j = 1, \dots, d$, are the eigenvectors, and $0 < \lambda_d \leq \dots \leq \lambda_2 \leq \lambda_1$ are the positive eigenvalues.

It can then be shown that the solution of (1.6) is given by

$$\varphi_j(\cdot) = \frac{1}{\sqrt{\lambda_j}} \sum_{\ell=1}^M (v_j)_\ell \mathbf{y}(\cdot, t_\ell), \quad 1 \leq j \leq r, \quad (1.8)$$

where $(v_j)_\ell$ is the ℓ -th component of the eigenvector v_j . It can also be shown that the following error formula holds

$$\frac{1}{M} \sum_{\ell=1}^M \left\| \mathbf{y}(\cdot, t_\ell) - \sum_{j=1}^r (\mathbf{y}(\cdot, t_\ell), \varphi_j(\cdot))_{\mathcal{H}} \varphi_j(\cdot) \right\|_{\mathcal{H}}^2 = \sum_{j=r+1}^d \lambda_j. \quad (1.9)$$

Remark 1 A popular choice of \mathcal{H} is the L^2 space. Since $\frac{1}{T} \int_{t_0}^{t_0+T} \|\mathbf{y}(\cdot, t)\|_2^2 dt = \sum_{j=1}^d \lambda_j$, which is known as the POD energy, we can compute the relative error

$$\mathcal{E}_r = \frac{\frac{1}{T} \int_{t_0}^{t_0+T} \left\| \mathbf{y}(\cdot, t) - \sum_{j=1}^r (\mathbf{y}(\cdot, t), \varphi_j(\cdot))_{\mathcal{H}} \varphi_j(\cdot) \right\|_2^2 dt}{\frac{1}{T} \int_{t_0}^{t_0+T} \|\mathbf{y}(\cdot, t)\|_2^2 dt} = \frac{\sum_{j=r+1}^d \lambda_j}{\sum_{j=1}^d \lambda_j}.$$

Naturally, $0 \leq \mathcal{E}_{r+1} \leq \mathcal{E}_r \leq 1$. In practice, \mathcal{E}_r is usually used in a criterion to heuristically determine the number of POD modes that should be kept in the ROMs.

Remark 2 When a finite element approach is utilized to discretize (1.2), the snapshot set Y is composed of $\mathbf{y}(x, t)$, i.e., $Y = [\mathbf{y}(\cdot, t_1), \dots, \mathbf{y}(\cdot, t_M)]$. We can do SVD on the matrix $\tilde{Y} = M_h^{\frac{1}{2}} Y$, where M_h is the mass matrix. Then the POD basis can be computed as $\boldsymbol{\varphi}_j = M_h^{-\frac{1}{2}} U_j$, where U_j is the j -th column of the left singular vector of \tilde{Y} . For a detailed discussion of this case, the reader is referred to [136].

Remark 3 The POD method can be extended to the case where a weight function is applied on various snapshots, degrees of freedom, or both. This can be achieved by posing the problems in a weighted inner product space $\langle x, y \rangle_W \equiv \langle x, Wy \rangle$ where W is a given symmetric positive definite matrix (see, e.g., [63]).

1.3 POD Reduced-Order Model

In this section, taking incompressible Newtonian fluid flows as an example, we first derive the standard POD-G-ROM for the NSE in Subsection 1.3.1. Then we summarize the existing numerical discretization error estimates of POD-ROMs in Subsection 1.3.2.

1.3.1 Galerkin Projection

Reduced-order models of structurally dominated turbulent flows are central to many applications in science and engineering. Here we consider complex incompressible fluid flows governed by the NSE:

$$\begin{cases} \mathbf{u}_t - \text{Re}^{-1} \Delta \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla p = 0 \\ \nabla \cdot \mathbf{u} = 0, \end{cases} \quad (1.10)$$

where \mathbf{u} is the velocity, p the pressure, and Re the Reynolds number. Appropriate boundary conditions are chosen such that boundary integrals vanish in the weak form. For example, in the flow past a cylinder problem, the flow satisfies a non-homogeneous Dirichlet inflow boundary condition, the “do-nothing” outflow boundary condition, and no-slip boundary conditions on the top and the bottom walls.

By collecting snapshots of the velocity field and applying the POD method, a POD-ROM of the flow is constructed from the POD basis by writing

$$\mathbf{u}(\mathbf{x}, t) \approx \mathbf{u}_r(\mathbf{x}, t) \equiv \mathbf{U}(\mathbf{x}) + \sum_{j=1}^r a_j(t) \boldsymbol{\varphi}_j(\mathbf{x}), \quad (1.11)$$

where $\mathbf{U}(\mathbf{x})$ is the centering trajectory. We now replace the velocity \mathbf{u} with \mathbf{u}_r in (1.10), and then project the resulting equations onto the subspace \mathbf{X}^r (1.3) spanned by the POD basis. We note that, by construction (1.8), the POD modes are linear combinations of the snapshots. Since the velocity field is solenoidal ($\nabla \cdot \mathbf{u} = 0$), the POD basis functions are also solenoidal, i.e., $\nabla \cdot \boldsymbol{\varphi} = 0, \forall \boldsymbol{\varphi} \in \mathbf{X}^r$.

Using the boundary conditions and the fact that all modes are solenoidal, one obtains the POD-G-ROM for the NSE:

$$\left(\frac{\partial \mathbf{u}_r}{\partial t}, \boldsymbol{\varphi} \right) + ((\mathbf{u}_r \cdot \nabla) \mathbf{u}_r, \boldsymbol{\varphi}) + \left(\frac{2}{\text{Re}} \mathbb{D}(\mathbf{u}_r), \nabla \boldsymbol{\varphi} \right) = 0, \quad \forall \boldsymbol{\varphi} \in \mathbf{X}^r, \quad (1.12)$$

where $\mathbb{D}(\mathbf{u}_r) := (\nabla \mathbf{u}_r + (\nabla \mathbf{u}_r)^T) / 2$ is the deformation tensor of \mathbf{u}_r . We note that, since the computational domains that we consider are large enough, the pressure terms can be neglected in (1.12) (for details, see [8, 175]). The POD-G-ROM (1.12) yields the following autonomous dynamical system for the vector of time coefficients, $\mathbf{a}(t)$:

$$\dot{\mathbf{a}} = \mathbf{b} + \mathbf{A}\mathbf{a} + \mathbf{a}^T \mathbf{B}\mathbf{a}, \quad (1.13)$$

where \mathbf{b} , \mathbf{A} , and \mathbf{B} correspond to the constant, linear, and quadratic terms in the numerical discretization of the NSE (1.10), respectively. The initial conditions are obtained by projection:

$$a_i(0) = \langle \boldsymbol{\varphi}_i, \mathbf{u}(\cdot, 0) - \mathbf{U}(\cdot) \rangle_{\mathcal{H}}, \quad i = 1, \dots, r. \quad (1.14)$$

The finite dimensional system (1.13) can be written componentwise as follows: For all $i = 1, \dots, r$,

$$\dot{a}_i(t) = b_i + \sum_{m=1}^r A_{im} a_m(t) + \sum_{m=1}^r \sum_{n=1}^r B_{imn} a_n(t) a_m(t), \quad (1.15)$$

where

$$b_i = -(\varphi_i, \mathbf{U} \cdot \nabla \mathbf{U}) - \frac{2}{\text{Re}} \left(\nabla \varphi_i, \frac{\nabla \mathbf{U} + \nabla \mathbf{U}^T}{2} \right), \quad (1.16)$$

$$A_{im} = -(\varphi_i, \mathbf{U} \cdot \nabla \varphi_m) - (\varphi_i, \varphi_m \cdot \nabla \mathbf{U}) - \frac{2}{\text{Re}} \left(\nabla \varphi_i, \frac{\nabla \varphi_m + \nabla \varphi_m^T}{2} \right), \quad (1.17)$$

$$B_{imn} = -(\varphi_i, \varphi_m \cdot \nabla \varphi_n). \quad (1.18)$$

1.3.2 Mathematical Foundations

The first steps in developing a mathematical theory for the numerical discretization of the POD-G-ROM were made in [137]. In that paper, the authors have considered a POD basis approximation in space (without specifying approximation properties) and various time discretizations (forward Euler, backward Euler, and Crank-Nicolson) of the Burgers equation. For these discretizations, they have rigorously proven that the discretization error comprises three parts: (i) the interpolation error of the initial solution; (ii) the usual time discretization error (Δt^α where $\alpha = 1$ for the forward and backward Euler methods and $\alpha = 2$ for the Crank-Nicolson method); and (iii) the POD Galerkin truncation error. The error estimation reads:

$$\|u(t_k) - u_k^{POD-G}\| \leq C \left(\|u(t_0) - u_0^{POD-G}\| + \Delta t^\alpha + \sqrt{\sum_{j=r+1}^d \lambda_j} \right), \quad (1.19)$$

where $u(t_k)$ and u_k^{POD-G} are the exact solution and the POD-G-ROM solution at time instance t_k , respectively, Δt is the time step, and λ_j is the j -th positive eigenvalue of the snapshot correlation matrix K . In a follow-up paper [138], these results were expanded and the same type of error estimates were proved for the two dimensional (2D) NSE. Furthermore, two different time discretizations in the generation of snapshots and the numerical integrations of POD-G-ROM were considered. In [190], error estimates for the NSE were derived, this time, however, including the spatial finite element discretization error. In [191], error estimates of POD-ROMs for the Boussinesq equations were studied. Both implicit and semi-implicit backward Euler methods for time integrations were employed and analyzed.

A different approach was used in [155]. The authors utilized a mixed finite element discretization in space and a semi-implicit Euler approximation in time, and proved the

following error estimate:

$$\begin{aligned} & \|u(t_k) - u_k^{POD-G}\| + \Delta t^{\frac{1}{2}} \|p(t_k) - p_k^{POD-G}\| + \Delta t^{\frac{1}{2}} \|\nabla (u(t_k) - u_k^{POD-G})\| \\ & \leq C \left(\Delta t + h^m + \sqrt{\Delta t^{\frac{1}{2}} \sum_{j=r+1}^d \lambda_j} \right), \end{aligned} \quad (1.20)$$

where $u(t_k)$ and $p(t_k)$ is the exact solution of the NSE, u_k^{POD-G} and p_k^{POD-G} is the POD-ROM solution at time instance t_k , respectively, h is the mesh size, and m is the order of polynomials used to approximate the velocity space. The authors also concluded that “it is unnecessary to take total transient solutions at all time instances as snapshots.” Instead, they only took snapshots that are useful and of interest. In [157], the same method was applied to the upper tropical Pacific Ocean model and similar error estimates were shown.

In the above reports, the snapshots are assumed to be from the same system as that for which the POD approximation properties are analyzed. Due to the application of parametrized differential equations in flow control and optimization, error analysis for parametric reduced-order systems also has been considered. In [84], the SPOD approximability of 2D parametric NSE were studied. In [186], the influence of small perturbations in snapshots on POD-ROMs for an ODE system were analyzed. In [109, 110], the authors considered an ODE system with perturbations in either model parameters or initial conditions and measured the *a priori* error bounds in corresponding POD-ROMs. The proposed approach is based on the small sample statistical method and a linear adjoint method. Moreover, the error bound can indicate ranges of perturbations in the original system over which the reduced model is still appropriate. The method was extended in [204] to a system of differential-algebraic equations. In [97, 171, 184, 198], the RBM, as a powerful parametric model reduction tool was analyzed for *a posteriori* error bounds.

Although the first steps have already been made, the mathematical theory for the numerical discretization of POD-G-ROM is not as mature and developed as that for traditional discretization methods, such as the finite element method.

1.4 POD Closure Modeling

The POD reduced-order modeling approach has been used successfully in the numerical simulation of laminar flows. In [65], the dynamics of the flow past a cylinder was modeled with an eight-dimensional POD-G-ROM. In [161], the POD-G-ROM was generalized to predict the 3D transition initiated by the Mode A instability (see [229] for the definition of the Mode A instability) at $\text{Re} \approx 180$. The POD-G-ROM results obtained in these studies are in agreement with the numerical simulations, at least for short-time integration. However, *long-time integration* of the POD-G-ROM might not produce the limit cycles obtained by the computational fluid dynamics (CFD) code. The solution can drift to some erroneous state even if it is initialized with the correct periodic state. This instability is associated with the presence of multiple spurious limit cycles. In [82], the existence of multiple spurious steady states in the Galerkin expansion of the Kuramoto-Sivashinsky equation were investigated. For a similar equation, in [21], spurious states were also found in the POD model that captured 99.99% of the system's energy.

In fact, from the very beginning of POD reduced-order modeling of fluid flows, it was recognized that a simple Galerkin truncation of the POD basis will generally produce erroneous results for turbulent flows [20]. The reason for the inadequate behavior of the POD Galerkin truncation is that, although the discarded POD modes $\{\varphi_{r+1}, \dots, \varphi_d\}$ do not contain a significant amount of the kinetic energy in the system, they do, however, have a significant role in the dynamics of the reduced-order system. Indeed, the interaction between the discarded POD modes $\{\varphi_{r+1}, \dots, \varphi_d\}$ and the POD modes retained in the ROM $\{\varphi_1, \dots, \varphi_r\}$ is essential for an accurate prediction of the dynamics of the ROM. This situation is similar to the traditional Fourier setting for turbulence, in which the effect of the discarded Fourier modes needs to be modeled, i.e., one needs to solve the celebrated *closure problem*. This similarity is not surprising, since in the limitation of homogenous flows, the POD basis reduces to the Fourier basis [108].

To address the POD-ROM closure problem, various approaches have been proposed. These closure models can be roughly classified into three main categories: (i) calibration (data fitting) methods; (ii) numerical stability enhancing closure models; and (iii) physics

based closure models. Although it is generally hard to separate the three classes of closure models, their development roots in different foundations.

Calibration Methods A closure modeling technique is the *calibration* approach, in which the polynomial coefficients of the POD-G-ROM are determined from an optimization problem. Galletti et al. [86, 87] introduced this procedure to determine the coefficient of the pressure term in the POD-G-ROM for the NSE. A linear model for the pressure term was considered:

$$\begin{cases} \dot{\mathbf{a}} = \mathbf{b} + (\mathbf{A} + \mathbf{C}) \mathbf{a} + \mathbf{a}^T \mathbf{B} \mathbf{a}, \\ a_i(0) = \langle \boldsymbol{\varphi}_i, \mathbf{u}(\cdot, 0) - \mathbf{U}(\cdot) \rangle_{\mathcal{H}} \text{ for } i = 1, \dots, r, \end{cases} \quad (1.21)$$

where the entry C_{im} satisfies the linear model for pressure

$$C_{im} a_i(t) = -(\boldsymbol{\varphi}_i, \nabla p). \quad (1.22)$$

Let $\mathbf{a}_j^\varepsilon(t_\ell)$ be the projection of ℓ -th snapshot onto the j -th POD basis. $\hat{\mathbf{a}}_j(t)$ is the spline interpolation of $\{\mathbf{a}_j^\varepsilon(t_\ell)\}_{\ell=1}^M$, which is regarded as a benchmark trajectory. Then the coefficient matrix \mathbf{C} is determined by minimizing the functional

$$J = \sum_{i=1}^r \int_{t_0}^{t_0+T} (a_i(t) - \hat{a}_i(t))^2 dt \quad (1.23)$$

under the constraints (1.21). In [173, 175], Noack et al. demonstrated that the pressure term is vital for a 2D mixing layer problem and its effect can be well represented by a linear term. Thus, the same ansatz as (1.22) was also employed in [173, 175] for a POD-G-ROM, but with a different optimization problem to calibrate \mathbf{C} . Instead of minimizing the difference between the Galerkin solution and the DNS projection, they minimized the difference of the pressure term in primitive variables from that in the Galerkin model representation. This model was further studied in [43] for a 3D wake flow.

The calibration method (1.23) is termed a state calibration method with dynamical constraints in [59], where Couplet et al. proposed two other calibration models — a state calibration method and a flow calibration method. All polynomial coefficients (linear and quadratic) in the POD-G-ROM (1.5) are calibrated. The calibrated f minimizes a functional \mathcal{J}^α where

$$\mathcal{J}^\alpha = (1 - \alpha)\mathcal{E}(f) + \alpha\mathcal{D}(f). \quad (1.24)$$

The term \mathcal{E} in (1.24) measures the “error” between snapshots and the dynamical system, and the term \mathcal{D} is related to the difference between the calibrated coefficients and the coefficients of the original POD-G-ROM. Therefore, when $\alpha = 1$, no calibration is employed, while when $\alpha = 0$, the calibration model is fully optimized. Besides a 2D flow past a square cylinder problem at $\text{Re} = 100$, a 3D flow past a backward-facing step case at $\text{Re} = 7432$ was also tested in [59].

Since the minimization problem is usually badly conditioned, a Tikhonov regularization ([9, 102]) was proposed in [226] to improve the robustness of the calibration approach. Snapshots at several realizations were used to generate the POD basis. However, the authors claimed that, “the actual real improvement in robustness is obtained by spanning the solution manifold, i.e., by including several control laws in the inverse problem definition.” Such an additional regularization actually makes the inverse problem well conditioned. This technique was further used in the feedback control of incompressible flows in [227]. The regularization technique was further studied in [58], where Cordier et al. summarized several existing calibration methods, including the intrinsic stabilization method proposed in [129], and compared them in the configuration of a 2D flow past a cylinder at $\text{Re} = 200$.

Closure Models that Enhance the Numerical Stability This approach in devising closure models aims at enhancing the numerical stability of POD-ROMs. In the POD-G-ROM of a flow past a cylinder, Sirisup and Karniadakis [205] showed that the onset of divergence from the correct limit cycle depends on the number of modes in the Galerkin expansion, the Reynolds number, and the flow geometry. They used a spectral vanishing viscosity method (SVVM) [212], which adds a small amount of mode-dependent dissipation satisfying the entropy condition while retaining the spectral accuracy. Thus, equation (1.5) is modified as

$$\dot{\mathbf{a}} = \mathbf{f}(\mathbf{a}(t), t) - \mathbf{H}(y; \epsilon, Q_\epsilon), \quad (1.25)$$

where $\epsilon \rightarrow 0$ is a viscosity amplitude and Q_ϵ is a viscosity convolution kernel. The SVVM is typically applied to the higher modes and the numerical value of ϵ depends on the number of modes for which the SVVM is activated. The parameters for the SVVM model are found by

an empirical method and a bifurcation analysis. However, their exact values are not known *a priori* and depend on the flow geometry and the number of POD modes.

A different closure modeling approach, which was used in [117, 118], replaces the L^2 inner product in the generation of the POD basis with the H^1 inner product. Therefore, the gradient information is also incorporated in the POD modes. Several numerical tests showed that the method actually stabilized the POD-G-ROM and “some actions of small eddies were recovered” [118]. Another stabilization method based on a Lax-Wendroff type artificial diffusion term was also proposed in [117] for compressible flows.

The numerical stabilization approach introduced in [29] is based on the residuals of the Navier-Stokes operator evaluated with the reduced flow field. Extra POD stabilization modes were extracted from the residual snapshots and used to stabilize the POD-G-ROM. In the same paper, the traditional streamline upwind Petrov Galerkin method and a residual based variational multiscale [26] numerical stabilization technique for convection-dominated flows yielded stable POD-ROMs.

Closure Models Based on Physical Insight Given the similarity between the POD and Fourier representations of vector fields [108], it is not surprising that the first closure model was based on an *eddy viscosity (EV)* assumption. Indeed, in [20], Aubry et al. used a Heisenberg type of closure model for the boundary layer, which truncated the POD basis and used an EV approximation to model the effect of the discarded POD modes on the POD modes kept in the model. To be precise, this model stipulates that the anisotropic small-scale stress tensor must be proportional to the strain rate of the resolved scales $\mathbb{D}(\mathbf{u}_<)$:

$$\tau_> = \tilde{\nu}_S \mathbb{D}(\mathbf{u}_<), \quad (1.26)$$

and

$$\tilde{\nu}_S = 2\alpha U_> L_>, \quad (1.27)$$

where $\mathbf{u}_<$ represents the resolved scale, $U_>$ and $L_>$ are velocity and length scales characteristic of the unresolved modes, respectively, and $\alpha \sim \mathcal{O}(1)$ is a non-dimensional parameter that characterizes the amount of energy being dissipated.

This mixing-length closure model (1.26)-(1.27) yielded truly impressive results considering the coarseness of the approximation and the complexity of the flow. The criterion used

to assess the accuracy of the model was the intermittency of bursting events in the turbulent boundary layer. This POD-ROM was further investigated numerically in a series of papers by Podvin and Lumley [154, 180, 181, 182] and Rempfer [192, 193, 194].

Further improvements of the POD-ROM proposed in [20] were made in [52, 53], where the dissipation coefficient was defined by

$$\frac{1}{\widetilde{\text{Re}}_j} = \max \left\{ \frac{1}{\text{Re}}, \frac{1}{\widetilde{\text{Re}}_j} \right\}, \quad (1.28)$$

and

$$\frac{1}{\widetilde{\text{Re}}_j} = \frac{1}{\text{Re}} + \frac{D_j}{\langle \nabla \boldsymbol{\varphi}_j \nabla \boldsymbol{\varphi}_j \rangle_{\Omega}}. \quad (1.29)$$

The constant D_j in (1.29) is the eddy viscosity of the j -th POD mode $\boldsymbol{\varphi}_j$ determined from the energy conservation equation of the resulting closure model, and $\langle \cdot \rangle_{\Omega}$ denotes an integration over the computational domain Ω .

Continuing the closure modeling approach based on the analogy to turbulence modeling, a further step was taken in [35, 38], where a Smagorinsky type of closure model was proposed (see also [174, 215] for similar approaches). The small-scale stress tensor is modeled by

$$\boldsymbol{\tau}_{>} = \tilde{\nu}_S \mathbb{D}(\mathbf{u}_{<}), \quad (1.30)$$

and

$$\tilde{\nu}_s = 2(C_S \delta)^2 \|\mathbb{D}(\mathbf{u}_{<})\|, \quad (1.31)$$

where C_S is the Smagorinsky constant, δ is the length scale, $\|\mathbb{D}(\mathbf{u}_{<})\|$ is the Frobenius norm of the strain rate of resolved scales. The advantage of using a variable (in both time and space) EV term over the simple mixing length term used in [20] will be illustrated in Chapter 4 and Chapter 5.

Noack et al. [172, 175] have applied the EV together with other mechanisms in a single framework. The resulting closure modeling strategy reveals two important amplitude-selection mechanisms. One is based on the mean-field deformation due to the fluctuation. This mechanism is particularly dominant in the neighborhood of the steady solution. The other process is the energy flow from large-scale modes to smaller-scale modes in the spirit of the energy cascade. The role of this process increases with the amplitude of oscillation.

Although much remains to be done, it is clear that EV closure models significantly improve the performance of the POD-ROMs. Obviously, the most plausible explanation for their success is the analogy between the POD and Fourier descriptions of the flows: the two descriptions are identical for homogeneous flows [108]. Thus, since EV has been one of the most popular closure models used in the Fourier description of turbulent flows, it is natural that a similar EV approach should also work in the POD setting. There is, however, one major issue that needs to be addressed first. The foundations of the EV modeling in the Fourier setting have been laid on the well tested, both experimentally and numerically, concept of *energy cascade*, i.e., the fact that energy flows from small wave numbers to large wave numbers. Thus, to legitimate the use of the EV modeling approach in a POD setting, one would first need to show that the energy cascade concept is also valid in the POD setting.

Surprisingly enough, the first and, to our knowledge, the only paper to address this issue was the report of Couplet et al. in [60]. By monitoring the triad interactions due to the nonlinear term in the NSE, the authors have concluded that the transfer of energy among the POD modes is similar to the transfer of energy among Fourier modes. Specifically, they found that there is a net forward energy transfer from low index POD modes to higher index POD modes and that this transfer of energy is local in nature (that is, energy is mainly transferred among POD modes whose indices are close to one another). This study (see also [174]) clearly suggests that *large eddy simulation (LES)* ideas based on the energy cascade concept could also be used in devising POD-ROMs. This paper together with the success of the EV POD-ROMs in practical computations [20, 30, 35, 38, 52, 53, 192, 193, 194] suggest that the physical accuracy of the EV assumption coupled with the computational efficiency of the POD could yield appropriate ROMs for turbulent flows dominated by organized structures.

Remark 4 *Although stemming from different mathematical and physical foundations, these three classes of closure methods are sometimes applied together. In [28, 30], the POD-G-ROM was stabilized by introducing a time-dependent artificial viscosity on each POD mode and the coefficients were determined by a calibration approach. In [213], the EV closure modeling proposed in [194] was utilized together with a dynamic calibrating method to improve the POD-G-ROM.*

1.5 Efficient Numerical Discretizations

The computational efficiency of the POD-G-ROM relies on the feasibility of precomputations of vectors and matrices in the ROM. However, for problems involving nonpolynomial nonlinearities, the efficiency of POD-ROMs is severely limited. Examples of such problems include the Buckley-Leverett equation [170] and nonlinear closure models such as Smagorinsky (1.30)-(1.31), variational multiscale, and dynamic subgrid-scale. To overcome this difficulty, several methods have been proposed.

The best-points interpolation method (BPIM) was developed in [169]. It was applied in [85] to develop an approach to uncertainty quantification in a nonlinear combustion problem. The missing points estimation method (MPE) was introduced in [18, 19]. In this approach, the full-order system was first reduced by choosing equations only corresponding to certain selected spatial points and restricting the POD basis onto these points, then projecting the extracted system onto the space spanned by the POD basis. The grid selection procedure is, obviously, vital in this method. Two heuristic approaches were proposed in [18] and improved in [228] with the help of a greedy algorithm. The trajectory piecewise-linear method (TPWL), presented in [195, 196], reduces a nonlinear model by a weighted sum of linearized models at selected points along a state trajectory.

An alternative approach widely used in the RBM is the EIM proposed in [24]. It was first used to approximate non-affine differential operators to enable an efficient offline-online computational strategy, and then was further applied to approximate nonlinear differential operators in [96]. A discrete variant of EIM, the discrete empirical interpolation method (DEIM), which combines EIM with the POD method, was introduced in [55] for model reduction of nonlinear dynamical systems and analyzed in [54]. This method approximates a general nonlinear function and greatly improves the efficiency of the POD Galerkin projection.

1.6 “Perfect” POD-ROM for Complex Flows

The goal of this dissertation is to develop POD-ROMs for structurally dominated turbulent flows in engineering and geophysical applications. The development of POD-ROMs for engineering and geophysical flows is a vast research area, and many reports on this topic are published every year. This section aims at clearly outlining the overarching goal of the research we started in this dissertation. To this end, we define what we believe the “perfect” POD-ROM for complex flows is. We emphasize that this “perfect” POD-ROM, which is schematically illustrated in Figure 1.1, represents what in our opinion is needed for *accurate*, *efficient*, and *robust* numerical simulation of complex flows. That being said, we acknowledge right from the start that other research groups might use different criteria to define “perfect” POD-ROMs.

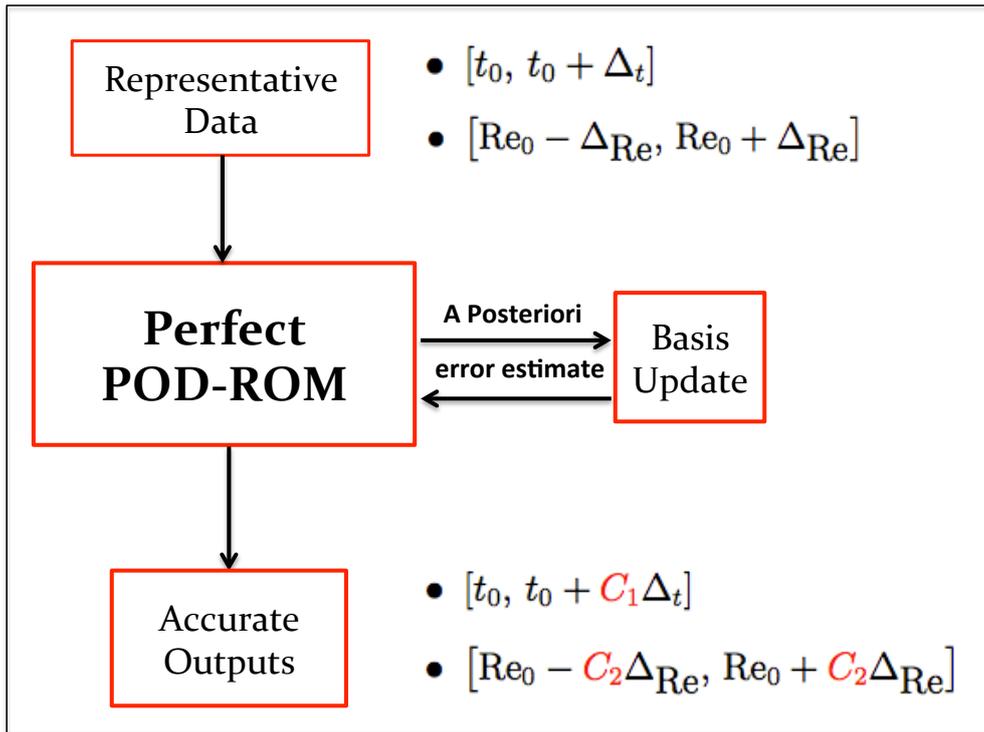


Figure 1.1: Schematic for the “perfect” POD-ROM for complex flows.

The input for the “perfect” POD-ROM is representative data for the targeted complex flow (e.g., structurally dominated turbulent flows, as those considered in this dissertation). This data is a collection of flow variables over appropriate time and parameter ranges. For

example, one could consider the time interval $[t_0, t_0 + \Delta_t]$ and Reynolds number interval $[\text{Re}_0 - \Delta_{\text{Re}}, \text{Re}_0 + \Delta_{\text{Re}}]$. Given this representative data, the “perfect” POD-ROM should be able to provide an accurate, efficient, and robust approximation of the flow field. That is, the “perfect” POD-ROM’s output should satisfy the following criteria:

- (i) *accuracy*, i.e., produce results that are close to those predicted by the full-order model;
- (ii) *efficiency*, i.e., the results should be achieved at a fraction of the computational cost of a full-order model; and
- (iii) *robustness*, i.e., the results should be valid for much larger time intervals $[t_0, t_0 + C_1\Delta_t]$ and much larger parameter intervals $[\text{Re}_0 - C_2\Delta_{\text{Re}}, \text{Re}_0 + C_2\Delta_{\text{Re}}]$, where the constants C_1 and C_2 are very large (e.g., $C_1, C_2 \sim \mathcal{O}(1000)$).

The first two criteria we used in the definition of the “perfect POD-ROM” (i.e., accuracy and efficiency) would probably be used in any standard definition of a “perfect” POD-ROM. We believe, however, that the third criterion we used (i.e., robustness) is what sets our goals apart from other POD-ROM approaches. Indeed, if we restricted the “perfect” POD-ROM to the *same* time interval $[t_0, t_0 + \Delta_t]$ and the *same* parameter interval $[\text{Re}_0 - \Delta_{\text{Re}}, \text{Re}_0 + \Delta_{\text{Re}}]$ as those used in the generation of the input data, then standard interpolation techniques could be used to ensure that the “perfect” POD-ROM’s trajectory is as close as possible to the input data. This, in our opinion, would be nothing more than data fitting. The “perfect POD-ROM”, we believe, should be able to work on much larger time intervals and much larger parameter intervals than those used in the derivation of the input data. This is the only way POD can fulfill its original promise of becoming a *predictive* tool in *realistic* engineering and geophysical applications. An obvious criticism to our “perfect” POD-ROM would be that the POD basis generated on the intervals $[t_0, t_0 + \Delta_t]$ and $[\text{Re}_0 - \Delta_{\text{Re}}, \text{Re}_0 + \Delta_{\text{Re}}]$, while optimal on these intervals, might be completely inappropriate on the intervals $[t_0, t_0 + C_1\Delta_t]$ and $[\text{Re}_0 - C_2\Delta_{\text{Re}}, \text{Re}_0 + C_2\Delta_{\text{Re}}]$. Indeed, one can simply imagine a physical structure (e.g., a turbulent burst) that appears in the time interval $[t_0 + 2\Delta_t, t_0 + 3\Delta_t]$ and, thus, cannot be captured by the input POD basis. To address this well-founded (and well known) criticism of POD-ROMs, we endowed our “perfect” POD-ROM with a basis updating mechanism. A

natural approach for deciding when this basis updating is needed might use *a posteriori* error indicators. We emphasize, however, that this updating mechanism should focus exclusively on generating an improved basis; it should *not* aim at generating an *improved model*. In fact, updating the basis as well as the model itself would amount, in some sense, to considering the same time and parameter intervals as those used in the generation of the input data (i.e., data fitting); Our definition of “perfect” POD-ROM avoids that.

The “perfect” POD-ROM outlined in this section and illustrated schematically in Figure 1.1 represents the overarching goal (the “Holy Grail”) of our research. To our knowledge, nobody has been able to develop this “perfect” POD-ROM. In this dissertation we take a first step in this direction. We use the wealth of physical insight built over that last decades in the LES community to develop computationally efficient and physically accurate POD-ROMs that can run over much larger time intervals than those over which the POD basis was generated. We also develop new algorithms that make possible the efficient numerical discretization of the new POD-ROMs. Finally, we develop a rigorous mathematical support for the finite element discretization of the new POD-ROMs.

Much remains to be done to achieve the quest for the “perfect” POD-ROM. We believe, however, that the steps taken in this dissertation lay a solid foundation for exciting, new developments in the years to come.

1.7 Outline of dissertation

In this dissertation, we develop state-of-the-art EV closure methods for POD-ROMs, which greatly improve the accuracy of the standard POD-G-ROM. However, since modern closure models for turbulent flows are generally nonlinear, their efficient numerical discretization within a POD framework is still challenging. We propose a new two-level method to overcome this computational hurdle. We also discuss the applications of the new POD-ROMs in realistic applications, such as airflow simulations in energy efficient building design and control problems as well as numerical simulations of large-scale ocean motions in climate modeling.

The rest of the dissertation is organized as follows:

1. In Chapter 2, two new state-of-the-art EV closure for POD-ROMs are proposed: the variational multiscale method and the dynamical subgrid-scale method. Two other EV closure POD-ROMs are presented: the mixing-length model [20] and the Smagorinsky model [38, 174, 215, 223]. Related concepts such as the POD filter and lengthscale are also defined.

The material in this chapter is based on the following report:

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.*, in press, 2012.

2. In Chapter 3, the new two-level discretization algorithms for general nonlinear POD closure models are introduced. As a numerical verification of the efficiency and accuracy of the new algorithms, three tests are performed: (i) the 1D Burgers equation with a small dissipation parameter $\nu = 10^{-3}$; (ii) a 2D flow past a circular cylinder at $\text{Re} = 200$; and (iii) a 3D flow past a circular cylinder at $\text{Re} = 1000$.

The material in this chapter is based on the following report:

Z. Wang, I. Akhtar, J. Borggaard and T. Iliescu. Two-Level Discretizations of Nonlinear Closure Models for Proper Orthogonal Decomposition. *J. Comput. Phys.*, 230(1), 2011, pp. 126-146.

3. In Chapter 4, we develop solid mathematical foundations for EV closures of POD-ROMs by analyzing the numerical errors in the finite element discretization of the Smagorinsky and the variational multiscale methods in two different systems: the Burgers equation and the convection-dominated convection-diffusion-reaction equations. The error estimates show the convergence properties of the new POD-ROMs.

The material in this chapter is based on the following reports:

J. Borggaard, T. Iliescu and Z. Wang. Artificial Viscosity Proper Orthogonal Decomposition. *Math. Comput. Model.*, 53 (1-2), 2011, pp. 269-279.

T. Iliescu and Z. Wang. Variational Multiscale Proper Orthogonal Decomposition: Convection-Dominated Convection-Diffusion Equations. *Math. Comp.*, in press, 2011.

4. In Chapter 5, a structurally dominated 3D turbulent flow problem is used to test the four POD-ROMs described in Chapter 2. These POD closure models are compared with the standard POD-G-ROM, benchmarked by the DNS data, and judged by five criteria: (i) the kinetic energy spectrum; (ii) the mean velocity; (iii) the Reynolds stresses; (iv) the root mean square values of the velocity fluctuations; and (v) the time evolution of the POD coefficients. The numerical results show that the variational multiscale and the dynamic subgrid-scale models yield the most accurate average and instantaneous numerical results.

The material in this chapter is based on the following report:

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.*, in press, 2012.

5. In Chapter 6, as a part of our future work, we will discuss the applications of POD closure models to two realistic applications in engineering and geophysics: building airflow simulations and large-scale oceanic flow simulations. We also address other research avenues that we plan to pursue in the future.

The material in this chapter is based on the following reports:

O. San, A. E. Staples, Z. Wang and T. Iliescu. Approximate Deconvolution Large Eddy Simulation of a Barotropic Ocean Circulation Model. *Ocean Modelling*, 40, 2011, pp. 120-132.

E. Foster, T. Iliescu, and Z. Wang. A Finite Element Discretization of the Stream-function Formulation of the Stationary Quasi-Geostrophic Equations of the Ocean. Submitted, 2012.

Z. Wang and T. Iliescu. A Mixed Finite Element Method for the Time-Dependent Quasi-Geostrophic Equations of the Ocean. Submitted, 2012.

Z. Wang, J. Borggaard and T. Iliescu. Reduced-Order Modeling of Airflow in Energy-Efficient Buildings. In preparation, 2012.

Chapter 2

POD Closure Models for Turbulent Flows [†]

2.1 Introduction

Despite their initial success, POD-ROMs have generally been limited to laminar flows and relatively few reports on closure modeling strategies for turbulent flows have appeared in the literature [20, 38, 43, 53, 104, 105, 161, 172, 174, 175, 176, 180, 182, 192, 194, 205, 215]. This is in stark contrast to the amount of work done in traditional turbulence modeling, such as LES, where literally hundreds of closure models have been proposed and investigated (see, e.g., [199]) over the same time period. This disparity in closure modeling between POD reduced-order modeling and classical turbulence modeling seems even more dramatic considering that the concept of an energy cascade, which is a fundamental modeling principle in LES, is also valid in a POD setting.

In this chapter, we develop two novel POD-ROMs, inspired from state-of-the-art LES closure modeling strategies: the *dynamic subgrid-scale (DS) model* [88, 163, 183] and the *variational multiscale (VMS) model* [113]. We also consider the standard mixing-length closure model proposed in [20] and the Smagorinsky model proposed in [174, 215, 223], both

[†]The material in this chapter is based on the following report:

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.*, in press, 2012.

being standard LES closure models.

Before we describe the new closure methods for POD-ROMs, we emphasize that POD is *not* intended to compete against well established, successful methodologies for the numerical simulation of general turbulent fluid flows, such as LES [32, 199]. In fact, such a comparison between POD-ROM and LES would be questionable. Indeed, as shown in [108], POD reduces to the standard Fourier decomposition when homogeneous flows are considered. Thus, for homogeneous, isotropic turbulent flows, POD-ROM (and the associated closure modeling) practically reduces to LES. This clearly suggests that POD-ROM and LES have different goals: the latter targets general, homogeneous, isotropic turbulent flows, whereas the former aims at highly anisotropic turbulent flows, whose dynamics are dominated by *coherent structures*. Moreover, the spatial resolutions employed by POD-ROM are much coarser than those used in LES. These suggest that POD-ROM and LES could actually be used together: Indeed, for turbulent flows at high Reynolds numbers (for which a DNS is unfeasible), LES could be used to generate the snapshot matrix for the POD-ROM. This idea has been exploited in [60].

We also note that the POD-ROM approach that we pursue in this dissertation has several similarities to the stochastic adaptive large eddy simulation (SCALES) introduced in [89]. SCALES combines the coherent vortex simulation (CVS) methodology proposed in [78, 79] with subgrid-scale modeling ideas imported from LES. Specifically, the CVS applies a wavelet filter to a turbulent vorticity field and separates it into energetic coherent structures and a residual, subgrid field that is homogeneous. To overcome the relatively high computational cost of CVS [64], SCALES employs a wider support for the wavelet filter and then takes advantage of LES-type modeling techniques to account for the effect of those coherent structures that are now treated as subgrid-scale terms. Thus, POD-ROM and SCALES are similar in that they both target the coherent structures in the underlying turbulent flow. They are different, however, in that POD-ROM employs POD to generate these coherent structures, whereas SCALES uses wavelets. Moreover, POD-ROM aims at highly non-homogeneous, anisotropic turbulent flows, whereas SCALES considers general turbulent flows. Finally, POD-ROM employs spatial resolutions that are coarser than those used by SCALES and CVS.

The rest of the chapter is organized as follows: Section 2.2 presents the definitions of the POD filter and lengthscale in the flow field approximated by the reduced-order model. Four POD closure models are described in Section 2.3. Finally, conclusions and several research directions that we are currently pursuing are provided in Section 2.4. All these four POD-ROMs will be tested in the numerical simulation of a 3D turbulent flow around a circular cylinder at $Re = 1000$ in Chapter 5.

2.2 Filter and Lengthscale

We start by describing the filtering operation utilized and the spatial lengthscale δ used in the POD closure models. Both are needed in order to define meaningful LES-inspired POD closure models.

POD Filter In LES, the filter is the central tool used to obtain simplified mathematical models that are computationally tractable. The filtering operation is effected by convolution of flow variables with a rapidly decaying *spatial* filter g_δ , where δ is the radius of the spatial filter. In the POD, however, there is no explicit spatial filter used. Thus, in order to develop LES-type POD closure models, a POD filter needs to be introduced. Given the hierarchical nature of the POD basis, a natural such filter appears to be the Galerkin projection. Let \mathbf{X} be an appropriate Hilbert space, \mathbf{X}^r the subspace spanned by the first r POD basis functions as (1.3). For all $\mathbf{u} \in \mathbf{X}$, the Galerkin projection $\bar{\mathbf{u}} \in \mathbf{X}^r$ is the solution of the following equation:

$$(\mathbf{u} - \bar{\mathbf{u}}, \boldsymbol{\varphi}) = 0, \quad \forall \boldsymbol{\varphi} \in \mathbf{X}^r. \quad (2.1)$$

The Galerkin projection defined in (2.1) will be the filter used in all POD closure models studied in this dissertation.

POD Lengthscale Next, we introduce the lengthscale δ used in the POD closure models. We emphasize that this choice is one of the fundamental issues in making a connection with LES. Indeed, we need such a lengthscale (δ) in order to define dimensionally sound POD models of LES flavor. To simplify the notation, in this section we utilize subscripts for the

three spatial components: x_1, x_2, x_3 and u_1, u_2, u_3 . In the rest of the dissertation, we use the standard notation x, y, z and u, v, w .

To derive the lengthscale δ , we use dimensional analysis. A dimensionally sound lengthscale $l_>$ for a turbulent pipe flow was defined in [20]. In fact, this lengthscale was only defined implicitly, through the turbulent eddy viscosity $\nu_T := u_> l_>$. Indeed, equation (22) in [20] reads

$$\nu_T := u_> l_> = \frac{\int_0^{X_2} \langle u_{i>} u_{i>} \rangle dx_2}{\left(X_2 \int_0^{X_2} \langle u_{i>,j} u_{i>,j} \rangle dx_2 \right)^{1/2}}, \quad (2.2)$$

where repeated indices denote summation, the subscript $>$ denotes unresolved POD modes,

$$\langle f \rangle = \frac{1}{L_1 L_3} \int_0^{L_1} \int_0^{L_3} f(\mathbf{x}, t) dx_1 dx_3 \quad (2.3)$$

denotes the spatial average of f in the homogeneous directions (here x_1 and x_3), and L_1, L_3 and X_2 are the streamwise, spanwise, and wall-normal dimensions of the computational domain, respectively. Note that the authors in [20] only considered the wall region, not the entire pipe flow. In (2.2), the following notation was used: $u_{i>} = \sum_{n=r+1}^N a_n^i \varphi_n$, $u_{i>} u_{i>} =$

$\sum_{i=1}^3 u_{i>} u_{i>}$, and $u_{i>,j} = \frac{\partial u_{i>}}{\partial x_j}$. Note that a quick dimensional analysis shows that the quantity defined in (2.2) has the units of a viscosity. Indeed,

$$[\nu_T] = \frac{\frac{m}{s} \frac{m}{s} m}{\left[m \left(\frac{1}{s} \frac{1}{s} m \right) \right]^{1/2}} = \frac{\frac{m^3}{s^2}}{\frac{m}{s}} = \frac{m^2}{s}. \quad (2.4)$$

In Appendix B of [20], the authors have further simplified (2.2) and expressed ν_T in terms of the first neglected POD modes:

$$\nu_T := u_> l_> = \frac{\sum_{(\mathbf{k}, n)} \lambda_{\mathbf{k}}^{(n)}}{\left(X_2 L_1 L_3 \sum_{(\mathbf{k}, n)} \lambda_{\mathbf{k}}^{(n)} \left(\int_0^{X_2} D\Phi_{i_{\mathbf{k}}}^{(n)} D\Phi_{i_{\mathbf{k}}}^{(n)*} dx_2 - k_1^2 - k_3^2 \right) \right)^{1/2}}, \quad (2.5)$$

where the triplets (\mathbf{k}, n) are the first neglected POD modes.

In equation (9.90) of [108], the authors define another dimensionally sound turbulent viscosity

$$\nu_T := u_> l_> = \frac{1}{X_2} \int_0^{X_2} \frac{\langle u_{i>} u_{i>} \rangle}{\langle u_{i>,j} u_{i>,j} \rangle^{1/2}} dx_2. \quad (2.6)$$

A quick dimensional analysis shows that the quantity defined in (2.6) also has the units of a viscosity.

We can use the two definitions of ν_T in (2.2) and (2.6) to define a lengthscale $l_{>}$. We obtain

$$l_{>} := \frac{\int_0^{X_2} \langle u_{i>} u_{i>} \rangle dx_2}{X_2 \int_0^{X_2} \langle u_{i>,j} u_{i>,j} \rangle dx_2} \quad (2.7)$$

and

$$l_{>} := \left(\frac{1}{X_2} \int_0^{X_2} \frac{\langle u_{i>} u_{i>} \rangle}{\langle u_{i>,j} u_{i>,j} \rangle} dx_2 \right)^{1/2}, \quad (2.8)$$

respectively.

In the 3D flow past a cylinder example that we consider in Chapter 5, both (2.7) and (2.8) are valid candidates for the definition of the lengthscale δ . The only modification we need to make (due to our computational domain) is to replace the horizontal averaging by spanwise averaging and take double integrals in the remaining directions. Specifically, we have

$$\delta := \left(\frac{\int_0^{L_1} \int_0^{L_2} \langle u_{i>} u_{i>} \rangle dx_1 dx_2}{\int_0^{L_1} \int_0^{L_2} \langle u_{i>,j} u_{i>,j} \rangle dx_1 dx_2} \right)^{1/2} \quad (2.9)$$

and

$$\delta := \left(\frac{1}{L_1 L_2} \int_0^{L_1} \int_0^{L_2} \frac{\langle u_{i>} u_{i>} \rangle}{\langle u_{i>,j} u_{i>,j} \rangle} dx_1 dx_2 \right)^{1/2}. \quad (2.10)$$

2.3 POD Closure Models

We are now ready to present the four POD closure models that will be investigated numerically in Chapter 5. We propose two new POD closure models: the dynamic subgrid-scale model and the variational multiscale model. These models were announced in [35]. In [224], a careful derivation and thorough numerical investigation was carried out. We also numerically test the mixing-length [20] and Smagorinsky [38, 174, 215, 223] POD closure models.

Since all four POD closure models are of EV type, we first present a general EV POD-ROM framework. Then, for each closure model, we specify its implementation in this general

framework. The general EV POD-ROM framework can be written as:

$$\dot{\mathbf{a}} = \left(\mathbf{b} + \tilde{\mathbf{b}}(\mathbf{a}) \right) + \left(\mathbf{A} + \tilde{\mathbf{A}}(\mathbf{a}) \right) \mathbf{a} + \mathbf{a}^T \mathbf{B} \mathbf{a}, \quad (2.11)$$

which is just a slight modification of the POD-G-ROM (1.13). The new terms in (2.11) (the vector $\tilde{\mathbf{b}}(\mathbf{a})$ and the matrix $\tilde{\mathbf{A}}(\mathbf{a})$) correspond to the numerical discretization of the POD closure model. In componentwise form, equation (2.11) can be written as

$$\begin{aligned} \dot{a}_i(t) &= \left(b_i + \tilde{b}_i(\mathbf{a}) \right) + \sum_{m=1}^r \left(A_{im} + \tilde{A}_{im}(\mathbf{a}) \right) a_m(t) \\ &+ \sum_{m=1}^r \sum_{n=1}^r B_{imn} a_n(t) a_m(t) \quad \text{for } i = 1, \dots, r, \end{aligned} \quad (2.12)$$

where b_i , A_{im} , and B_{imn} are the same as those in equations (1.13), $\tilde{b}_i(\mathbf{a})$ and $\tilde{A}_{im}(\mathbf{a})$ depend on the specific closure model used.

The Mixing-Length POD Reduced-Order Model (ML-POD-ROM)

The first POD closure model was the *mixing-length* model proposed in [20]. This closure model is of EV type and amounts to increasing the viscosity coefficient ν by

$$\nu_{ML} = \alpha \nu_T = \alpha U_{ML} L_{ML}, \quad (2.13)$$

where U_{ML} and L_{ML} are characteristic velocity and length scales for the unresolved scales, and α is an $\mathcal{O}(1)$ nondimensional parameter that characterizes the energy being dissipated. Using the EV ansatz in (2.13), the *mixing-length POD reduced-order model (ML-POD-ROM)* has the form (2.11), where

$$\tilde{b}_i(\mathbf{a}) = -\nu_{ML} \left(\nabla \varphi_i, \frac{\nabla \mathbf{U} + \nabla \mathbf{U}^T}{2} \right), \quad (2.14)$$

$$\tilde{A}_{im}(\mathbf{a}) = -\nu_{ML} \left(\nabla \varphi_i, \frac{\nabla \varphi_m + \nabla \varphi_m^T}{2} \right). \quad (2.15)$$

The parameter α is expected to vary in a real turbulent flow, and different values of α may result in different dynamics of the flow [20, 108, 180, 182]. There are also different ways to define ν_T in (2.13): relation (2.2) was used in [20], whereas relation (2.6) was used in [108]. We also mention that several other authors have used the ML-POD-ROM (2.14)-(2.15) (see, e.g., [35, 223]). Improvements to the ML-POD-ROM in which the EV coefficient is mode dependent were proposed in [53, 181, 194].

The Smagorinsky POD Reduced-Order Model (S-POD-ROM)

A potential improvement over the simplistic mixing-length hypothesis is to replace the constant ν_{ML} in (2.14)-(2.15) (which is computed only once, at the beginning of the simulation) with a variable turbulent viscosity (which is recomputed at every time step), such as that proposed in [211]. This yields a POD closure model in which the viscosity coefficient is increased by

$$\nu_S := 2 (C_S \delta)^2 \|\mathbb{D}(\mathbf{u}_r)\|, \quad (2.16)$$

where C_S is the Smagorinsky constant, δ is the lengthscale defined in Section 2.2 and $\|\mathbb{D}(\mathbf{u}_r)\|$ is the Frobenius norm of the deformation tensor $\mathbb{D}(\mathbf{u}_r)$. Using the EV ansatz in (2.16), the *Smagorinsky POD reduced-order model (S-POD-ROM)* has the form (2.11), where

$$\tilde{b}_i(\mathbf{a}) = -2 (C_S \delta)^2 \left(\nabla \varphi_i, \|\mathbb{D}(\mathbf{u}_r)\| \frac{\nabla \mathbf{U} + \nabla \mathbf{U}^T}{2} \right), \quad (2.17)$$

$$\tilde{A}_{im}(\mathbf{a}) = -2 (C_S \delta)^2 \left(\nabla \varphi_i, \|\mathbb{D}(\mathbf{u}_r)\| \frac{\nabla \varphi_m + \nabla \varphi_m^T}{2} \right). \quad (2.18)$$

The S-POD-ROM (2.17)-(2.18) was proposed in [35] (see also [174]) and was used in the reduced-order modeling of structurally dominated 3D turbulent flows in [215, 223]. Its advantage over the ML-POD-ROM (2.14)-(2.15) is obvious: the latter utilizes a constant EV coefficient at every time step, whereas the former recomputes the EV coefficient (which depends on $\|\mathbb{D}(\mathbf{u}_r)\|$) at every time step. To address the significant computational burden posed by the recalculation of the Smagorinsky EV coefficient at every time step, a novel two-level discretization algorithm was employed in [223] (see Chapter 5 for more details).

The Variational Multiscale POD Reduced-Order Model (VMS-POD-ROM)

The VMS method, a state-of-the-art LES closure modeling strategy, was introduced in [113, 114, 115]. The VMS method is based on the principle of *locality* of energy transfer, i.e., it uses the ansatz that energy is transferred mainly between neighboring scales. In [60], the transfer of energy among POD modes for turbulent flow past a backward-facing step (a non-homogeneous separated flow) was investigated numerically. In that report, it was shown that the Fourier-decomposition based concepts of energy cascade and locality of energy transfer

are also valid in the POD context (see Figures 3 and 4 in [60]). Thus, VMS closure models represent a natural choice for POD-ROM.

To develop the VMS POD closure model, we start by decomposing the finite set of POD modes \mathbf{X}^r into the direct sum of *large resolved* POD modes \mathbf{X}_L^r and *small resolved* POD modes \mathbf{X}_S^r :

$$\mathbf{X}^r = \mathbf{X}_L^r \oplus \mathbf{X}_S^r, \quad \text{where} \quad (2.19)$$

$$\mathbf{X}_L^r := \text{span} \{ \varphi_1, \varphi_2, \dots, \varphi_{r_L} \} \quad \text{and} \quad (2.20)$$

$$\mathbf{X}_S^r := \text{span} \{ \varphi_{r_L+1}, \varphi_{r_L+2}, \dots, \varphi_r \}. \quad (2.21)$$

Accordingly, we decompose \mathbf{u}_r into two components: \mathbf{u}_r^L representing the *large resolved* scales, and \mathbf{u}_r^S representing the *small resolved* scales:

$$\mathbf{u}_r = \mathbf{u}_r^L + \mathbf{u}_r^S, \quad (2.22)$$

where

$$\mathbf{u}_r^L = \mathbf{U} + \sum_{j=1}^{r_L} a_j \varphi_j, \quad (2.23)$$

$$\mathbf{u}_r^S = \sum_{j=r_L+1}^r a_j \varphi_j. \quad (2.24)$$

The two components \mathbf{u}_r^L and \mathbf{u}_r^S represent the projections of \mathbf{u}_r onto the two spaces \mathbf{X}_L^r and \mathbf{X}_S^r , respectively. The general POD-ROM framework (2.11) can now be separated into two equations - one for \mathbf{a}^L (the vector of POD coefficients of \mathbf{u}_r^L) and one for \mathbf{a}^S (the vector of POD coefficients of \mathbf{u}_r^S). The *Variational Multiscale POD reduced-order model (VMS-POD-ROM)* applies an eddy viscosity term to the small resolved scales only, following the principle of locality of energy transfer. The VMS-POD-ROM reads:

$$\begin{aligned} \begin{bmatrix} \dot{\mathbf{a}}^L \\ \dot{\mathbf{a}}^S \end{bmatrix} &= \begin{bmatrix} \mathbf{b}^L \\ \mathbf{b}^S \end{bmatrix} + \mathbf{A}^r \begin{bmatrix} \mathbf{a}^L \\ \mathbf{a}^S \end{bmatrix} + \begin{bmatrix} \mathbf{A}^L & \mathbf{0} \\ \mathbf{0} & \mathbf{A}^S + \tilde{\mathbf{A}}^S(\mathbf{a}^S) \end{bmatrix} \begin{bmatrix} \mathbf{a}^L \\ \mathbf{a}^S \end{bmatrix} \\ &+ \begin{bmatrix} \mathbf{a}^L \\ \mathbf{a}^S \end{bmatrix}^T \mathbf{B} \begin{bmatrix} \mathbf{a}^L \\ \mathbf{a}^S \end{bmatrix} \end{aligned} \quad (2.25)$$

The finite dimensional system (2.25) can be written componentwise as follows:

$$\dot{a}_i^L(t) = b_i^L + \sum_{m=1}^r A_{im}^r a_m(t) + \sum_{j=1}^{r_L} A_{ij}^L a_j(t) + \sum_{m=1}^r \sum_{n=1}^r B_{imn} a_n(t) a_m(t), \quad (2.26)$$

$$\forall i = 1, \dots, r_L,$$

$$\dot{a}_i^S(t) = b_i^S + \sum_{m=1}^r A_{im}^r a_m(t) + \sum_{j=r_L+1}^r \left(A_{ij}^S + \tilde{A}_{ij}^S \right) a_j(t) \quad (2.27)$$

$$+ \sum_{m=1}^r \sum_{n=1}^r B_{imn} a_n(t) a_m(t) \quad \forall i = r_L + 1, \dots, r,$$

where

$$b_i^L = -(\varphi_i, \mathbf{U} \cdot \nabla \mathbf{U}) - \frac{2}{\text{Re}} \left(\nabla \varphi_i, \frac{\nabla \mathbf{U} + \nabla \mathbf{U}^T}{2} \right), \quad (2.28)$$

$$A_{im}^r = -(\varphi_i, \mathbf{U} \cdot \nabla \varphi_m) - (\varphi_i, \varphi_m \cdot \nabla \mathbf{U}), \quad (2.29)$$

$$A_{ij}^L = -\frac{2}{\text{Re}} \left(\nabla \varphi_i, \frac{\nabla \varphi_j + \nabla \varphi_j^T}{2} \right), \quad (2.30)$$

$$B_{imn} = -(\varphi_i, \varphi_m \cdot \nabla \varphi_n), \quad (2.31)$$

$$b_i^S = -(\varphi_i, \mathbf{U} \cdot \nabla \mathbf{U}), \quad (2.32)$$

$$A_{ij}^S = -\frac{2}{\text{Re}} \left(\nabla \varphi_i, \frac{\nabla \varphi_j + \nabla \varphi_j^T}{2} \right), \quad (2.33)$$

$$\tilde{A}_{ij}^S(\mathbf{a}) = -2(C_S \delta)^2 \left(\nabla \varphi_i, \|\mathbb{D}(\mathbf{u}_r^S + \mathbf{U})\| \frac{\nabla \varphi_j + \nabla \varphi_j^T}{2} \right). \quad (2.34)$$

We emphasize that the system of equations (2.25) is *coupled* through two terms: (i) $\mathbf{a}^T \mathbf{B} \mathbf{a}$, which represents the nonlinearity $(\mathbf{u}^r \cdot \nabla) \mathbf{u}^r$; and (ii) $\mathbf{A}^r \mathbf{a}$, which represents the term $(\mathbf{u}^r \cdot \nabla) \mathbf{u}^r$ linearized around the centering trajectory \mathbf{U} . The difference between the VMS-POD-ROM (2.25)-(2.34) and the S-POD-ROM (2.17)-(2.18) is that the former acts only on the small resolved scales (since the Smagorinsky EV term $(C_S \delta)^2 \|\mathbb{D}(\mathbf{u}_r^S + \mathbf{U})\|$ is included only in the equation corresponding to \mathbf{a}^S), whereas the latter acts on all (both large and small) resolved scales.

The VMS-POD-ROM (2.25)-(2.34) was announced in [35]. A careful derivation and thorough investigation in the numerical simulation of a 3D turbulent flow was implemented in [224]. We note that a fundamentally different VMS LES closure model that utilizes the NSE residual was proposed in [26]; this model was used in a POD setting in [29]. Yet

another VMS-POD-ROM, inspired from the numerical stabilization methods developed in [98, 121, 126, 144], was proposed, analyzed and tested in [116]. We emphasize that the VMS-POD-ROM (2.25)-(2.34) is different from both the model used in [29] and that used in [116].

The Dynamic Subgrid-Scale POD Reduced-Order Model (DS-POD-ROM)

For all three POD-ROM closure models defined above (i.e., ML-POD-ROM (2.14)-(2.15), S-POD-ROM (2.17)-(2.18), and VMS-POD-ROM (2.25)-(2.34)), the definition has been entirely phenomenological. Indeed, arguing that the role of the discarded POD modes is to extract energy from the system, we used an EV ansatz to derive closure models of increasing complexity and physical accuracy. The DS POD-ROM closure model is also of EV type. Its derivation, however, requires a precise definition of the filtering operation. The DS closure model has its origins in LES, where it is considered state-of-the-art (see, e.g., [199]). In LES, the filtering operation is effected by convolving the flow variables with a rapidly decaying spatial filter. In POD, the filtering operation is effected by using the POD Galerkin projection described in Section 2.2 (see (2.1)). To derive the precise POD filtered equations, we start with the NSE (1.10) in which the velocity \mathbf{u} is replaced by its POD approximation $\mathbf{u}(\mathbf{x}, t) \approx \mathbf{u}_r(\mathbf{x}, t) \equiv \mathbf{U}(\mathbf{x}) + \sum_{j=1}^r a_j(t)\boldsymbol{\varphi}_j(\mathbf{x})$ in (1.11), and obtain

$$\frac{\partial \mathbf{u}_r}{\partial t} - \text{Re}^{-1} \Delta \mathbf{u}_r + (\mathbf{u}_r \cdot \nabla) \mathbf{u}_r + \nabla p = 0. \quad (2.35)$$

Using the fact that $\nabla \cdot \mathbf{u}_r = 0$ in (2.35), we get $(\mathbf{u}_r \cdot \nabla) \mathbf{u}_r = \nabla \cdot (\mathbf{u}_r \mathbf{u}_r)$. Thus, (2.35) can be rewritten as

$$\frac{\partial \mathbf{u}_r}{\partial t} - \text{Re}^{-1} \Delta \mathbf{u}_r + \nabla \cdot (\mathbf{u}_r \mathbf{u}_r) + \nabla p = 0. \quad (2.36)$$

Applying the POD filtering operation (2.1) to (2.36), using the fact that the POD Galerkin projection is a linear operator, and *assuming* that differentiation and POD filtering commute, we obtain

$$\frac{\partial \bar{\mathbf{u}}_r}{\partial t} - \text{Re}^{-1} \Delta \bar{\mathbf{u}}_r + \nabla \cdot (\bar{\mathbf{u}}_r \bar{\mathbf{u}}_r) + \nabla \bar{p} = 0. \quad (2.37)$$

We note that, if filtering and differentiation do not commute, one has to estimate the commutation error (see, e.g., [32, 217, 218]). We also note that, since the POD filtering operation is

the Galerkin projection (2.1), $\bar{\mathbf{u}}_r = \mathbf{u}_r$. To maintain a notation consistent with the nonlinear term, we continue to use the $\bar{\mathbf{u}}_r$ notation in the following development.

The POD filtered equation (2.37) can be rewritten as

$$\frac{\partial \bar{\mathbf{u}}_r}{\partial t} - \text{Re}^{-1} \Delta \bar{\mathbf{u}}_r + \nabla \cdot (\bar{\mathbf{u}}_r \bar{\mathbf{u}}_r) + \nabla \cdot (\boldsymbol{\tau}_r) + \nabla \bar{p} = 0, \quad (2.38)$$

where

$$\boldsymbol{\tau}_r = \overline{\mathbf{u}_r \mathbf{u}_r} - \bar{\mathbf{u}}_r \bar{\mathbf{u}}_r \quad (2.39)$$

is the POD subfilter-scale stress tensor. Thus, the POD-G-ROM (1.12) amounts to setting $\boldsymbol{\tau}_r = 0$. For turbulent flows, as we have already mentioned, this approximation is flawed. Thus, one needs to address the POD closure problem, i.e., to model the POD subfilter-scale stress tensor $\boldsymbol{\tau}_r$ in terms of the POD filtered velocity $\bar{\mathbf{u}}_r$. We note that the POD closure problem is exactly the LES closure problem, in which the spatial filtering is replaced by POD Galerkin projection. For all three POD-ROM closure models defined so far in this section (i.e., ML-POD-ROM (2.14)-(2.15), S-POD-ROM (2.17)-(2.18), and VMS-POD-ROM (2.25)-(2.34)), the closure problem has been addressed by assuming an EV ansatz for $\boldsymbol{\tau}_r$. The DS-POD-ROM employs an EV ansatz as well; specifically, the Smagorinsky model is utilized:

$$\boldsymbol{\tau}_r := -2(C_S \delta)^2 \|\mathbb{D}(\bar{\mathbf{u}}_r)\| \mathbb{D}(\bar{\mathbf{u}}_r), \quad (2.40)$$

in which C_S is not a constant (as in the Smagorinsky model), but a function of space and time, i.e., $C_S = C_S(\mathbf{x}, t)$. To compute $C_S(\mathbf{x}, t)$, we follow the LES derivation in [199] and replace the LES spatial filtering with the POD Galerkin projection. Since there are two spatial filters in the LES derivation of the DS model, we define a second POD Galerkin projection (in addition to that defined in (2.1)): For all $\mathbf{u} \in \mathbf{X}$, the second (test) Galerkin projection $\tilde{\mathbf{u}} \in \mathbf{X}^R$ (where $R < r$) is the solution of the following equation:

$$(\mathbf{u} - \tilde{\mathbf{u}}, \boldsymbol{\varphi}) = 0 \quad \forall \boldsymbol{\varphi} \in \mathbf{X}^R. \quad (2.41)$$

Applying the second POD filtering operation (2.41) to (2.37), we obtain:

$$\frac{\partial \tilde{\mathbf{u}}_r}{\partial t} - \text{Re}^{-1} \Delta \tilde{\mathbf{u}}_r + \nabla \cdot (\tilde{\mathbf{u}}_r \tilde{\mathbf{u}}_r) + \nabla \cdot (\mathbf{T}_r) + \nabla \tilde{p} = 0, \quad (2.42)$$

where

$$\mathbf{T}_r = \widetilde{\overline{\mathbf{u}_r \mathbf{u}_r}} - \widetilde{\mathbf{u}_r} \widetilde{\mathbf{u}_r} \quad (2.43)$$

is the second POD subfilter-scale stress tensor. We note that the following identity (known as the ‘‘Germano identity’’ in LES) holds:

$$\mathbf{T}_r = \widetilde{\overline{\mathbf{u}_r \mathbf{u}_r}} - \widetilde{\mathbf{u}_r} \widetilde{\mathbf{u}_r} = \left(\widetilde{\overline{\mathbf{u}_r \mathbf{u}_r}} - \widetilde{\mathbf{u}_r} \widetilde{\mathbf{u}_r} \right) + \left(\widetilde{\overline{\mathbf{u}_r \mathbf{u}_r}} - \widetilde{\overline{\mathbf{u}_r \mathbf{u}_r}} \right) = \mathbf{L}_r + \widetilde{\boldsymbol{\tau}}_r, \quad (2.44)$$

where $\mathbf{L}_r = \widetilde{\overline{\mathbf{u}_r \mathbf{u}_r}} - \widetilde{\mathbf{u}_r} \widetilde{\mathbf{u}_r}$ and $\widetilde{\boldsymbol{\tau}}_r = \widetilde{\overline{\mathbf{u}_r \mathbf{u}_r}} - \widetilde{\overline{\mathbf{u}_r \mathbf{u}_r}}$. We assume the same EV ansatz for the two POD subfilter-scale stress tensors, $\boldsymbol{\tau}_r$ and \mathbf{T}_r :

$$\mathbf{T}_r \approx -2 (C_S \delta)^2 \|\mathbb{D}(\widetilde{\mathbf{u}_r})\| \mathbb{D}(\widetilde{\mathbf{u}_r}) \quad (2.45)$$

$$\boldsymbol{\tau}_r \approx -2 (C_S \delta)^2 \|\mathbb{D}(\overline{\mathbf{u}_r})\| \mathbb{D}(\overline{\mathbf{u}_r}), \quad (2.46)$$

where δ is the filter radius used in the second POD filtering operation (2.41). Assuming that C_S remains constant under the second POD filtering (2.41), we obtain:

$$\widetilde{\boldsymbol{\tau}}_r \approx -2 (C_S \delta)^2 \|\widetilde{\mathbb{D}(\overline{\mathbf{u}_r})}\| \widetilde{\mathbb{D}(\overline{\mathbf{u}_r})} \approx -2 (C_S \delta)^2 \|\mathbb{D}(\overline{\mathbf{u}_r})\| \mathbb{D}(\overline{\mathbf{u}_r}). \quad (2.47)$$

Substituting (2.45) and (2.47) into (2.44) we obtain:

$$-2 (C_S \delta)^2 \|\mathbb{D}(\widetilde{\mathbf{u}_r})\| \mathbb{D}(\widetilde{\mathbf{u}_r}) = \left(\widetilde{\overline{\mathbf{u}_r \mathbf{u}_r}} - \widetilde{\mathbf{u}_r} \widetilde{\mathbf{u}_r} \right) - 2 (C_S \delta)^2 \|\widetilde{\mathbb{D}(\overline{\mathbf{u}_r})}\| \widetilde{\mathbb{D}(\overline{\mathbf{u}_r})}. \quad (2.48)$$

We note that C_S is the only unknown in (2.48), all other terms being computable quantities. Since all the terms in (2.48) are tensors, the unknown C_S cannot satisfy all nine equations. Thus, the following least squares approach is considered instead:

$$\begin{aligned} \min_{C_S^2} & \left[\left(\widetilde{\overline{\mathbf{u}_r \mathbf{u}_r}} - \widetilde{\mathbf{u}_r} \widetilde{\mathbf{u}_r} \right) - 2 (C_S \delta)^2 \|\widetilde{\mathbb{D}(\overline{\mathbf{u}_r})}\| \widetilde{\mathbb{D}(\overline{\mathbf{u}_r})} + 2 (C_S \delta)^2 \|\mathbb{D}(\widetilde{\mathbf{u}_r})\| \mathbb{D}(\widetilde{\mathbf{u}_r}) \right] : \\ & \left[\left(\widetilde{\overline{\mathbf{u}_r \mathbf{u}_r}} - \widetilde{\mathbf{u}_r} \widetilde{\mathbf{u}_r} \right) - 2 (C_S \delta)^2 \|\widetilde{\mathbb{D}(\overline{\mathbf{u}_r})}\| \widetilde{\mathbb{D}(\overline{\mathbf{u}_r})} + 2 (C_S \delta)^2 \|\mathbb{D}(\widetilde{\mathbf{u}_r})\| \mathbb{D}(\widetilde{\mathbf{u}_r}) \right]. \end{aligned} \quad (2.49)$$

The solution $C_S(\mathbf{x}, t)$ of (2.49) is:

$$C_S^2(\mathbf{x}, t) = \frac{\left[\widetilde{\overline{\mathbf{u}_r \mathbf{u}_r}} - \widetilde{\mathbf{u}_r} \widetilde{\mathbf{u}_r} \right] : \left[2 \delta^2 \|\widetilde{\mathbb{D}(\overline{\mathbf{u}_r})}\| \widetilde{\mathbb{D}(\overline{\mathbf{u}_r})} - 2 \delta^2 \|\mathbb{D}(\widetilde{\mathbf{u}_r})\| \mathbb{D}(\widetilde{\mathbf{u}_r}) \right]}{\left[2 \delta^2 \|\widetilde{\mathbb{D}(\overline{\mathbf{u}_r})}\| \widetilde{\mathbb{D}(\overline{\mathbf{u}_r})} - 2 \delta^2 \|\mathbb{D}(\widetilde{\mathbf{u}_r})\| \mathbb{D}(\widetilde{\mathbf{u}_r}) \right] : \left[2 \delta^2 \|\widetilde{\mathbb{D}(\overline{\mathbf{u}_r})}\| \widetilde{\mathbb{D}(\overline{\mathbf{u}_r})} - 2 \delta^2 \|\mathbb{D}(\widetilde{\mathbf{u}_r})\| \mathbb{D}(\widetilde{\mathbf{u}_r}) \right]}. \quad (2.50)$$

Since the stress tensors can be computed directly from the resolved field, (2.50) yields a time- and space-dependent formula for $C_S(\mathbf{x}, t)$.

Thus, the DS-POD-ROM increases the viscosity coefficient by

$$\nu_{DS} := 2 (C_S(\mathbf{x}, t) \delta)^2 \|\mathbb{D}(\mathbf{u}_r)\|, \quad (2.51)$$

where $C_S(\mathbf{x}, t)$ is the coefficient in (2.50), δ is the lengthscale defined in Section 2.2 and $\|\mathbb{D}(\mathbf{u}_r)\|$ the Frobenius norm of the deformation tensor $\mathbb{D}(\mathbf{u}_r)$. Thus, the *Dynamic Subgrid-Scale POD reduced-order model (DS-POD-ROM)* has the form (2.11), where

$$\tilde{b}_i(\mathbf{a}) = -2 \delta^2 \left(\nabla \varphi_i, C_S^2(\mathbf{x}, t) \|\mathbb{D}(\mathbf{u}_r)\| \frac{\nabla \mathbf{U} + \nabla \mathbf{U}^T}{2} \right), \quad (2.52)$$

$$\tilde{A}_{im}(\mathbf{a}) = -2 \delta^2 \left(\nabla \varphi_i, C_S^2(\mathbf{x}, t) \|\mathbb{D}(\mathbf{u}_r)\| \frac{\nabla \varphi_m + \nabla \varphi_m^T}{2} \right). \quad (2.53)$$

Note that ν_{DS} defined in (2.51) can take negative values. This can be interpreted as *backscatter*, the inverse transfer of energy from high index POD modes to low index modes. The notion of backscatter, well-established in LES (see, e.g., [199]), was also found in a POD setting in the numerical investigation in [60].

2.4 Summary

This chapter put forth two novel POD-ROMs (the DS-POD-ROM and the VMS-POD-ROM), which are inspired from state-of-the-art LES closure modeling strategies. Both new closure models employ the concept of energy cascade and introduce extra eddy viscosity in the reduced-order system to close the POD-ROM. These two new POD-ROMs together with the ML-POD-ROM and the S-POD-ROM will be tested in the numerical simulation of a 3D turbulent flow past a cylinder at $\text{Re} = 1000$ in Chapter 5.

Chapter 3

Two-Level Algorithms for Nonlinear Closure Models [†]

3.1 Introduction

Despite its success as a general reduced-order modeling tool for complex systems, POD has yet to fulfill its original promise of computationally efficient, yet accurate approximation of coherent structures in high Reynolds number turbulent flows, as put forth in the pioneering work [20]. Indeed, in this context, the ultimate goal of POD-ROM is to capture the coherent structures in turbulent flows at high Reynolds numbers for time intervals that are much longer than those over which the snapshots (i.e., the flow field data at different time instants) were collected and at parameter values (e.g., Reynolds number) that are different from those at which the snapshots were generated. To our knowledge, this goal has not yet been achieved. We believe that one of the main reasons is the following. To balance the low computational cost required by a ROM and the complexity of the targeted turbulent flows, appropriate closure modeling strategies need to be employed. Since modern closure models for turbulent flows are generally nonlinear, their efficient numerical discretization within a POD framework is challenging. Indeed, most POD-ROMs, employ precomputed matrices and tensors, which

[†]The material in this chapter is based on the following report: Z. Wang, I. Akhtar, J. Borggaard and T. Iliescu. Two-Level Discretizations of Nonlinear Closure Models for Proper Orthogonal Decomposition. *J. Comput. Phys.*, 230(1), 2011, pp. 126-146.

precludes a straightforward discretization of nonlinear closure model terms. This chapter aims at developing a computationally efficient numerical discretization of general, nonlinear closure modeling strategies for POD-ROMs. Before describing this new methodology in detail, we feel that the following remarks regarding the very scope of this chapter are needed.

We start by emphasizing that, in this chapter, we do *not* answer and do *not* try to answer the question of whether or not POD-ROM can be employed to simulate dominant coherent structures in high Reynolds number turbulent flows. We also do *not* find the best closure model for POD-ROM and we do *not* assess the robustness of the POD-ROM with respect to changes in the Reynolds number or length of simulation time. Instead, we try to elucidate whether nonlinear closure modeling strategies can be utilized efficiently in a POD context. We believe that this is a *necessary* step in answering the above open question. Indeed, we believe that without employing modern closure modeling strategies, which are generally nonlinear, POD-ROM does not stand any chance in successfully modeling dominant coherent structures in high Reynolds number turbulent flows. We also emphasize that, although all numerical tests in this chapter involve fluid flow problems, the two-level methodology we propose can be applied to *any nonlinear* closure model and *any* application in which POD is used as a reduced-order modeling technique.

We now present the two-level methodology for the numerical discretization of nonlinear closure modeling in POD-ROMs. One of the main computational hurdles in the development of sophisticated closure models for POD-ROMs of coherent structures in turbulent flows has been the lack of efficient computational strategies for the discretization of *nonlinear* closure modeling terms. Indeed, the main advantage of the ROM (1.13) over the usual discretizations of (1.10) (such as finite elements or finite differences) is that the dimension of the dynamical system (1.13) is usually extremely small. Thus, the time integration of (1.13) can be effected very efficiently. There is, however, a caveat. *The matrices in (1.13) are computed only once at the beginning of the simulation.* This is essential for the computational efficiency of the ROM. Indeed, if we had to recompute the matrices at every time step, this would increase the computational time dramatically, since these matrices are *full* and the POD basis functions $\{\varphi_1, \dots, \varphi_r\}$ are *global*. Thus, current POD closure models use matrices that are *precomputed*. *This restriction has been a major computational hurdle in the development*

of modern closure strategies for POD of turbulent flows dominated by coherent structures.

Many of these closure modeling strategies introduce nonlinear, non-polynomial terms that need to be efficiently approximated. Existing methods to treat terms of this form (see Section 1.5 for an overview of these alternative approaches) are interpolatory and thus not directly amenable to the chaotic nature of turbulent flows.

In this chapter, we propose a novel way to overcome this major computational hurdle. We emphasize that this chapter is *not* concerned with the derivation of improved POD closure models ([35, 224]). Instead, for a given closure model (one of the most basic LES closure models), we develop a new algorithm that allows an efficient and accurate numerical discretization of that POD closure model. Our new approach is based on the observation that only a small subset of the original POD basis is used in the ROM and this basis is typically dominated by coarse, large scale structures. Thus, the numerical discretization of the nonlinear closure term can be effected on a mesh at a resolution that is significantly lower than that of the mesh utilized in the derivation of the full POD basis. Based on the above observation, we develop a two-level algorithm for the discretization of the nonlinear closure model. This two-level algorithm allows for an efficient computation of the nonlinear POD closure model at each time step without compromising its accuracy. Numerical results supporting our new approach are presented for three test problems.

The rest of the chapter is organized as follows. In section 3.2, we describe the new two-level algorithms. Section 3.3 presents results using these new algorithms on three test problems: the 1D Burgers equation with a very small viscosity parameter ($\nu = 10^{-3}$), a 2D flow past a cylinder at a Reynolds number $Re = 200$, and a 3D turbulent flow past a cylinder at a moderate Reynolds number $Re = 1000$. Conclusions are presented in Section 3.4.

3.2 Two-Level Algorithms

In this section, we describe new two-level algorithms for an efficient and accurate discretization of nonlinear POD closure models.

We start by describing the standard algorithm used in the numerical discretization of the POD-G-ROM (1.13), written in componentwise in (1.15). The *POD Galerkin (POD-G)*

algorithm, a numerical discretization of POD-G-ROM, reads

$\begin{aligned} & \ell = 0; \text{ compute } \mathbf{b}, \mathbf{A}, \mathbf{B} ; \\ & \text{for } \ell = 0 \text{ to } M - 1 \\ & \quad \mathbf{a}^{\ell+1} := F(\mathbf{a}^\ell); \\ & \text{endfor} \end{aligned} \tag{3.1}$	POD-G algorithm
--	--------------------

where the superscript ℓ denotes the current time step, M is the total number of time steps, and F is the function corresponding to the time-advancement method (e.g., Euler's method).

Closure modeling is needed when using the POD-G-ROM (1.13) for turbulent flows (see 2). One of the most popular closure models used in LES of turbulent flows is the Smagorinsky model [211], which augments the diffusion coefficient by $2(C_S\delta)^2\|\mathbb{D}(\mathbf{u}_r)\|$ in (2.16). This yields the S-POD-ROM, which has the form (2.11) with closure terms in componentwise form, \tilde{b}_k and \tilde{A}_{km} given in (2.17) and (2.18).

We emphasize that we are concerned with closure models (such as S-POD-ROM) that *cannot* be cast as a quadratic nonlinearity and thus cannot be simply included in \mathbf{B} . The brute force approach in the numerical discretization of the general nonlinear closure model (2.11) yields the *one-level algorithm*:

$\begin{aligned} & \ell = 0; \text{ compute } \mathbf{b}, \mathbf{A}, \mathbf{B} \text{ on the } \underline{\text{fine}} \text{ mesh} ; \\ & \text{for } \ell = 0 \text{ to } M - 1 \\ & \quad \text{compute } \tilde{\mathbf{b}}(\mathbf{a}^\ell), \tilde{\mathbf{A}}(\mathbf{a}^\ell) \text{ on the } \underline{\text{fine}} \text{ mesh} \tag{3.2} \\ & \quad \mathbf{a}^{\ell+1} := \tilde{F}(\mathbf{a}^\ell); \\ & \text{endfor} \end{aligned}$	one-level algorithm
--	------------------------

where \tilde{F} is a modification of F in (3.1) accounting for the additional nonlinear closure terms. Because we need to recompute (reassemble) $\tilde{\mathbf{b}}(\mathbf{a})$ and $\tilde{\mathbf{A}}(\mathbf{a})$ at each time step, algorithm (3.2) is impractical. Indeed, this algorithm removes a major advantage of ROM — the decrease of

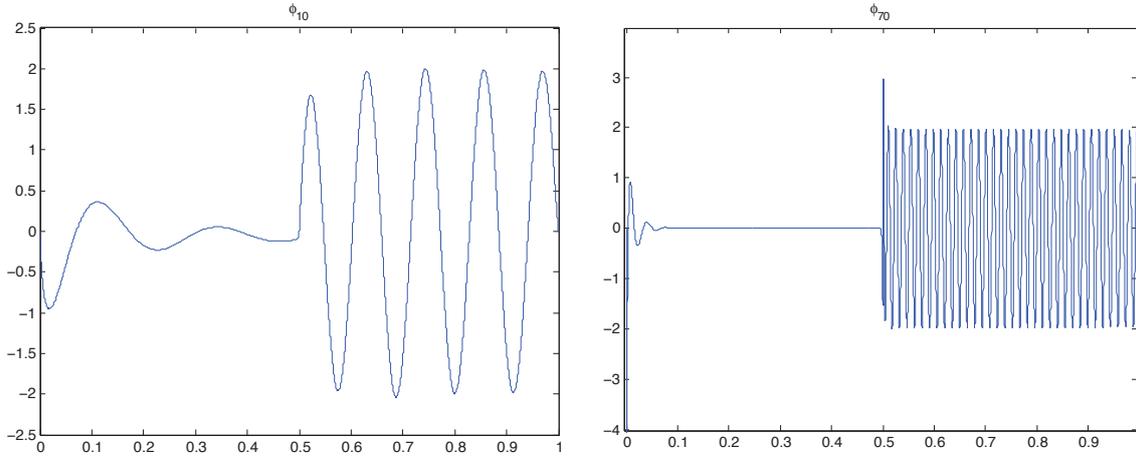


Figure 3.1: 1D Burgers equation with $\nu = 10^{-3}$. POD basis functions: (left) φ_{10} ; (right) φ_{70} .

computational time. This is because, although the number of retained POD bases r is much smaller than the rank of snapshots d , the POD basis functions are global and reassembling $\tilde{\mathbf{b}}(\mathbf{a})$ and $\tilde{\mathbf{A}}(\mathbf{a})$ at each time step dramatically increases the CPU time.

To make algorithm (3.2) practical, we must avoid the computational price required by the matrix reassembly. The two-level algorithms we propose are based on the observation that only a small subset of the original POD basis is used in the ROM. Indeed, if a computation on a fine mesh of size h is employed to obtain the POD basis $\{\varphi_1, \dots, \varphi_d\}$, one is usually able to use a coarser mesh of size H to represent the POD basis $\{\varphi_1, \dots, \varphi_r\}$, where $r \ll d$. Figure 3.1 clearly supports this statement. Indeed, the numerical resolution needed for an accurate approximation of φ_{10} is *much lower* than that needed for the approximation of φ_{70} . We also note that a similar conclusion would be drawn if we used Fourier basis functions [108].

Thus, to avoid the high computational cost required by the one-level algorithm (3.2), we propose an algorithm that computes $\tilde{\mathbf{b}}(\mathbf{a})$ and $\tilde{\mathbf{A}}(\mathbf{a})$ at each time step on a *coarse* mesh of size H , and not on the same fine mesh of size h used in the computation of the POD basis functions $\{\varphi_1, \dots, \varphi_d\}$. The resulting algorithm, which we call the *hybrid two-level algorithm*, is the following:

```

 $\ell = 0$ ; compute  $\mathbf{b}, \mathbf{A}, \mathbf{B}$  on the fine mesh ;
for  $\ell = 0$  to  $M - 1$ 
  compute  $\tilde{\mathbf{b}}(\mathbf{a}^\ell), \tilde{\mathbf{A}}(\mathbf{a}^\ell)$  on the coarse mesh (3.3)
   $\mathbf{a}^{\ell+1} := \tilde{F}(\mathbf{a}^\ell)$ ;
endfor

```

hybrid
two-level
algorithm

The advantage of the two-level algorithm (3.3) over the one-level algorithm (3.2) is obvious: the former achieves *the same level of accuracy* as the latter while decreasing the computational cost by an *order of magnitude*.

$$\begin{aligned} \text{error}_{two-level} &\sim \text{error}_{one-level} \\ \text{CPU}_{two-level} &\ll \text{CPU}_{one-level} \end{aligned}$$

The dramatic reduction in CPU time for the two-level algorithm is due to the efficient computation of $\tilde{\mathbf{b}}(\mathbf{a})$ and $\tilde{\mathbf{A}}(\mathbf{a})$ on the *coarse* mesh at *each* time step.

For completeness, we also investigate a modification of the *hybrid two-level algorithm* (3.3). This algorithm is denoted as the *coarse two-level algorithm*:

```

 $\ell = 0$ ; compute  $\mathbf{b}, \mathbf{A}, \mathbf{B}$  on the coarse mesh ;
for  $\ell = 0$  to  $M - 1$ 
  compute  $\tilde{\mathbf{b}}(\mathbf{a}^\ell), \tilde{\mathbf{A}}(\mathbf{a}^\ell)$  on the coarse mesh (3.4)
   $\mathbf{a}^{\ell+1} := \tilde{F}(\mathbf{a}^\ell)$ ;
endfor

```

coarse
two-level
algorithm

Note that the coarse two-level algorithm (3.4) is actually the one-level algorithm (3.2) applied to POD bases that have been interpolated on a coarse mesh. Thus, once the fine grid is used

in the generation of the snapshots, it is no longer used in the coarse two-level algorithm (3.4).

A central role in our new two-level POD-ROM algorithm is played by the coarsening and interpolation procedures. Since this study represents the first investigation of the new two-level algorithm, we chose the *simplest* approach — uniformly skipping mesh points to coarsen the structured mesh and interpolate on the coarse mesh. We emphasize, however, that there are obvious ways to improve the coarsening and interpolation procedures. We plan to pursue these approaches in the future (see Section 6.4).

We also note that the two-level methodology introduced here is different from modeling approaches in which the POD computed on a coarse mesh is used as a surrogate for the entire underlying system (see, e.g., [70]). Indeed, the POD basis in our two-level methodology is computed on the fine mesh. After extracting the POD basis from the fine mesh computation, we interpolate it on the coarse mesh and compute the POD closure model. Employing a POD basis computed directly on the coarse mesh yields highly inaccurate results for the complex flows considered in this chapter.

Next, we present numerical experiments that test the improvement of the two-level algorithms (3.3) and (3.4) over the one-level algorithm (3.2).

3.3 Numerical Results

In this section, we investigate the two-level algorithms in the numerical simulation of three test problems: (i) the 1D Burgers equation with a small diffusion coefficient (section 3.3.1); (ii) a 2D flow past a circular cylinder at $Re = 200$ (section 3.3.2); and (iii) a 3D flow past a circular cylinder at $Re = 1000$ (section 3.3.3). Both two-level algorithms (3.3) and (3.4) employ *two meshes*: (i) a *fine mesh* of size h (the mesh on which the POD basis $\{\varphi_1, \dots, \varphi_d\}$ was computed); and (ii) a *coarse mesh* of size H (the mesh on which the computation of the POD closure model is effected at each time step). For a given fine mesh of size h , we consider different coarse meshes with sizes H . The coarsening factor R_c represents the ratio H/h . We investigate whether the coarse two-level algorithm (3.4) and the hybrid two-level algorithm (3.3) can achieve the same order of accuracy as the standard one-level algorithm

(3.2), while significantly decreasing the CPU time. The accuracy of the three algorithms is benchmarked against a DNS. The relative error in the algorithms is computed as

$$error = \frac{\frac{1}{M} \sum_{\ell=1}^M \|\mathbf{u}^{POD-ROM}(\mathbf{x}, t_\ell) - \mathbf{u}^{DNS}(\mathbf{x}, t_\ell)\|_0^2}{\frac{1}{M} \sum_{\ell=1}^M \|\mathbf{u}^{DNS}(\mathbf{x}, t_\ell)\|_0^2}, \quad (3.5)$$

where M is the number of snapshots and $\|\cdot\|_0$ is the L^2 norm. To ensure the fairness of our numerical investigation, we also performed numerical experiments on the POD-G-ROM (1.13) (i.e., without any closure model) with a small number of POD basis functions (small r). Poor results with POD-G-ROM indicate the need for the closure modeling employed in the two-level and one-level algorithms. We also mention that, for both the one-level and the two-level algorithms, we only recompute $\tilde{\mathbf{b}}$ and $\tilde{\mathbf{A}}$ every one hundred time steps. The reason is that, had we computed $\tilde{\mathbf{b}}$ and $\tilde{\mathbf{A}}$ at every time step for the one-level algorithm, this would have increased the CPU time for the one-level algorithm beyond our present computational limits. We emphasize that these computational savings were achieved *without* sacrificing the numerical accuracy. Indeed, the same qualitative results (on a shorter time interval) were achieved by evaluating $\tilde{\mathbf{b}}$ and $\tilde{\mathbf{A}}$ at every time step. This is not surprising, since we used a very small time step in the time discretization. In practical computations with the two-level algorithms (when the one-level algorithm does not need to be used) $\tilde{\mathbf{b}}$ and $\tilde{\mathbf{A}}$ will be evaluated at every time step. Finally, we mention that in all the models tested (both POD-G-ROM and S-POD-ROM), the time discretization was effected by using the explicit Euler method with a small time-step. Since the main focus of this chapter is the efficient spatial discretization for POD closure modeling, we chose this simple time discretization method; using the same time-discretization for both POD-G-ROM and S-POD-ROM allowed for a fair comparison. Obviously, higher-order methods can (and should) be used for further computational savings. We plan to pursue this research direction in the future (see Section 6.4).

3.3.1 1D Burgers Equation with $\nu = 10^{-3}$

In this section, we investigate the two-level algorithms in the numerical simulation of the 1D *Burgers equation* with a small diffusion coefficient $\nu = 10^{-3}$:

$$u_t - \nu u_{xx} + u u_x = f. \quad (3.6)$$

We use a computational setting that is similar to that used by Kunisch and Volkwein in [136]. The initial condition on domain $\Omega = [0, 1]$ is defined as $u_0(x) = 1$ if $x \in (0, 1/2]$ and $u_0(x) = 0$ otherwise. The forcing term is $f = 0$ and the time interval is $[0, T] = [0, 1]$. The boundary conditions are homogeneous Dirichlet.

A DNS is used as benchmark. This is computed using the finite element method with piecewise linear polynomials to discretize the problem in space, the implicit Euler method to discretize in time, and Newton's method to solve the nonlinear system. We use a mesh-size $h = 1/8192$ and a time step $\Delta t = 10^{-3}$. A mesh refinement study indicates that the DNS convergence has been achieved. The CPU time for DNS is 6155 s.

We use the DNS data consisting of 1001 snapshots to generate the POD basis. This POD basis is then used in all POD-ROMs that we investigate next.

The qualitative behavior of all the models tested is represented by the time evolution of the POD coefficients $a_1(\cdot)$ and $a_6(\cdot)$ plotted in Figures 3.3-3.4. We note that the other POD coefficients have a similar behavior; for clarity of exposition, we included only $a_1(\cdot)$ and $a_6(\cdot)$. For comparison purposes, Figures 3.3-3.4 include the time evolution of the POD coefficients obtained by projecting the DNS onto each POD mode.

The POD-G-ROM (1.13) with low r ($r = 10$) performs poorly, although it is computationally efficient (CPU time of 19 s). Indeed, the temporal evolutions of POD coefficients in Figures 3.3-3.4 are inaccurate. The amplitude of the temporal evolution of the POD coefficient $a_6(\cdot)$ for POD-G-ROM with $r = 10$ is *two times larger* than that for DNS. The relative error of POD-G with $r = 10$ is 7.54×10^{-2} . Thus, although the first 10 POD modes capture most (99.44%) of the system's energy, an additional term to stabilize the computations is needed [38].

We now consider the S-POD-ROM (2.16) with an artificial viscosity coefficient $C = 7 \times 10^{-4}$ as a closure model. Although the Smagorinsky model targets 3D turbulent flows, for this test problem it plays the role of numerical stabilization. For the numerical discretization of this nonlinear POD-ROM, we will use three different strategies: (i) the one-level algorithm (3.2); (ii) the hybrid two-level algorithm (3.3); and (iii) the coarse two-level algorithm (3.4). In comparing the three algorithms, the time discretization was effected by using the explicit Euler method with $\Delta t = 10^{-5}$.

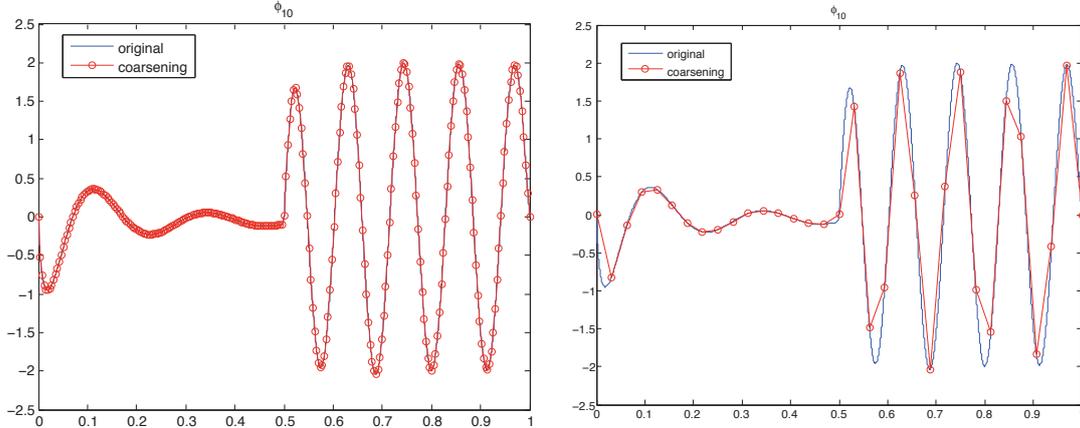


Figure 3.2: 1D Burgers equation with $\nu = 10^{-3}$. Coarsening and interpolation of φ_{10} for two rates of coarsening: 32 (left) and 256 (right).

The one-level algorithm (3.2) with $r = 10$ yields the temporal evolution of POD coefficients in Figures 3.3-3.4. Comparing these results with those for the DNS projection clearly shows that the one-level algorithm yields relatively accurate results although it uses only a few ($r = 10$) POD basis functions. Indeed, the one-level algorithm (3.2) produces results that are much closer to those for the DNS than the results for POD-G-ROM with $r = 10$. The relative error of the one-level algorithm is 2.55×10^{-2} , whereas that of POD-G-ROM with $r = 10$ is 7.54×10^{-2} . This, however, comes at a high computational cost. The CPU time is 19 s for POD-G-ROM and is 858 s for the one-level algorithm.

The coarsening and interpolation procedures used in the two-level algorithms (3.3) and (3.4) were straightforward: The coarsening of the mesh was effected by skipping nodes and the interpolation of the POD basis function φ_j on the coarse mesh was carried out by simply keeping the values of φ_j corresponding to the nodes of the coarse mesh. The results in Figure 3.2 indicate that the coarsening and interpolation preserve the structure of the POD basis remarkably well.

The *coarse two-level algorithm* (3.4) with $r = 10$ and $R_c = 16, 32$ produces the temporal evolution of POD coefficients in Figures 3.3-3.4. Comparing these results with those for the one-level algorithm (3.2), we conclude that the coarse two-level algorithm (3.4) and the one-level algorithm (3.2) yield results that are qualitatively similar (see Table 3.1).

The *hybrid two-level algorithm* (3.3) with $r = 10$ and $R_c = 16, 32$ yields the temporal

evolution of POD coefficients in Figures 3.3-3.4. Comparing these results with those for the one-level algorithm (3.2), we conclude that the hybrid two-level algorithm (3.3) yields an improved accuracy while keeping the CPU time at a modest level (see Table 3.1).

To measure the efficiency of our two-level algorithms, we define a *speed-up factor*

$$S_f \equiv \frac{\text{CPU time of one-level algorithm (3.2)}}{\text{CPU time of two-level algorithm (3.4) or (3.3)}}. \quad (3.7)$$

We present the speed-up factors along with relative errors for both two-level algorithms in Table 3.1. The relative error of the one-level algorithm is 2.55×10^{-2} . The data in Table 3.1 clearly shows that both two-level algorithms significantly decrease the CPU time (by up to an *order of magnitude*) while maintaining similar accuracy as the more expensive one-level algorithm. We also emphasize that, as expected, the two-level algorithms decrease significantly the CPU time of the underlying DNS, by a factor of more than 300.

R_c	coarse two-level		hybrid two-level	
	S_f	error	S_f	error
2	1.92	2.43×10^{-2}	1.87	2.43×10^{-2}
4	3.68	2.36×10^{-2}	3.59	2.36×10^{-2}
8	6.76	2.31×10^{-2}	6.60	2.33×10^{-2}
16	12.62	2.27×10^{-2}	12.08	2.30×10^{-2}
32	21.45	2.20×10^{-2}	19.50	2.27×10^{-2}

Table 3.1: 1D Burgers equation with $\nu = 10^{-3}$. Speed-up factors S_f and relative errors for different coarsening factors R_c .

Thus, for this test problem, we conclude that the two-level algorithms achieve the same order of accuracy as the standard one-level algorithm, while decreasing the CPU time by more than *an order of magnitude*. We note that, for this test case, the coarse two-level algorithm (3.4) performs better than the hybrid two-level algorithm (3.3).

3.3.2 2D Flow Past a Circular Cylinder at $\text{Re} = 200$

In this section, we investigate the two-level algorithm using the numerical simulation of 2D flow past a circular cylinder at $\text{Re} = 200$. We simulate 2D flow past a cylinder using a parallel

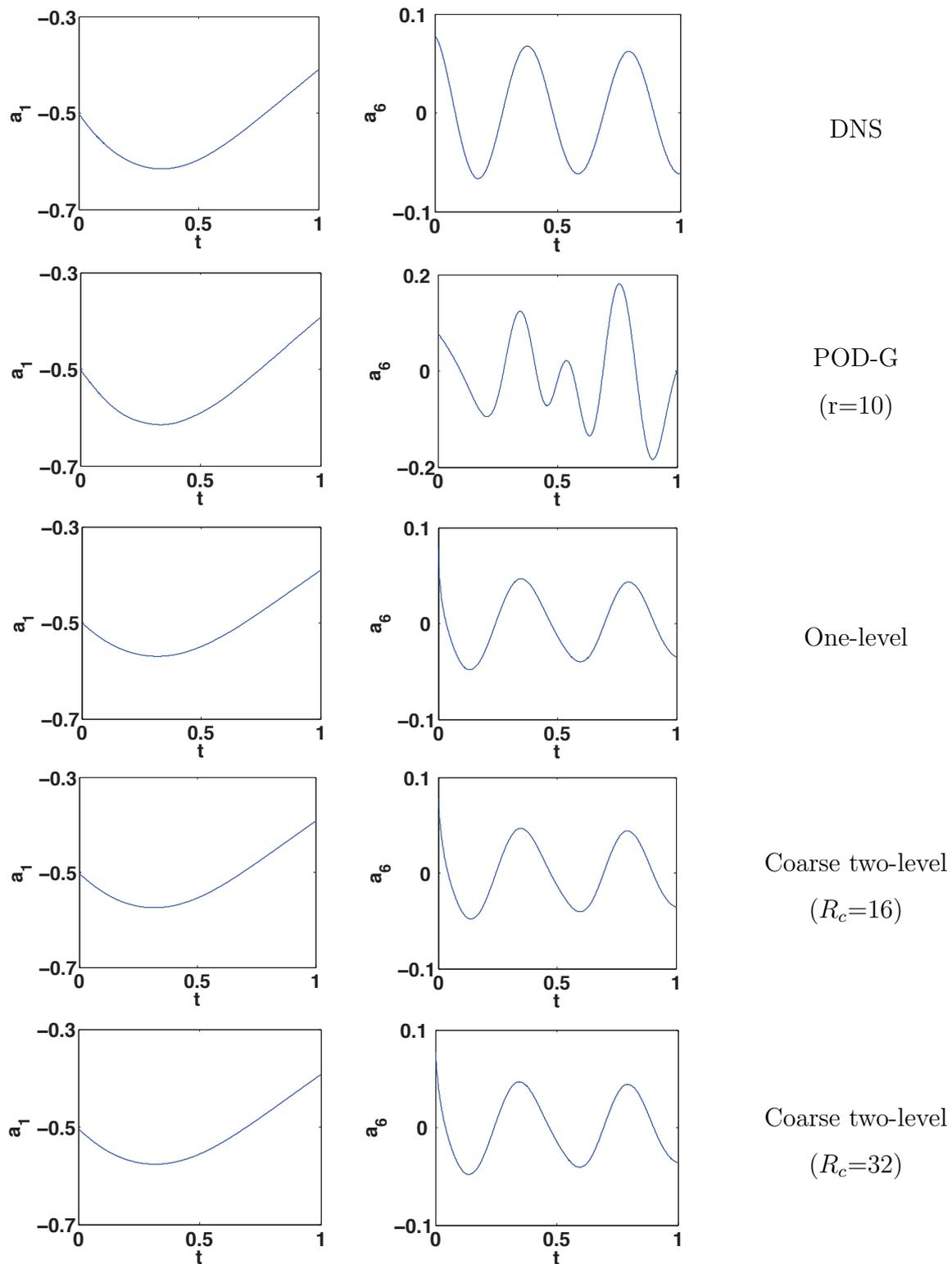


Figure 3.3: (Figure continued on next page.) 1D Burgers equation with $\nu = 10^{-3}$. Temporal evolution of POD coefficients $a_1(\cdot)$ and $a_6(\cdot)$ over $[0, 1]$.

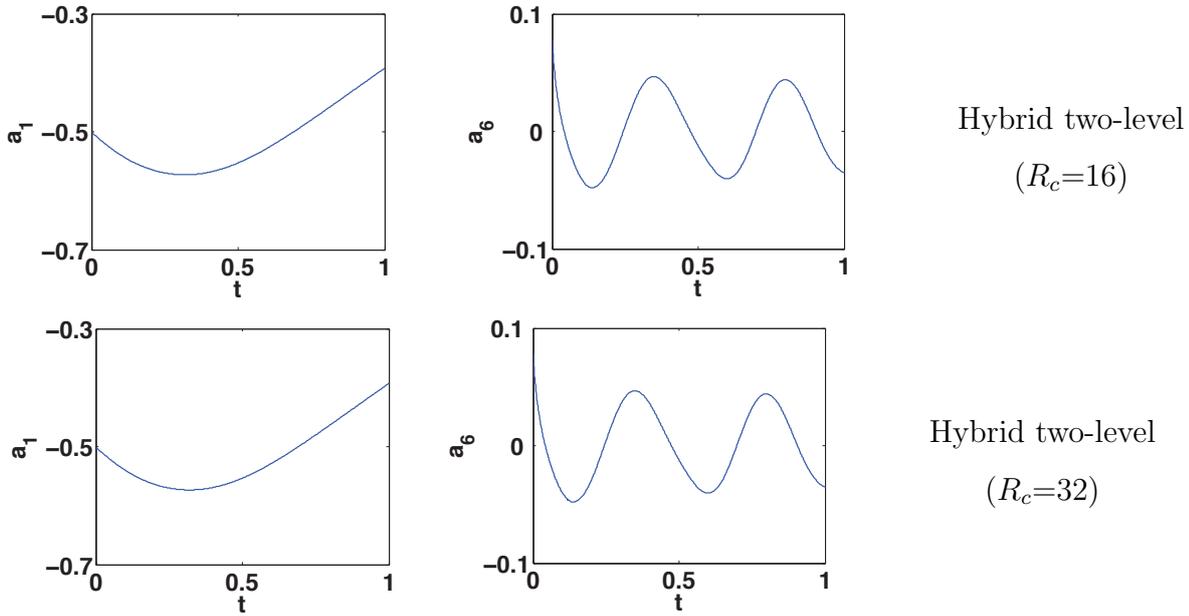


Figure 3.4: Figure 3.3 continued. See figure caption on previous page.

CFD solver on a 4-processor platform and record the snapshot data of the flow field. The details of the numerical algorithm employed are provided in Section 3.3.3 and the appendix in [223]. The reader is referred to [4, 6, 8] for validation and verification results of the 2D version of the CFD solver.

The qualitative behavior of all the models tested is represented by the time evolution of the POD coefficients $a_1(\cdot)$ and $a_3(\cdot)$ and the 2D projections of the limit cycle on the phase planes — (a_1, a_i) , $i = 2, 3$ presented in Figures 3.5-3.6 and 3.7-3.8, respectively. We note that the other POD coefficients have a similar behavior; for clarity of exposition, we include only $a_1(\cdot)$ and $a_3(\cdot)$.

After we collect the DNS data, we construct a structured quadratic finite element mesh with nodes coinciding with the nodes used in the original DNS finite volume discretization. For the last period, we generate 40 snapshots, then use the method of snapshots [206] to generate the POD basis. Deane et al. [65] observed that 20 snapshots are sufficient for the construction of the first eight eigenfunctions at $Re = 100-200$. In general, numerical studies [175] suggest that the first r POD modes, where r is even, resolve the first $r/2$ temporal harmonics and require a $2r$ number of snapshots for convergence.

This POD basis is then used in all POD-ROMs for the 2D flow past a circular cylinder.

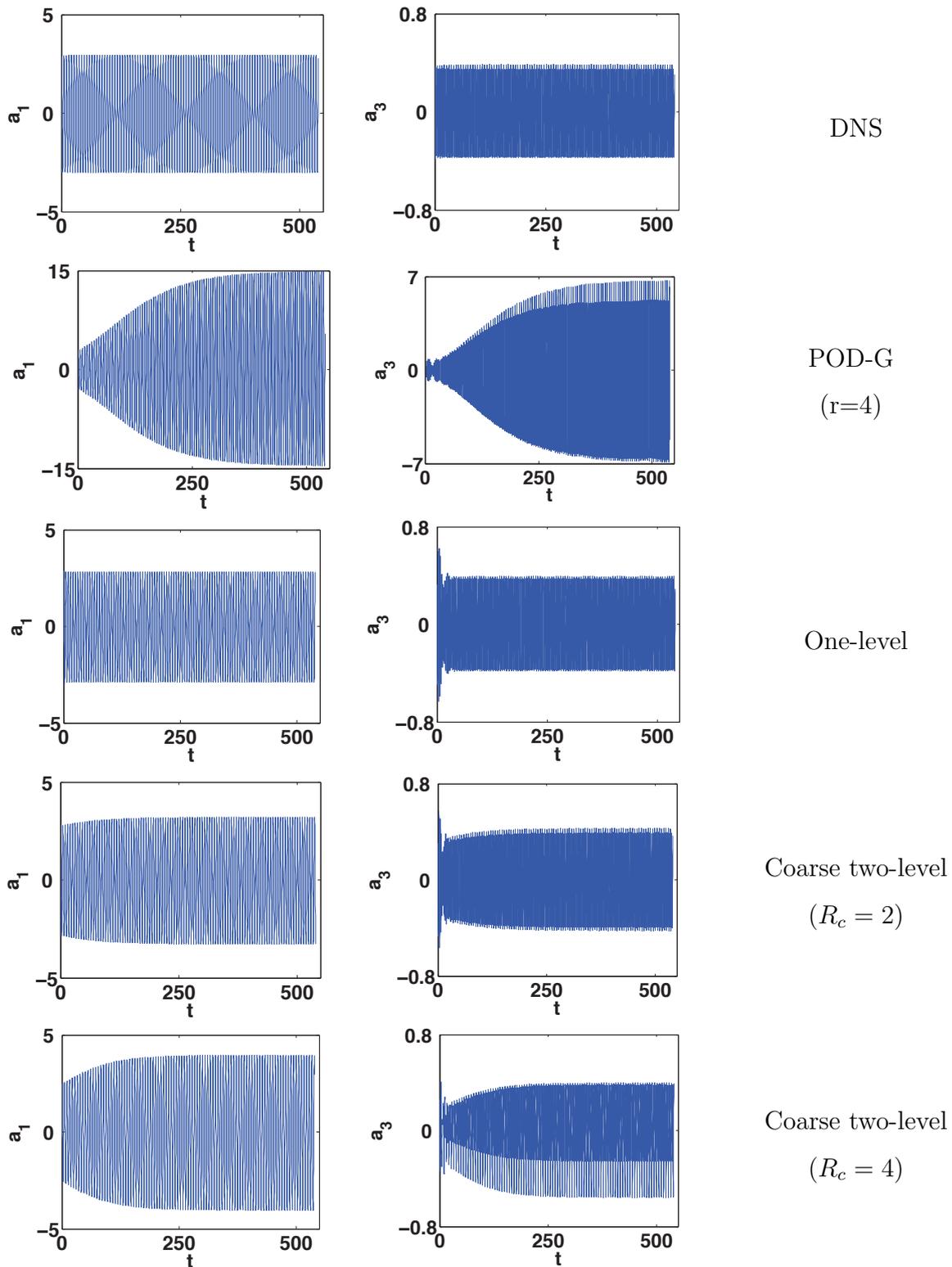


Figure 3.5: (Figure continued on next page.) 2D flow past a cylinder at $Re = 200$. Temporal evolution of POD coefficients $a_1(\cdot)$ and $a_3(\cdot)$ over 100 vortex shedding cycles.

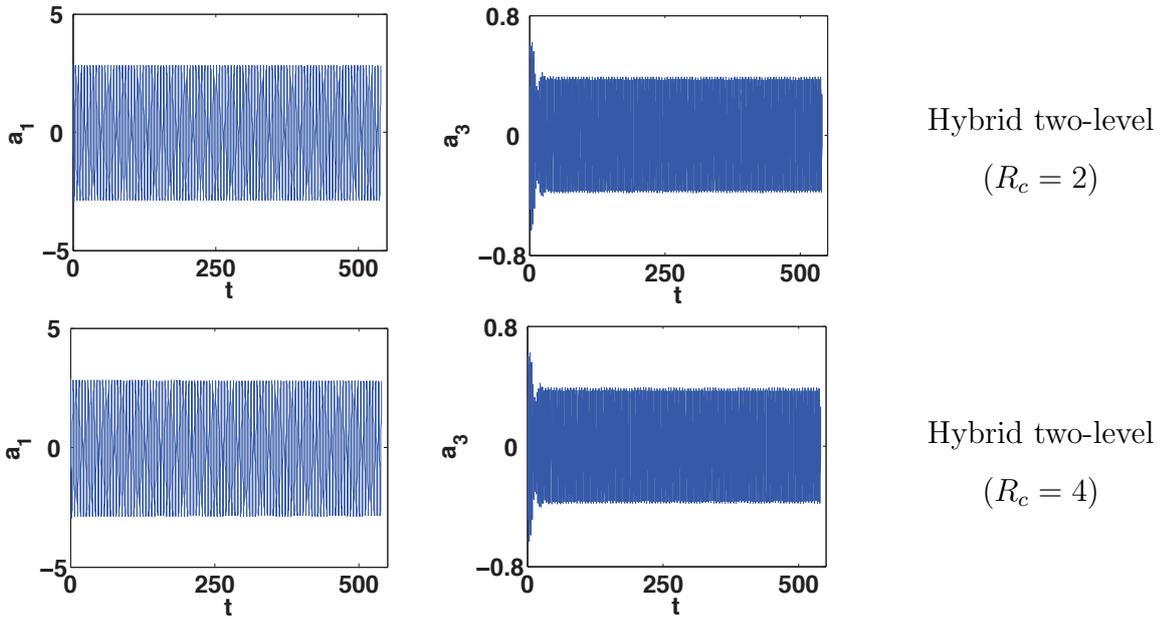


Figure 3.6: Figure 3.5 continued. See figure caption on previous page.

For all these POD models, the time discretization was effected by using the explicit Euler method with $\Delta t = 1.4 \times 10^{-3}$.

For comparison purposes, Figures 3.5-3.6 and 3.7-3.8 include the time evolution of the POD coefficients obtained by projecting the DNS onto each POD mode and the 2D projection of the limit cycle on the phase planes, respectively.

The POD-G-ROM (3.1) with r small ($r = 4$) yields poor results, although it is computationally efficient (its CPU time is 246 s). Indeed, the temporal evolution of POD coefficients and the 2D projections of the limit cycle on the phase planes in Figures 3.5-3.6 and 3.7-3.8, respectively, are inaccurate. The amplitude of the temporal evolution of the POD coefficient $a_3(\cdot)$ for POD-G-ROM with $r = 4$ is *more than eight times larger* than that for the DNS. The relative error of POD-G-ROM with $r = 4$ is 9.57×10^{-2} . In fact, the solution diverges to a spurious limit cycle. Furthermore, the 2D projections of the limit cycle on the phase planes — $(a_1, a_i), i = 2, 3$ for POD-G-ROM with $r = 4$ are significantly different from the projected DNS. Thus, although the first 4 POD modes capture 98.74% of the system's energy, these results clearly indicate the need for closure modeling.

We will use the S-POD-ROM (2.16) with an artificial viscosity coefficient $C = 1.9 \times 10^{-3}$ as a closure model. Although the Smagorinsky model targets 3D turbulent flows, for this test

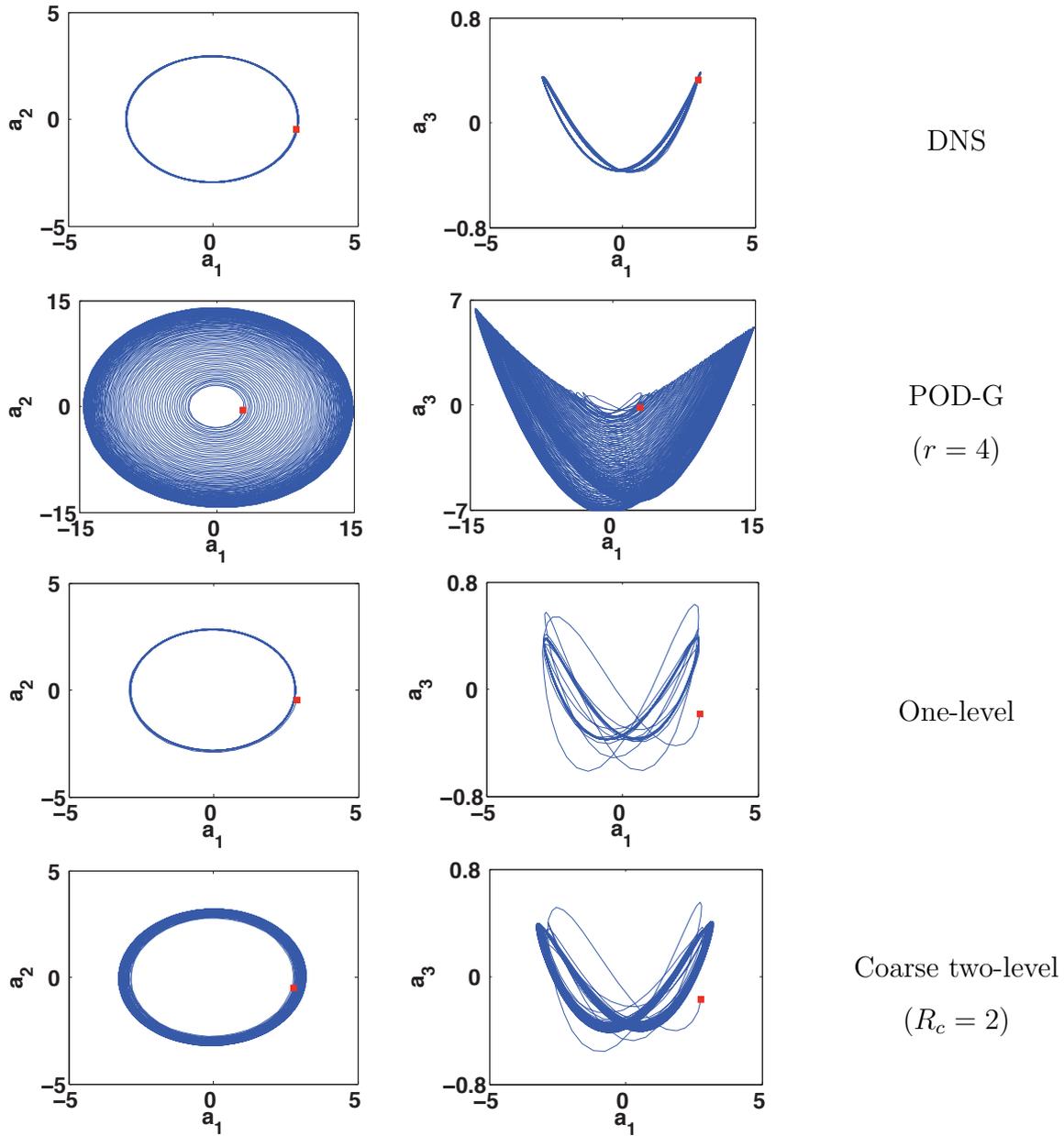


Figure 3.7: (Figure continued on next page.) 2D flow past a cylinder at $Re = 200$. 2D projections of the limit cycle on the phase planes (a_1, a_2) and (a_1, a_3) over 100 vortex shedding cycles. The red square denotes the initial position of $(a_1(0), a_i(0))$, $i = 2, 3$.

problem it plays the role of numerical stabilization. Nevertheless, by using formulas (2.9) and (2.10), we can determine the values for the corresponding Smagorinsky constant and filter radius: $C_S = 0.1845$ and $\delta = 0.1987$, respectively. For the numerical discretization of this nonlinear POD-ROM, we will use three different strategies: (i) the one-level algorithm

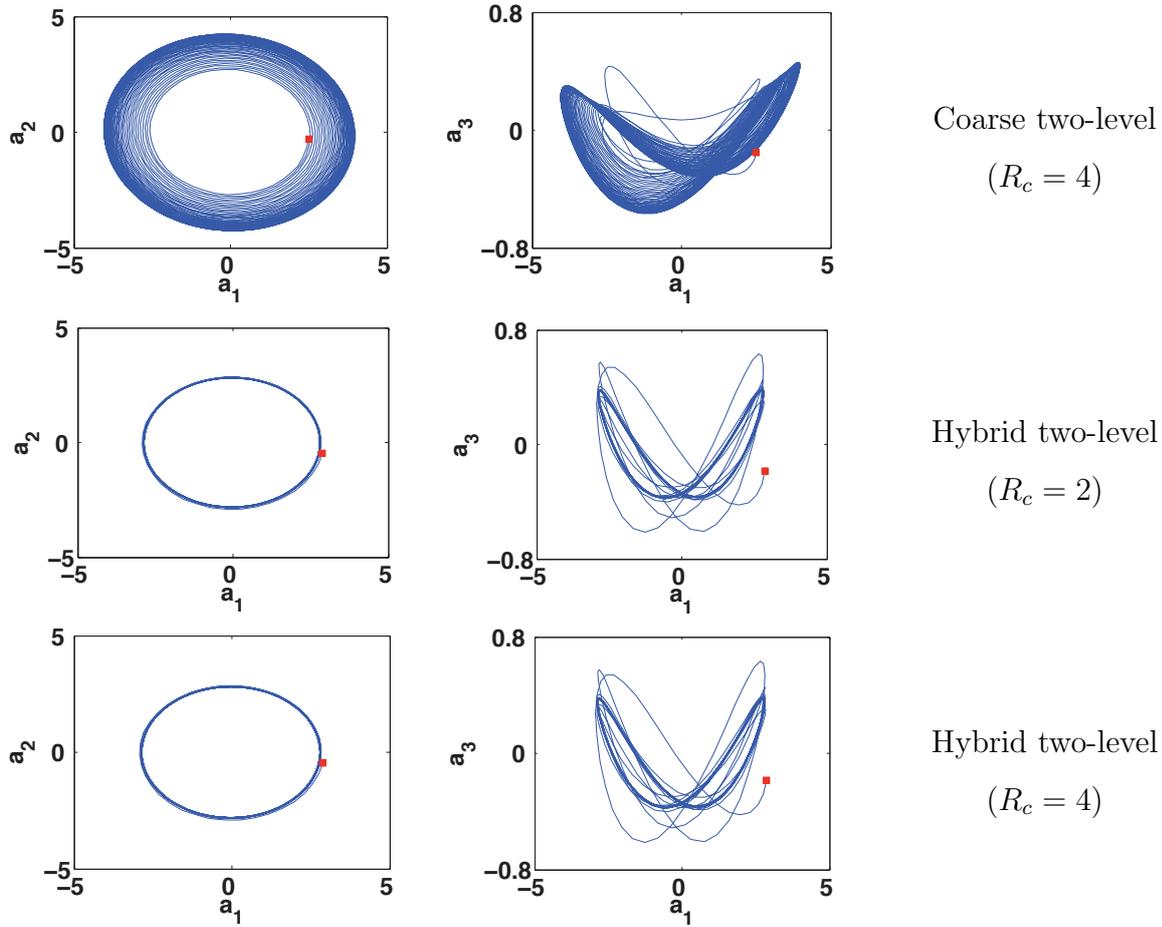


Figure 3.8: Figure 3.7 continued. See figure caption on previous page.

(3.2); (ii) the hybrid two-level algorithm (3.3); and (iii) the coarse two-level algorithm (3.4).

The one-level algorithm (3.2) with $r = 4$ yields the temporal evolution of POD coefficients and the 2D projections of the limit cycle on the phase planes — (a_1, a_i) , $i = 2, 3$ in Figures 3.5-3.6 and 3.7-3.8. Comparing these results with those for the DNS, it is clear that the one-level algorithm yields relatively accurate results, although it uses only a few ($r = 4$) POD basis functions. Indeed, the one-level algorithm (3.2) produces results that are much closer to those for DNS than the results for POD-G-ROM with $r = 4$. The relative error of the one-level algorithm is 9.0×10^{-3} , whereas that of POD-G-ROM with $r = 4$ is 9.57×10^{-2} . This, however, comes at a high computational cost. The CPU time is 246 s for POD-G and is 4.01×10^4 s for the one-level algorithm.

To implement our two-level algorithms (3.3) and (3.4), we again used the simplest coarsening and interpolation procedures. The coarsening of the mesh was effected by skipping

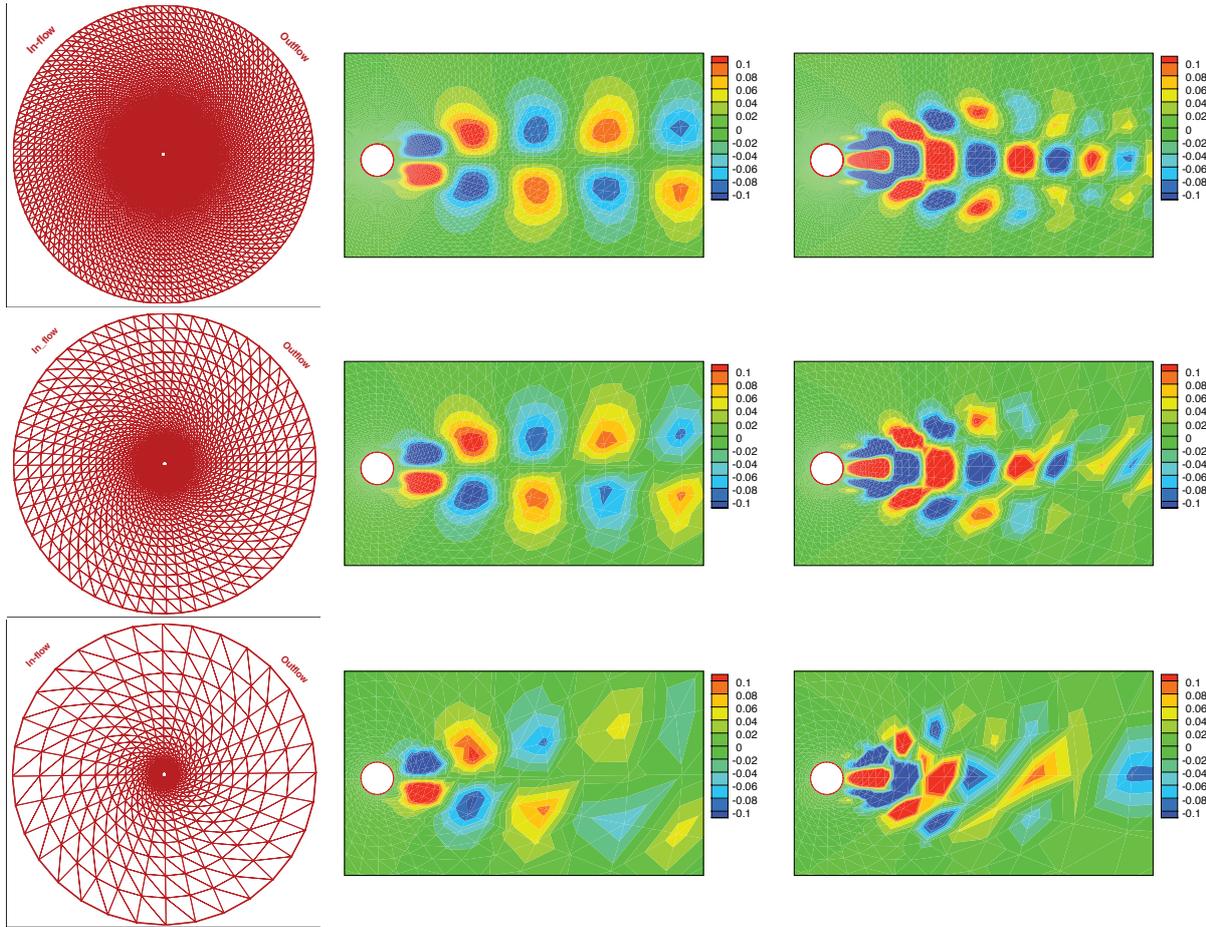


Figure 3.9: 2D flow past a cylinder at $Re = 200$: Mesh (left column), streamwise component of φ_1 (middle column), and streamwise component of φ_3 (right column). (For clarity, the POD modes are plotted only in a neighborhood of the cylinder.) Original fine mesh (top row), mesh coarsened by a factor of 2 (middle row), and mesh coarsened by a factor of 4 (bottom row).

nodes in the radial and azimuthal directions. For example, when the coarsening factor $R_c = 2$, we skip every other node in both directions; when $R_c = 4$, we skip over three nodes in both directions (and so on). To interpolate a POD basis function φ_j on the coarse mesh, we just keep the values of φ_j corresponding to the nodes of the coarse mesh. The effect of the mesh coarsening and interpolation is shown in Figure 3.9 for two coarsening factors ($R_c = 2$ and $R_c = 4$). It is clear from Figure 3.9 that, although the accuracy of the POD modes degrades as the mesh is coarsened, the *geometric structure is preserved*.

The *coarse two-level algorithm* (3.4) with $r = 4$ and $R_c = 2, 4$ yields the temporal

evolution of POD coefficients and the 2D projections of the limit cycle on the phase planes — $(a_1, a_i), i = 2, 3$ in Figures 3.5-3.6 and 3.7-3.8. Comparing these results with those for the one-level algorithm (3.2), we conclude that the coarse two-level algorithm (3.4) and the one-level algorithm (3.2) yield results that are qualitatively similar. The coarse two-level algorithm, however, reduces that computational time by an order of magnitude (see Table 3.2).

The *hybrid two-level algorithm* (3.3) with $r = 4$ and $R_c = 2, 4$ yields the temporal evolution of POD coefficients and 2D projections of the limit cycle on the phase planes — $(a_1, a_i), i = 2, 3$ in Figures 3.5-3.6 and 3.7-3.8. Comparing these results with those for the one-level algorithm (3.2), we conclude that the hybrid two-level algorithm (3.3) yields an improved accuracy (a shorter initial transition in the temporal evolution of the POD coefficients $a_i(\cdot)$ and tighter 2D projections of the limit cycle on the phase planes) while keeping the CPU time at a modest level (see Table 3.2).

To measure the efficiency of our two-level algorithms, we present the speed-up factors S_f defined in (3.7) along with relative errors for both two-level algorithms in Table 3.2. Again, the relative error of the one-level algorithm is 9.0×10^{-3} . The data in Table 3.2 clearly shows that both two-level algorithms significantly decrease the CPU time (by up to an order of magnitude), while maintaining similar accuracy as the more expensive one-level algorithm.

R_c	coarse two-level		hybrid two-level	
	S_f	error	S_f	error
2	4.03	1.02×10^{-2}	3.97	9.0×10^{-3}
4	15.91	1.12×10^{-2}	14.58	9.0×10^{-3}

Table 3.2: 2D flow past a cylinder at $\text{Re} = 200$. Speed-up factors S_f and relative errors for different coarsening factors R_c .

Thus, for this test problem, we conclude that the two-level algorithms achieve the same order of accuracy as the standard one-level algorithm, while decreasing the CPU time by more than *an order of magnitude*. We also note that, in this case, the hybrid two-level algorithm (3.3) performs better than the coarse two-level algorithm (3.4).

We now compare the computational costs of the DNS and the POD-ROMs. This com-

parison, however, is challenging, since the discretizations used in the two approaches are completely different. Indeed, the spatial discretization used in the DNS was the finite volume method, whereas for POD-ROMs we used a finite element method. Furthermore, the time-discretization used in DNS was second-order (Crank-Nicolson and Adams-Bashforth), whereas in POD-ROM we used a first-order time discretization (explicit Euler). The time-steps employed were also different: $\Delta t = 2 \times 10^{-3}$ in the DNS and $\Delta t = 1.4 \times 10^{-3}$ in the POD-ROMs. Finally, the DNS was performed on a parallel machine (on 4 processors), whereas the POD-ROMs were carried out on a single-processor machine.

Keeping in mind these significant differences between the two types of discretization employed, the two-level algorithm significantly reduced the computational time of the DNS: The CPU time for the DNS was 1.17×10^4 s, whereas for the POD-ROM with the two-level discretization it was 2.52×10^3 s.

3.3.3 3D Flow Past a Circular Cylinder at $\text{Re} = 1000$

In this section, we investigate the two-level algorithms in the numerical simulation of 3D flow past a circular cylinder at $\text{Re} = 1000$. A parallel CFD solver is employed to generate the DNS data [4]. For details on numerical discretization, the reader is referred to the appendix in [223]. The qualitative behavior of all reduced-order models investigated is represented by the time evolution of the POD coefficients $a_1(\cdot)$ and $a_4(\cdot)$ in Figures 3.11-3.12. We note that the other POD coefficients have a similar behavior; for clarity of exposition, we included only $a_1(\cdot)$ and $a_4(\cdot)$.

We record snapshot data of the flow field to compute the POD basis functions. At $\text{Re} = 1000$, the wake is fully turbulent and the flow is not periodic anymore as observed in the lower Re regime. We notice that there is a spectrum of frequencies present in the flow field. This requires a large snapshot data set comprising many shedding cycles.

By using multiple processors, we reduced the load per processor and were able to record a large data set of the flow field. Each processor records 1000 snapshots of the velocity field (u_1, u_2, u_3) over 15 shedding cycles on its local grid points. Thus, parallel computing is a key enabling technology for model reduction in 3D flows. Using the method of snapshots [206], we construct a correlation matrix and compute the POD basis $\{\varphi_1, \dots, \varphi_d\}$ using MPI-

ALLREDUCE (SUM) operation [5]. We observe that the 3D POD modes are not symmetric anymore due to presence of multiple frequencies in the dynamical system as shown in Figure 3.10.

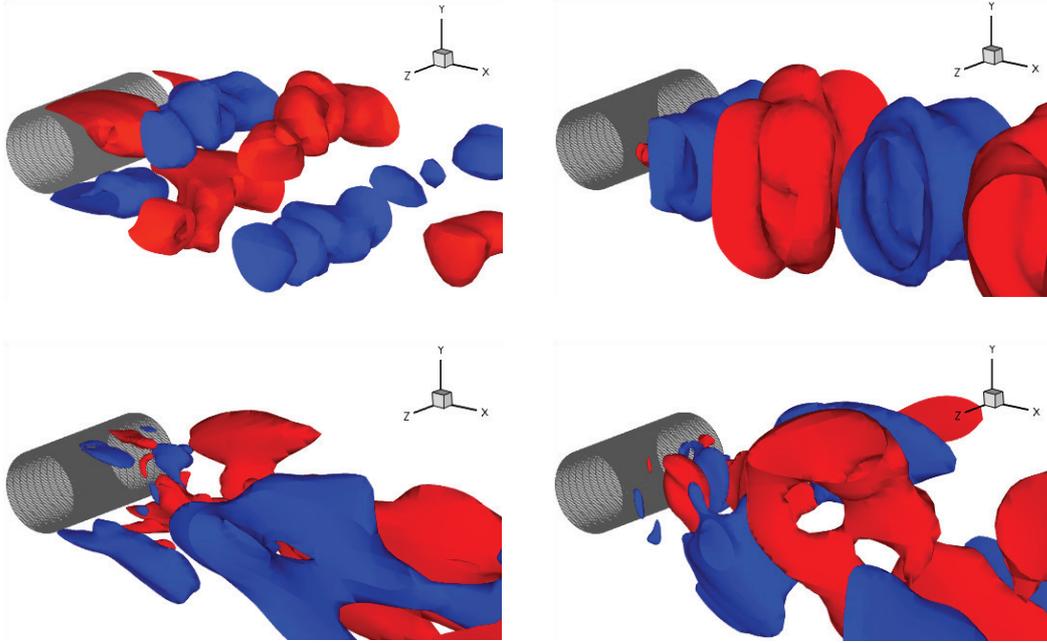


Figure 3.10: 3D flow past a cylinder at $Re = 1000$. First streamwise POD mode (top left), first normal POD mode (top right), third streamwise POD mode (bottom left), and third normal POD mode (bottom right).

These POD modes are then interpolated onto a structured quadratic finite element mesh with nodes coinciding with the nodes used in the original DNS finite volume discretization. The POD basis is then used in all POD-ROMs that we investigate next. For all these POD-ROMs, the time discretization was effected by using the explicit Euler method with $\Delta t = 7.5 \times 10^{-4}$. We note that POD-ROMs have also been extended to the pressure field [8, 172]. The key advantage of pressure ROM is the computation of hydrodynamic forces on the structure. Akhtar et al. [8] developed a pressure-Poisson based ROM that requires the POD modes of the pressure fields in addition to those for the velocity field. Their model predicted the lift and drag forces on a cylinder at $Re = 100$. In the current study, however, we restrict ourselves to the ROM of the velocity field. Thus, lift and drag forces cannot be computed and compared with those from DNS.

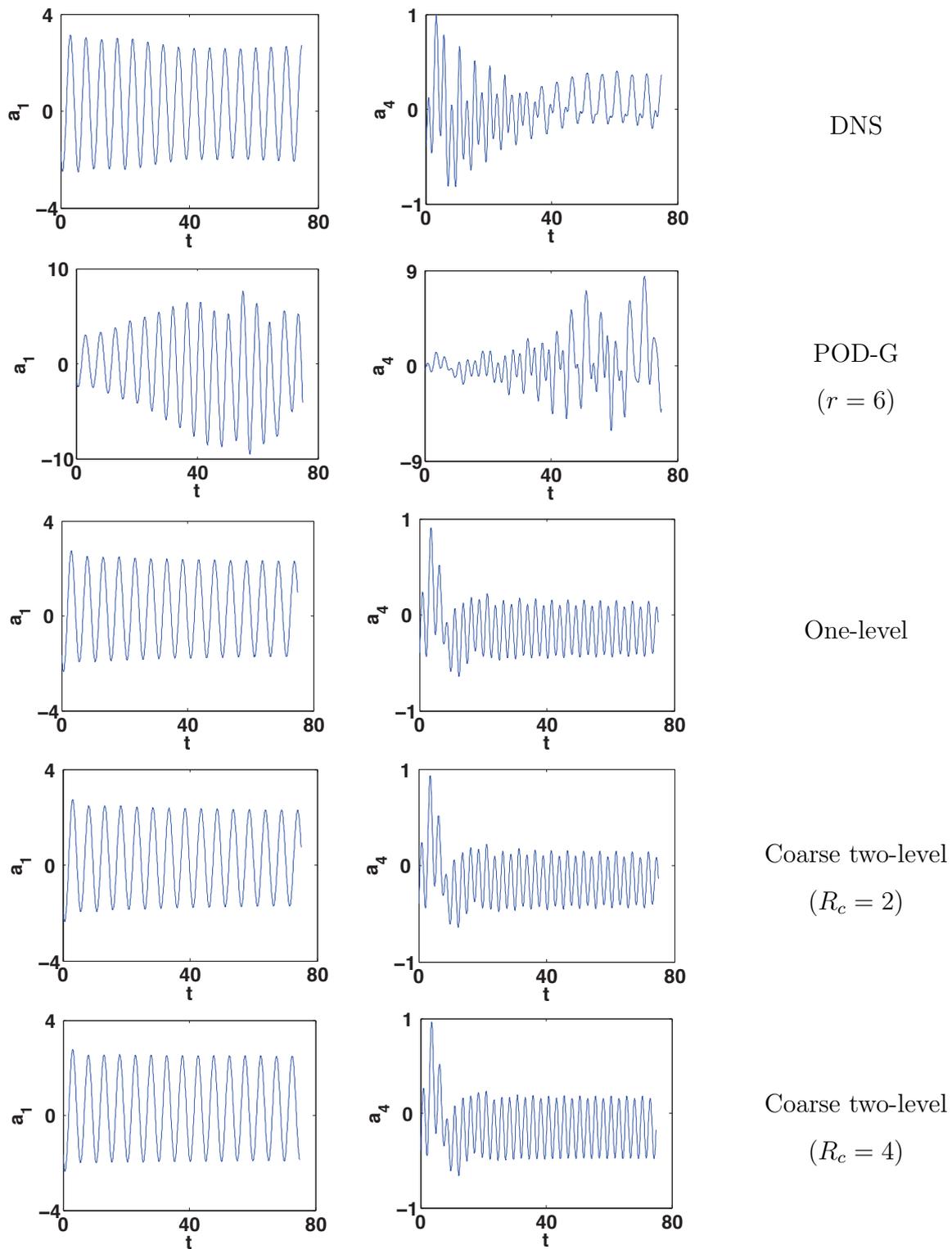


Figure 3.11: (Figure continued on next page.) 3D flow past a cylinder at $Re = 1000$. Temporal evolution of POD coefficients $a_1(\cdot)$ and $a_4(\cdot)$ over the time interval $[0, 75]$.

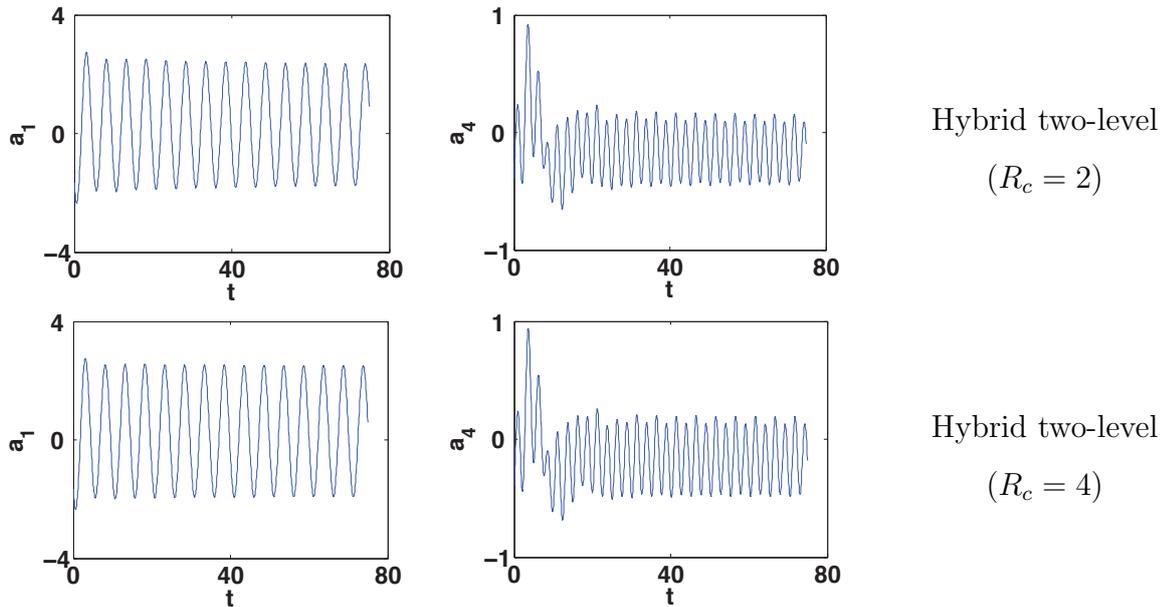


Figure 3.12: Figure 3.11 continued. See figure caption on previous page.

For comparison purposes, we include the time evolution of the POD coefficients obtained by projecting the DNS onto each POD mode.

The POD-G-ROM (3.1) with r small ($r = 6$) is not appropriate, although it is computationally efficient (its CPU time is 486 s). Indeed, the temporal evolutions of POD coefficients in Figures 3.11-3.12 are inaccurate: The amplitude of the temporal evolution of the POD coefficient $a_4(\cdot)$ for $r = 6$ is *nine times larger* than that for the DNS projection. The relative error of POD-G-ROM with $r = 6$ is 1.23×10^{-1} . Thus, although the first 6 POD modes capture 84% of the system's energy, these results clearly indicate the need for closure modeling.

We will use the S-POD-ROM (2.16) with an artificial viscosity coefficient $C = 4 \times 10^{-4}$ as a POD closure model for turbulent flows. By using formula (2.9), we can determine the values for the corresponding Smagorinsky constant and filter radius: $C_S = 0.1426$ and $\delta = 0.1179$, respectively. For the numerical discretization of the resulting nonlinear POD-ROM, we will use three different strategies: (i) the one-level algorithm (3.2); (ii) the hybrid two-level algorithm (3.3); and (iii) the coarse two-level algorithm (3.4).

The one-level algorithm (3.2) with $r = 6$ yields the temporal evolution of POD coefficients in Figures 3.11-3.12. Comparing these results with those for the DNS, it is clear that the

one-level algorithm yields relatively accurate results although it uses only a few ($r = 6$) POD basis functions. Indeed, the one-level algorithm (3.2) produces results that are much closer to those for the DNS than the results for POD-G-ROM with $r = 6$. The relative error of the one-level algorithm is 4.46×10^{-2} , whereas that of POD-G-ROM with $r = 6$ is 1.23×10^{-1} . This, however, comes at a high computational cost. Indeed, the CPU time is 486 s for POD-G-ROM with $r = 6$ and is 5.32×10^5 s for the one-level algorithm.

We used the simplest coarsening and interpolation procedures. The coarsening of the mesh was effected by skipping nodes in the radial and azimuthal directions. For example, when the coarsening factor $R_c = 2$, we skip every other node in both directions; when $R_c = 4$, we skip over three nodes in both directions (and so on). For this test problem, we only coarsened the mesh in the x and y directions, since the mesh resolution in the z direction was already low. To interpolate a POD basis function φ_j on the coarse mesh, we just keep the values of φ_j corresponding to the nodes of the coarse mesh.

The *coarse two-level algorithm* (3.4) with $r = 6$ and $R_c = 2, 4$ yields the temporal evolution of POD coefficients in Figures 3.11-3.12. Comparing these results with those for the one-level algorithm (3.2), we conclude that the coarse two-level algorithm (3.4) and the one-level algorithm (3.2) yield results that are qualitatively similar. The coarse two-level algorithm, however, reduces that computational time by an order of magnitude (see Table 3.3).

The *hybrid two-level algorithm* (3.3) with $r = 6$ and $R_c = 2, 4$ yields the temporal evolution of POD coefficients in Figures 3.11-3.12. Comparing these results with those for the one-level algorithm (3.2), we conclude that the hybrid two-level algorithm (3.3) yields an improved accuracy while keeping the CPU time at a modest level (see Table 3.3).

Although the main goal of this chapter is to investigate whether, for a given closure model (the basic Smagorinsky model), the two-level algorithms represent an efficient, yet accurate alternative to the one-level algorithm, we note that the POD-ROM results are not always accurate when compared with DNS results. Indeed, we note that the time evolutions for the POD-ROMs' a_1 coefficients in Figures 3.11-3.12 are close to those corresponding to the DNS. The time evolutions for the POD-ROMs' a_4 coefficients, however, are less accurate. We believe that these inaccuracies are due to the fact that the closure model employed is

just the standard Smagorinsky model; more accurate closure models (such as the DS-POD-ROM and the VMS-POD-ROM) yield improved results (see, for example, Figure 5.10). We also emphasize that the plots in Figures 3.11-3.12 represent instantaneous quantities (not averages), which are generally hard to capture accurately by turbulence closure models.

To measure the efficiency of our two-level algorithms, we present the speed-up factors S_f given in (3.7) along with relative errors for both two-level algorithms in Table 3.3. The relative error of the one-level algorithm is 4.46×10^{-2} . The data in Table 3.3 clearly shows that both two-level algorithms significantly decrease (by over an order of magnitude) the CPU time while maintaining similar accuracy as the more expensive one-level algorithm.

R_c	coarse two-level		hybrid two-level	
	S_f	error	S_f	error
2	4.97	4.44×10^{-2}	5.22	4.52×10^{-2}
4	24.52	3.85×10^{-2}	24.18	4.73×10^{-2}

Table 3.3: 3D flow past a cylinder at $Re = 1000$. Speed-up factors S_f and relative errors for different coarsening factors R_c .

Thus, for this test problem, we conclude that the two-level algorithms achieve the same order of accuracy as the standard one-level algorithm, while decreasing the CPU time by more than *an order of magnitude*. We also notice that, in this case, the coarse two-level algorithm (3.4) performs better than the hybrid two-level algorithm (3.3).

As in the 2D test case, we mention that a comparison between the computational cost of the DNS and that of the POD-ROMs used is challenging, since the discretizations used in the two approaches are completely different. Indeed, the spatial discretization used in the DNS was the finite volume method, whereas for POD-ROMs we used a finite element method. Furthermore, the time-discretization used in DNS was second-order (Crank-Nicolson and Adams-Bashforth), whereas in POD-ROM we used a first-order time discretization (explicit Euler). The time-steps employed were also different: $\Delta t = 2 \times 10^{-3}$ in the DNS and $\Delta t = 7.5 \times 10^{-4}$ in the POD-ROM. Finally, the DNS was performed on a parallel machine (on 16 processors), whereas the POD-ROMs were carried out on a single-processor machine.

Keeping in mind these significant differences between the two types of discretization

employed, the two-level algorithm reduced the computational time of the DNS: The CPU time for the DNS was $3.58 \times 10^4 s$, whereas for the POD-ROM with the two-level discretization it was $2.17 \times 10^4 s$. We believe that by increasing the time-step, using a higher-order time-discretization and migrating the code to parallel machines, the computational cost of the two-level algorithm can be decreased even further. We plan to pursue these research directions in the future (see Section 6.4).

3.4 Summary

In this chapter, we proposed a two-level method for the numerical discretization of *nonlinear* POD closure models. This two-level method opens new research avenues in the investigation of physically accurate POD closure modeling strategies, similar to the state-of-the-art closure models used in LES of turbulent flows. The main idea behind the new two-level algorithm is the evaluation of the nonlinear POD closure model on a coarse mesh. This allows for an efficient computation of the nonlinear POD closure model at *each* time step.

We considered two versions of the two-level algorithm: (i) a hybrid two-level algorithm in which only the nonlinear term was evaluated on the coarse mesh; and (ii) a coarse two-level algorithm in which all the terms of the POD-ROM were evaluated on the coarse mesh. Both two-level algorithms were compared to a standard brute-force computational approach, which we denoted the one-level algorithm. We investigated the two-level algorithms in the numerical simulation of three test problems: (i) the one-dimensional Burgers equation with a small diffusion parameter $\nu = 10^{-3}$; (ii) a 2D flow past a cylinder at Reynolds number $Re = 200$; and (iii) a 3D flow past a cylinder at Reynolds number $Re = 1000$. For completeness, we also included results with POD-G-ROM (i.e., a POD-ROM without any closure model) with a low number of POD modes, as well as a DNS projection of the evolution of the POD modes, which served as benchmark for our numerical simulations. We emphasize that the goal of this chapter is to propose an efficient numerical discretization of a given nonlinear POD closure model, and *not* to advocate a specific POD closure model. Therefore, we just chose one of the standard closure models used in LES, the Smagorinsky model.

For all three numerical tests, we drew the same conclusions. The two-level algorithms achieved the same order of accuracy as the standard one-level algorithm, but they both decreased the computational cost of the latter by more than an *order of magnitude*. Thus, the two-level algorithms appear as promising approaches in the numerical discretization of nonlinear closure modeling strategies for POD-ROMs. The coarse two-level algorithm performed better than the hybrid two-level algorithm in the 1D and 3D tests, whereas the latter performed better in the 2D test. The differences between the two two-level algorithms in terms of computational efficiency and numerical accuracy, however, were minimal. Since there are no major differences in terms of implementation, we conclude that both two-level algorithms should be considered in practical POD-ROM computations. We also note that, in general, the errors in both two-level algorithms seem to decrease as R_c increases. This behavior, however, is not always observed (e.g., the coarse two-level algorithm in Table 3.2 and the hybrid two-level algorithm in Table 3.3). This behavior is counterintuitive, since coarsening the mesh should result in an error increase. We do mention, however, that given the mesh sizes considered in our study (especially those for the 2D and 3D numerical tests), we cannot infer the true asymptotic behavior of the error with respect to the mesh size.

Chapter 4

Numerical Analysis [†]

4.1 Introduction

In this chapter, we start the development of a rigorous error analysis framework for the new POD-ROMs. The goal of such an analysis is twofold: one is to build solid mathematical foundations to the new POD-ROMs; the other is to determine optimal phenomenological parameters in the closure methods.

Although POD closure models are developed to achieve a low-order approximation of the complex system for structurally dominated turbulent flows, as the first step of our investigations, we analyze the closure methods for simplified equations: The Burgers equation and a convection-dominated convection-diffusion-reaction equation with small diffusion parameters. The closure strategies we investigate are: Smagorinsky and VMS methods. The former is one of the standard LES closure methods, whereas the latter is considered as state-of-the-art turbulence modeling strategy.

We emphasize that, although the Burgers equation and the convection-dominated convection-diffusion-reaction equation are always considered as simplifications of NSE and small diffu-

[†]The material in this chapter is based on the following reports:

J. Borggaard, T. Iliescu and Z. Wang. Artificial Viscosity Proper Orthogonal Decomposition. *Math. Comput. Model.*, 53 (1-2), 2011, pp. 269-279.

T. Iliescu and Z. Wang. Variational Multiscale Proper Orthogonal Decomposition: Convection-Dominated Convection-Diffusion Equations. *Math. Comp.*, in press, 2011.

sion parameters are corresponding to high Reynolds numbers in NSE, they do not represent the real turbulence. Therefore, the role of closure methods changes correspondingly. Instead of improving physical accuracy, they increase the numerical stability. Of course, once we fully understand the behavior of the POD closure models in these simplified settings, we will analyze them in the NSE setting.

This chapter is organized as follows: In Section 4.2, we perform the numerical analysis for an artificial viscosity POD model of the Burgers equation. In Section 4.3, we carry out the analysis for a variational VMS-POD-ROM (2.25) for the convection-dominated convection-diffusion-reaction equation. Concluding remarks are summarized in Section 4.4.

4.2 Artificial Viscosity POD-ROM

In this section, we present and analyze an artificial viscosity (AV) POD-ROM. The mathematical model we consider is the *Burgers equation*

$$\begin{cases} u_t - \nu u_{xx} + u u_x = f & \text{in } \Omega, \\ u(x, 0) = u_0(x) & \text{in } \Omega, \\ u(x, t) = g(x, t) & \text{on } \partial\Omega, \end{cases} \quad (4.1)$$

where ν is the diffusion parameter, f the forcing term, $\Omega \subset \mathbb{R}$ the computational domain, $t \in [0, T]$, with T the final time, $u_0(\cdot)$ the initial condition, and $g(\cdot)$ the boundary conditions. Without loss of generality, we assume that $g = 0$ in the sequel. We emphasize that Burgers equation (4.1), while being commonly used as a one-dimensional approximation of the NSE, does not model turbulence. We use it, however, for simplicity and clarity of exposition.

Let L^2 , H_0^1 , $W^{1,3}$, $W_0^{1,3}$ denote the usual Sobolev spaces on Ω [2] and $(\cdot, \cdot) = (\cdot, \cdot)_{L^2}$. Let $a : H_0^1 \times H_0^1 \rightarrow \mathbb{R}$ be defined as

$$a(\varphi, \psi) := \nu (\varphi', \psi') \quad \forall \varphi, \psi \in H_0^1. \quad (4.2)$$

For all $\varphi, \psi \in H_0^1$, we define

$$(F(\varphi), \psi) := (\varphi \varphi', \psi), \quad (4.3)$$

Furthermore, for $\varphi, \psi \in W^{1,3}$ and c a positive constant, we define

$$(G(\varphi), \psi) := c(|\varphi'| \varphi', \psi'). \quad (4.4)$$

Let $u_0 \in L^2$ and $f \in \mathcal{C}(0, T; L^2)$. Then the weak formulation of (4.1) reads

$$\begin{cases} (u_t, \varphi) + a(u, \varphi) + (F(u), \varphi) = (f, \varphi) & \forall \varphi \in H_0^1, \\ (u(0), \chi) = (u_0, \chi) & \forall \chi \in L^2. \end{cases} \quad (4.5)$$

We denote the time-step $\Delta t := T/M$ and the time instances $t_k = k \Delta t$, $k = 0, 1, \dots, M$. Following [137], we consider the following $2M + 1$ snapshots

$$\{u(t_0), u(t_1), \dots, u(t_M), \bar{\partial}u(t_1), \dots, \bar{\partial}u(t_M)\}, \quad (4.6)$$

where $\bar{\partial}u(t_k) = \frac{u(t_k) - u(t_{k-1})}{\Delta t}$. As explained in Remark 1 in [137], we included the finite difference quotients $\bar{\partial}u(t_k)$ in the set of snapshots in order to avoid an extra $(\Delta t)^{-2}$ factor in the error estimates. This point is further elaborated in the proof of Corollary 1 in Section 4.3 (on Page 87), where it is shown how this expanded POD snapshot set influences the error analysis. A different, very interesting point of view was espoused by Chaturantabut and Sorensen in [54]. The authors argued that the separate POD basis they used in DEIM to approximate the nonlinearity is closely related to the finite difference quotient $\bar{\partial}u(t_k)$ used in [137], since “ $\frac{\mathbf{y}_{k+1} - \mathbf{y}_k}{\Delta t} \approx \dot{\mathbf{y}}(t_k) = f(\mathbf{y}_k)$, where $\mathbf{y}_k \approx \mathbf{y}(t_k)$ and $\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y})$.”

By using the backward Euler method for the time discretization and denoting $\bar{\partial}u_r^k = \frac{u_r^k - u_r^{k-1}}{\Delta t}$, the POD-G-ROM corresponding to (4.5) is

$$\begin{cases} (\bar{\partial}u_r^k, \varphi) + a(u_r^k, \varphi) + (F(u_r^k), \varphi) = (f(t_k), \varphi) & \forall \varphi \in X^r, \\ (U_0, \chi) = (u_0, \chi) & \forall \chi \in X^r, \end{cases} \quad (4.7)$$

where X^r is the space spanned by the first r POD basis functions defined in (1.3). As mentioned in the introduction, in turbulent flow computations, the simple POD-G-ROM (4.7) would not suffice. The model we propose herein is an improvement of the mixing-length model in [20]. Specifically, we use an AV term similar to the one used in the Smagorinsky model [211], which yields the AV Burgers equation

$$\begin{cases} (u_t, \varphi) + a(u, \varphi) + (F(u), \varphi) + (G(u), \varphi) = (f, \varphi) & \forall \varphi \in W_0^{1,3}, \\ (u(0), \chi) = (u_0, \chi) & \forall \chi \in L^2. \end{cases} \quad (4.8)$$

This approach yields the following *artificial viscosity POD model (AV-POD-ROM)*

$$\begin{cases} (\bar{\partial}u_r^k, \varphi) + a(u_r^k, \varphi) + (F(u_r^k), \varphi) + (G(u_r^k), \varphi) = (f(t_k), \varphi) & \forall \varphi \in X^r, \\ (U_0, \chi) = (u_0, \chi) & \forall \chi \in X^r. \end{cases} \quad (4.9)$$

We emphasize again that, since we illustrate our new model in the context of Burgers equation, its role is somehow different from its original purpose. Indeed, in the numerical simulation of turbulent flows, the AV-POD-ROM (4.9) should dissipate energy according to the well known concept of energy cascade [32, 199] (see also Section 1.4). In the context of Burgers equation, however, the AV-POD-ROM's role is to stabilize the numerical solution in test problems displaying shock-like behavior.

4.2.1 Error Estimates

In this section, we prove estimates for the *time discretization* error $\frac{1}{M} \sum_{k=1}^M \|u_r^k - u(t_k)\|^2$, where the approximation u_r^k is the solution of (4.9) and $u(t_k)$ is the solution of (4.8). To this end, we follow the presentation in [137]. We emphasize that, although the general approach that we use is that in [137], there are several differences stemming from the nonlinear AV term that was not present in the formulation found in [137]. We also note that the error estimates in this section, as well as those in [137, 138], only concern the time discretization and the POD truncation. For a finite element spatial discretization error analysis, the reader is referred to Section 4.3. A similar approach and alternative error estimates can be found in [155] and [109], respectively.

We start by listing several results that will be used throughout this section. The first is the Poincaré inequality: There exists a constant $\alpha > 0$ such that the following relationship between the L^2 -norm and the H^1 -seminorm holds

$$\|\varphi\|^2 \leq \alpha |\varphi|_1^2 \quad \forall \varphi \in H_0^1. \quad (4.10)$$

The bilinear form $a(\cdot, \cdot)$ is continuous and coercive: There exist constants β and κ such that

$$|a(\varphi, \psi)| \leq \beta \|\varphi\|_1 \|\psi\|_1 \quad \forall \varphi, \psi \in H_0^1 \quad (4.11)$$

$$|a(\varphi, \varphi)| \geq \kappa \|\varphi\|_1^2 \quad \forall \varphi \in H_0^1. \quad (4.12)$$

We will also use the following lemma [150, 164]:

Lemma 1 (*Strong monotonicity and Lipschitz continuity of G*) For all $\varphi_1, \varphi_2, \psi \in W^{1,3}$,

there exists a generic constant C independent of φ_1, φ_2 or ψ , such that the following inequalities hold:

$$(G(\varphi_1) - G(\varphi_2), (\varphi_1 - \varphi_2)) \geq C \|(\varphi_1 - \varphi_2)'\|_{L^3}^3, \quad (4.13)$$

$$(G(\varphi_1) - G(\varphi_2), \psi) \leq C \widetilde{M} \|(\varphi_1 - \varphi_2)'\|_{L^3} \|\psi'\|_{L^3}, \quad (4.14)$$

where $\widetilde{M} := \max \{ \|\varphi_1'\|_{L^3}, \|\varphi_2'\|_{L^3} \}$.

Using the strong monotonicity of G (4.13) and following [71, 72, 159], one can prove that both (4.8) and (4.9) have a unique solution.

For clarity of notation, we will denote by C_k , $k \in \mathbb{N}$, a generic constant that can depend on all the parameters in the system, except on r (the number of POD modes retained in the POD-ROM) and m (the number of snapshots).

We define the projection $P_R : H_0^1 \rightarrow X^r$ as follows: For all $u \in H_0^1$,

$$a(P_R u, \psi) = a(u, \psi) \quad \forall \psi \in X^r. \quad (4.15)$$

For Δt small enough, the projection operator P_R has the following approximation properties [137]:

$$\frac{1}{M} \sum_{k=1}^M \|u(t_k) - P_R u(t_k)\|_1^2 \leq C_1 \sum_{j=r+1}^d \lambda_j, \quad (4.16)$$

$$\frac{1}{M} \sum_{k=1}^M \|\bar{\partial} u(t_k) - P_R \bar{\partial} u(t_k)\|_1^2 \leq C_2 \sum_{j=r+1}^d \lambda_j. \quad (4.17)$$

We will also be using the following inverse estimate:

Lemma 2 *There exist constants C_3 and C_4 , such that, for all $v \in \mathcal{V}$, the following inequalities hold*

$$\|v\|_{L^3} \leq C_3 \|v\|_{L^2} \quad \text{and} \quad \|v\|_{L^2} \leq C_4 \|v\|_{L^3}. \quad (4.18)$$

Proof. Since $\{\varphi_1, \dots, \varphi_d\}$ is an orthonormal basis for \mathcal{V} and $v \in \mathcal{V}$, there exist scalars $v_i = (v, \varphi_i)$, $i = 1, \dots, d$, such that $v = \sum_{i=1}^d v_i \varphi_i$. Thus,

$$\|v\|_{L^3} = \left\| \sum_{i=1}^d v_i \varphi_i \right\|_{L^3} \leq \sum_{i=1}^d |v_i| \|\varphi_i\|_{L^3} \leq \left(\max_{i=1, \dots, d} \|\varphi_i\|_{L^3} \right) \sum_{i=1}^d |v_i|. \quad (4.19)$$

By using Lemma 2.4-1 in [135], there is a constant $C > 0$ independent of v such that

$$\sum_{i=1}^d |v_i| \leq C \|v\|_{L^2}, \quad (4.20)$$

Inequalities (4.19) and (4.20) clearly imply the first inequality in (4.18). The other inequality is derived in a similar fashion. \square

Remark 5 *Although the constants $\max_{i=1,\dots,d} \|\varphi_i\|_{L^3}$ in (4.19) and C in (4.20) do not depend on r , they might depend on M . We notice that the dependence on M also appears in the inverse estimate in Lemma 2 in [137] through the factor $\|S\|$ (with S being the stiffness matrix $S_{ij} = (\varphi'_j, \varphi'_i)$). We emphasize, however, that this dependence on M will disappear in the asymptotic limit $M \rightarrow \infty$ (or, equivalently, $\Delta t \rightarrow 0$). Since the error estimates we prove in this section are for $\Delta t \rightarrow 0$, we can safely assume that C_3 and C_4 in (4.18) do not depend on M .*

Next, we prove the following *a priori* stability estimates:

Lemma 3 *Assume that the solution to (4.8) satisfies $u' \in H_0^1$. Then the following stability estimates hold: There exist constants C_5, C_6 and C_7 , such that, for all $k = 1, \dots, m$, the following inequalities hold*

$$\|u'(t_k)\|_{L^2} \leq C_5, \quad \|u'(t_k)\|_{L^3} \leq C_6, \quad \text{and} \quad \|P_R u'(t_k)\|_{L^2} \leq C_7. \quad (4.21)$$

Proof. The first two inequalities in (4.21) follow immediately by letting $\varphi = u$ in (4.8), using $(F(u), u) = 0$, and (4.13) (the strong monotonicity of G). To prove the last inequality, we first use (4.2)

$$a(P_R u', \psi) = a(u', \psi) \quad \forall \psi \in X^r. \quad (4.22)$$

Next, we let $\psi := P_R u'$ in (4.22):

$$a(P_R u', P_R u') = a(u', P_R u'). \quad (4.23)$$

We then use the continuity of a (4.11) in (4.23):

$$a(P_R u', \psi) \leq \beta \|P_R u'\|_{L^2} \|u'\|_{L^2} \leq \beta C_4 \|P_R u'\|_{L^3} \|u'\|_{L^2}. \quad (4.24)$$

On the other hand, using the coercivity of a (4.12) in (4.23), we get

$$a(P_R u', P_R u') \geq \kappa \|P_R u'\|_{L^2}^2 \geq \kappa C_3^{-2} \|P_R u'\|_{L^3}^2. \quad (4.25)$$

Thus, (4.23)–(4.25) imply the last inequality in (4.21). \square

We are now ready to prove the main result of this section.

Theorem 1 *Let u be the solution of (4.8) and $\{u_r^k\}_{k=1}^M$ be the solution of (4.9). Assume that $u' \in H_0^1$ and $u_{tt} \in L^2(0, T; L^2)$. If Δt is small enough, then there exists a constant $C > 0$, such that*

$$\frac{1}{M} \sum_{k=1}^M \|u_r^k - u(t_k)\|^2 \leq C \left(\|u_0 - P_R u_0\|^2 + \sum_{j=r+1}^d \lambda_j + \Delta t^2 \right). \quad (4.26)$$

Proof. The proof follows along the same lines as the proofs of Theorems 7 and 11 in [137]. Therefore, we highlight the main differences and only sketch the rest of the proof. We start by decomposing the error as

$$u_r^k - u(t_k) = u_r^k - P_R u(t_k) + P_R u(t_k) - u(t_k) = \vartheta_k + \varrho_k, \quad (4.27)$$

where $\vartheta_k := u_r^k - P_R u(t_k)$ and $\varrho_k := P_R u(t_k) - u(t_k)$. Clearly, we have

$$\frac{1}{M} \sum_{k=1}^M \|u_r^k - u(t_k)\|^2 \leq \frac{2}{M} \sum_{k=1}^M \|\vartheta_k\|^2 + \frac{2}{M} \sum_{k=1}^M \|\varrho_k\|^2. \quad (4.28)$$

The approximation properties (4.16)–(4.17) and Poincaré's inequality (4.10) imply

$$\frac{1}{M} \sum_{k=1}^M \|\varrho_k\|^2 \leq C_8 \sum_{j=r+1}^d \lambda_j. \quad (4.29)$$

We now introduce the following notation: For $k = 1, \dots, m$, $\bar{\partial}\vartheta_k := \frac{\vartheta_k - \vartheta_{k-1}}{\Delta t}$, $v_k := u_t(t_k) - \bar{\partial}P_R u(t_k) = w_k + z_k$, $w_k := u_t(t_k) - \bar{\partial}u(t_k)$, $z_k := \bar{\partial}u(t_k) - \bar{\partial}P_R u(t_k)$. For all $\psi \in X^r$,

we have

$$\begin{aligned}
(\bar{\partial}\vartheta_k, \psi) + a(\vartheta_k, \psi) &= (\bar{\partial}u_r^k, \psi) + a(u_r^k, \psi) - (\bar{\partial}P_Ru(t_k), \psi) - a(P_Ru(t_k), \psi) \\
&= (f(t_k), \psi) - (F(u_r^k), \psi) - (G(u_r^k), \psi) - (\bar{\partial}P_Ru(t_k), \psi) - a(u(t_k), \psi) \\
&= (v_k, \psi) + (F(u(t_k)) - F(u_r^k), \psi) + (G(u(t_k)) - G(u_r^k), \psi).
\end{aligned}$$

By letting $\psi := \vartheta_k$ in the above equation, we get

$$\begin{aligned}
\|\vartheta_k\|^2 - (\vartheta_k, \vartheta_{k-1}) + \Delta t \nu \|\vartheta_k'\|^2 &= \Delta t (v_k, \vartheta_k) \\
&\quad + \Delta t (F(u(t_k)) - F(u_r^k), \vartheta_k) + \Delta t (G(u(t_k)) - G(u_r^k), \vartheta_k).
\end{aligned} \tag{4.30}$$

We now have to estimate the terms on the RHS of (4.30). To this end, we follow [137] (see page 139 in [137]). First, by using the Cauchy-Schwartz inequality, we get

$$(v_k, \vartheta_k) \leq \|v_k\| \|\vartheta_k\|. \tag{4.31}$$

Then, exactly as in [137], we get

$$|(F(u(t_k)) - F(u_r^k), \vartheta_k)| \leq \nu \|\vartheta_k\|_1^2 + C_9 \|\vartheta_k\|^2 + C_{10} \|\varrho_k\|_1^2. \tag{4.32}$$

The last term on the RHS of (4.30), however, needs to be treated differently, since it did not show up in the analysis in [137]. We start by adding and subtracting terms

$$\begin{aligned}
(G(u(t_k)) - G(u_r^k), \vartheta_k) &= (G(u(t_k)) - G(P_Ru(t_k)), \vartheta_k) \\
&\quad + (G(P_Ru(t_k)) - G(u_r^k), \vartheta_k).
\end{aligned} \tag{4.33}$$

We now move the last term on the RHS of (4.33) to the LHS of (4.30) (and thus change its sign) and apply the strong monotonicity property of G (4.13) to obtain

$$(G(u_r^k) - G(P_Ru(t_k)), \vartheta_k) \geq C_{11} \|\vartheta_k'\|_{L^3}^3. \tag{4.34}$$

For the first term on the RHS of (4.33), we need to apply the Lipschitz continuity of G (4.14):

$$(G(u(t_k)) - G(P_Ru(t_k)), \vartheta_k) \leq C \widetilde{M} \|\rho_k'\|_{L^3} \|\vartheta_k'\|_{L^3}, \tag{4.35}$$

where $\widetilde{M} = \max\{\|u'(t_k)\|_{L^3}, \|P_R u'(t_k)\|_{L^3}\}$. Here we have used the stability of $u'(t_k)$ and $P_R u'(t_k)$ in the L^3 norm, which we proved in Lemma 3.

To estimate the term on the RHS of (4.35), we use the inverse estimates in Lemma 2 and the Cauchy-Schwartz inequality:

$$\|\rho'_k\|_{L^3} \|\vartheta'_k\|_{L^3} \leq C_3^2 \|\rho'_k\| \|\vartheta'_k\| \leq \frac{\nu \|\vartheta'_k\|^2}{2} + \frac{1}{2\nu} C_3^4 \|\rho'_k\|^2. \quad (4.36)$$

The rest of the proof now follows exactly as in [137]. Specifically, by using (4.31)-(4.36) in (4.30), by choosing Δt sufficiently small, and by summing over k , we get

$$\|\vartheta'_k\|^2 \leq e^{C_{12}T} \left(\|\vartheta'_0\|^2 + \Delta t \sum_{j=1}^k (\|v_j\|^2 + C_{13} \|\rho\|_1^2) \right) \quad (4.37)$$

By using (4.29) and the estimates for v_k in the proof of Theorem 7 in [137], we get (4.26). \square

4.2.2 Numerical Results

The goal of this section is twofold: First, it will provide numerical evidence for the improved performance of the AV-POD-ROM (4.9) over the simple POD-G-ROM (4.7). Second, the numerical results will illustrate the theoretical estimate proved in Theorem 1. Our numerical study is presented in the context of Burgers equation. We emphasize again that Burgers equation (4.1), while being commonly used as a 1D approximation of the NSE, does not model turbulence. We use it, however, for simplicity and clarity of exposition.

The following numerical discretization has been used in all our numerical tests. We used a finite element discretization in space. The computational domain $\Omega = [0, 1]$ was uniformly discretized and the piecewise linear finite element space was used. We used backward Euler for the temporal discretization. The nonlinearity in (4.1) was treated with a Newton iteration that was stopped when the norm of the difference between two consecutive iterations was less than 10^{-8} . All computations were carried out using MATLAB.

We consider two test problems. Both problems develop shock-like phenomena that are challenging to capture with the usual POD-G-ROM (4.7). For both test problems we provide graphical illustrations of the improvement in accuracy for the AV-POD-ROM (4.9) over the usual POD-G-ROM (4.7). To illustrate the theoretical estimate in Theorem 1, we also present tables with error convergence rates with respect to the number of POD modes.

Experiment 1 This test problem is similar to that used in [137]. The initial condition in (4.1) is

$$u_0(x) = \begin{cases} 1 & \text{if } x \in (0, \frac{1}{2}] \\ 0 & \text{if } x \in (\frac{1}{2}, 1) . \end{cases} \quad (4.38)$$

The boundary conditions in (4.1) are homogeneous Dirichlet $u(0, t) = u(1, t) = 0$ for all t , the forcing term is $f = 0$, and the time interval is $[0, T] = [0, 1]$. The following parameters were used in the numerical discretization: Number of equally spaced mesh-points $N = 2048$ (which implies a mesh-size $h = 1/2048$); time-step $\Delta t = 10^{-3}$; diffusion parameter $\nu = 10^{-5}$; number of snapshots $M = 1000$. We deliberately chose a small value for ν , since this would be similar to choosing a large Reynolds number in the NSE. We also chose relatively small values for Δt and h , since the main goal of this study is to investigate the contribution to the total error of the POD truncation and modeling, and not the contribution of spatial and temporal discretizations.

We first ran a DNS. Indeed, given the spatial and temporal resolutions used, we can safely consider the results of the Galerkin finite element discretization in Figure 4.1 as our benchmark solution. The 1000 snapshots were used to generate the POD modes. We followed [137] and we did not include $\bar{\partial}u(t_k)$, since we used a small time-step Δt . The first 20 POD modes were used to run (i) the POD-G-ROM (4.7); and (ii) the AV-POD-ROM (4.9) with $c = 10^{-4}$.

The time evolutions of the two models are presented in Figures 4.2–4.3. Notice the clear improvement in solution accuracy in the AV-POD-ROM. Indeed, the large numerical oscillations of the POD-G-ROM are dramatically decreased by the AV-POD-ROM. A better visualization of this improvement is given in Figure 4.4, where results from both POD-ROMs are compared with the “truth” (DNS) solution at the final time $T = 1$.

A quantitative illustration of the theoretical estimate (4.26) in Theorem 1 is given in Table 4.1. We present the L^2 norms of the errors for both models - the POD-G-ROM (4.7) (third column) and the AV-POD-ROM (4.9) (fourth column). As noticed earlier, the spatial and temporal discretizations account only for a small percentage of the total error. Indeed, since $\Delta t = 10^{-3}$ and $h = 1/2048$, the first and third terms on the right-hand side of the

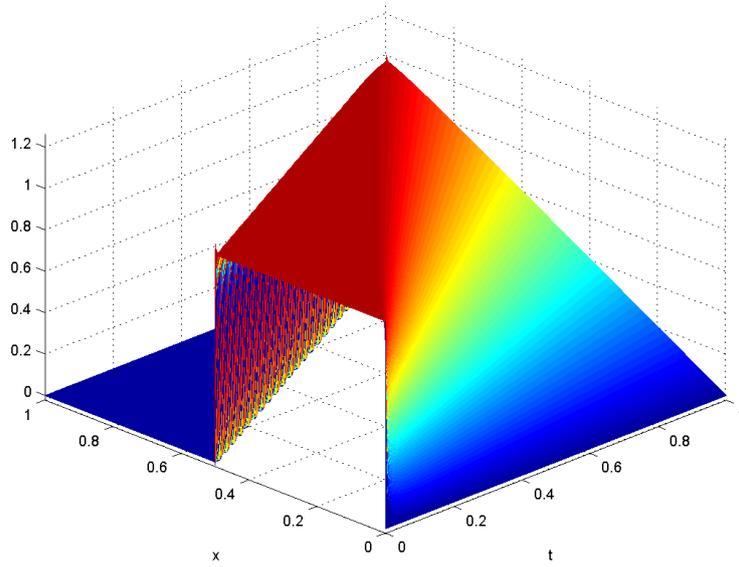


Figure 4.1: DNS solution in Experiment 1.

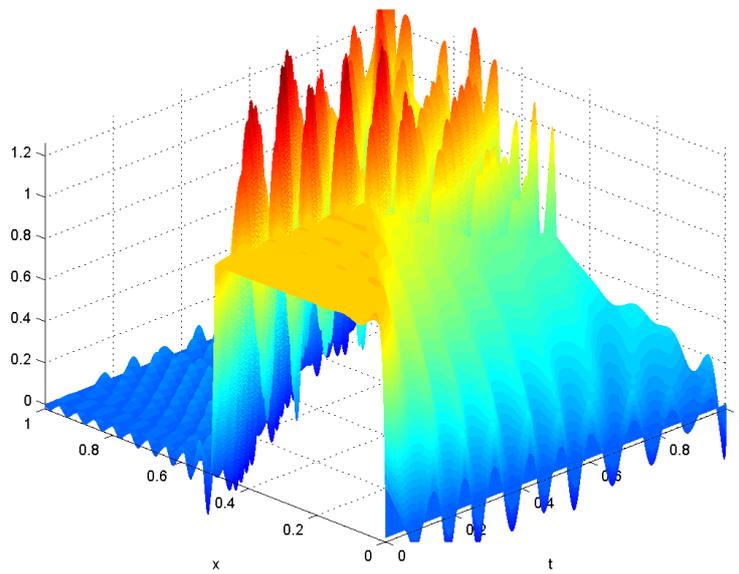


Figure 4.2: POD-G-ROM (4.7) solution in Experiment 1.

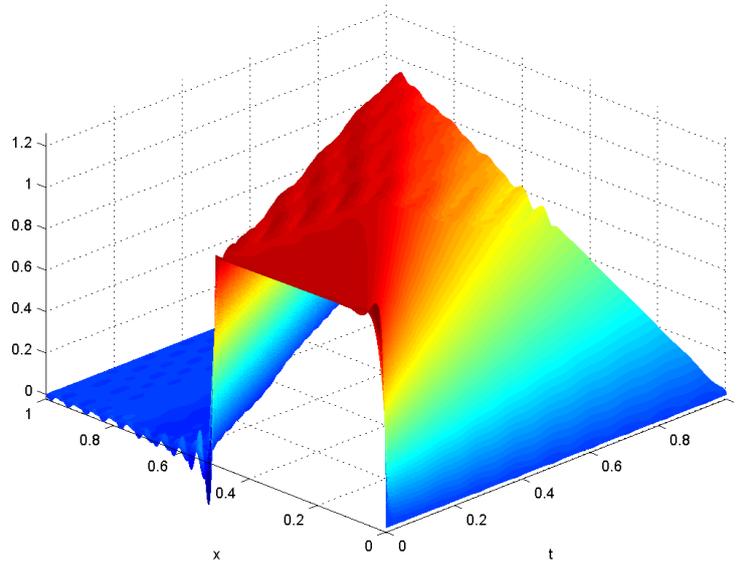


Figure 4.3: AV-POD-ROM (4.9) solution in Experiment 1.

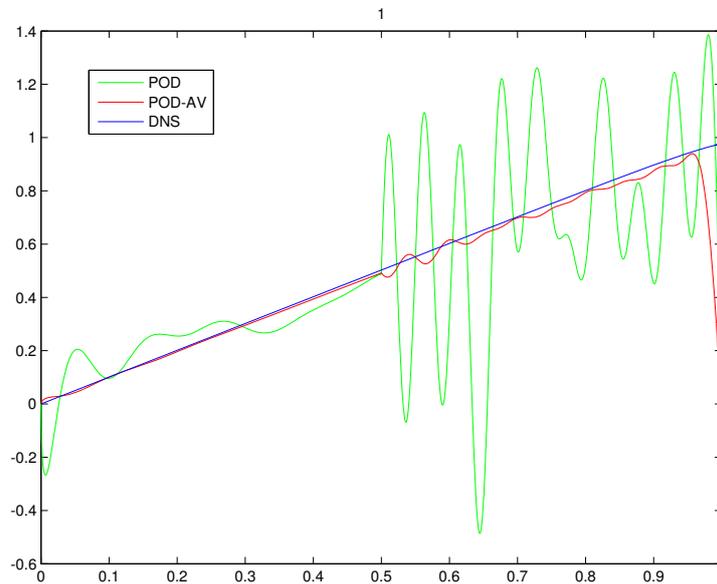


Figure 4.4: Snapshots of DNS, POD-G-ROM and AV-POD-ROM at the final time $T = 1$ in Experiment 1.

r	Normalized energy $\frac{\sum_{i=r}^{1001} \lambda_i}{\sum_{i=1}^{1001} \lambda_i}$	POD	AV-POD-ROM
6	0.0223	0.0778	0.0305
11	0.0111	0.0943	0.0090
20	0.0056	0.0428	0.0053
36	0.0028	0.0130	0.0051
62	0.0014	0.0038	0.0050

Table 4.1: Average errors for POD-G-ROM (4.7) and AV-POD-ROM (4.9) in Experiment 1.

inequality (4.26) in Theorem 1 are much smaller than the second term, which represents the POD truncation and modeling errors. The number of POD modes (first column) was chosen so that the POD truncation error (second column) decrease by a factor of 2. As indicated by the theoretical estimate in [137], the square of the error of the POD-G-ROM should decrease by a factor of 4. This behavior is observed in the third column of Table 4.1 (for $r \geq 20$). The numerical results for the AV-POD-ROM (in the fourth column) are, as expected, much better than those of the POD-G-ROM (third column). They, however, do not seem to follow the theoretical estimate (4.26) in Theorem 1. We think that this is due to the fact that the POD truncation and modeling error in the AV-POD-ROM initially dominates the spatial and temporal discretization errors, but once it reaches a certain threshold, the latter become significant.

Experiment 2 This test problem is similar to that used in [165]. We simulate the propagation of an initial Gaussian disturbance given by

$$u_0(x) = \frac{10}{5\sqrt{\pi}} \exp\left(-\frac{(x-50)^2}{\omega}\right), \quad (4.39)$$

where $\omega = 400$ is the initial width of the wave and $t \in [0, 600]$. We rescale parameters so that $x \in [0, 1]$. Thus, the time interval becomes $[0, 1.5]$ and $\nu = 0.015/400 = 3.75 \times 10^{-5}$.

The boundary conditions in (4.1) are homogeneous Dirichlet $u(0, t) = u(1, t) = 0$ for all t and the forcing term is $f = 0$. The following parameters were used in the numerical

discretization: Number of equally spaced mesh-points $N = 2048$ (which implies a mesh-size $h = 1/2048$); time-step $\Delta t = 10^{-3}$; number of snapshots $M = 1000$. As in the first numerical example, we again chose a small value for ν and relatively small values for Δt and h .

We first ran a DNS, which we consider as our benchmark solution. The time evolution of the DNS solution is shown in Figure 4.5. The 1000 snapshots were used to generate the POD modes. We followed [137] and we did not include $\bar{\partial}u(t_k)$ since we used a small time-step Δt . The first 20 POD modes are used to run (i) the POD-G-ROM (4.7); and (ii) the AV-POD-ROM (4.9) with $c = 5 \times 10^{-5}$.

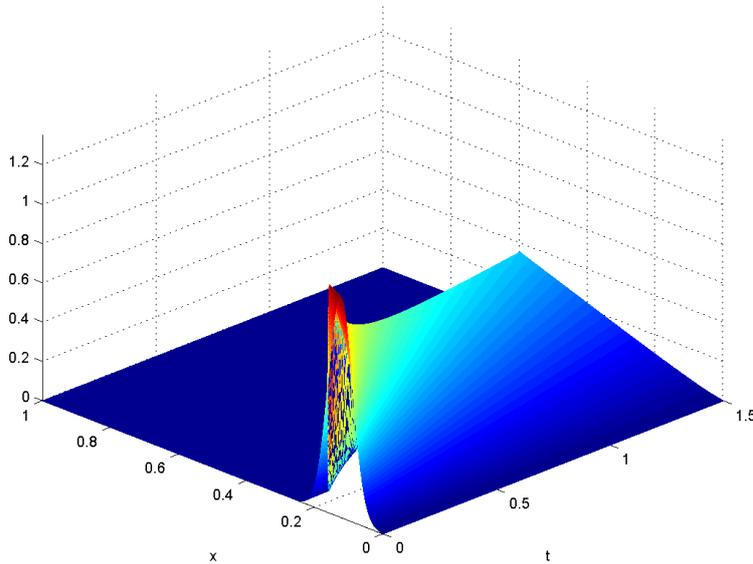


Figure 4.5: DNS solution in Experiment 2.

The time evolutions of the two models are presented in Figures 4.6–4.7. As in Experiment 1, there is a clear improvement in solution accuracy in the AV-POD-ROM. Indeed, the large numerical oscillations of the POD-G-ROM are dramatically decreased by the AV-POD-ROM (see also Figure 4.8).

To illustrate the theoretical estimate (4.26) in Theorem 1, we present in Table 4.2 the L^2 norms of the errors for the POD-G-ROM (4.7) (third column) and the AV-POD-ROM (4.9) (fourth column). As in Experiment 1, we notice that the numerical results for the

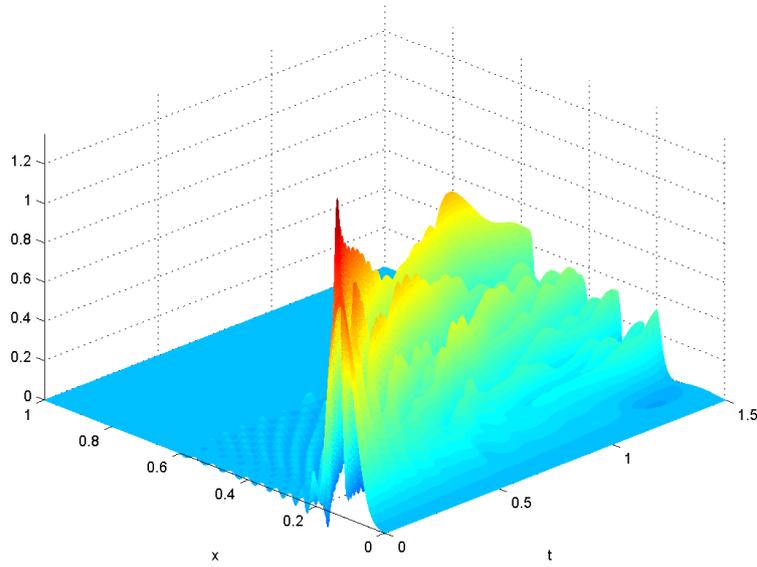


Figure 4.6: POD-G-ROM (4.7) solution in Experiment 2.

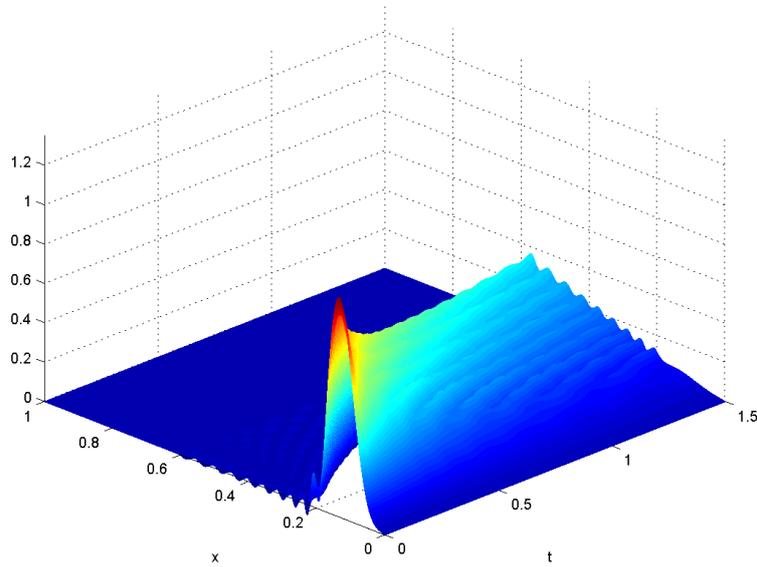


Figure 4.7: AV-POD-ROM (4.9) solution in Experiment 2.

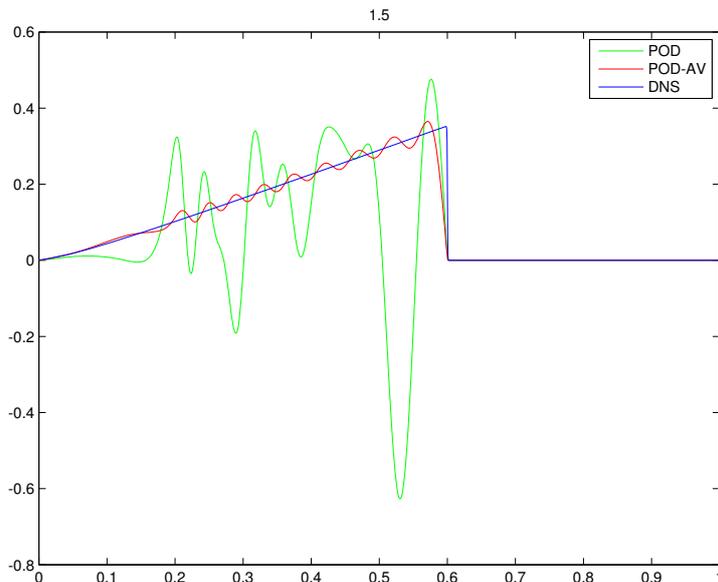


Figure 4.8: Snapshots of DNS, POD-G-ROM and AV-POD-ROM at the final time $T = 1.5$ in Experiment 2.

r	Normalized energy $\frac{\sum_{i=r}^{858} \lambda_i}{\sum_{i=1}^{858} \lambda_i}$	POD	AV-POD-ROM
6	0.0702	0.0358	0.0118
11	0.0345	0.0562	0.0022
20	0.0167	0.0263	8.78e-4
34	0.0085	0.0056	7.56e-4
56	0.0042	0.0009	7.44e-4

Table 4.2: Average errors for POD-G-ROM (4.7) and AV-POD-ROM (4.9) in Experiment 2.

AV-POD-ROM are much better than those for the POD-G-ROM. We also notice that the error for the POD-G-ROM in the third column of Table 4.2 follows the theoretical estimates in [137] (for $r \geq 20$). Although the numerical results for the AV-POD-ROM are much better than those of the POD-G-ROM, they do not seem to follow the theoretical estimate (4.26) in Theorem 1. We think that this behavior is due to the fact that for $r \geq 20$, the spatial

and temporal discretization errors have the same order of magnitude as the POD truncation and modeling error.

4.3 Variational Multiscale POD-ROM

In this section, the new VMS-POD-ROM is analyzed and tested in the numerical approximation of a *convection-dominated convection-diffusion-reaction* problem

$$\begin{cases} u_t - \varepsilon \Delta u + \mathbf{b} \cdot \nabla u + gu = f & \text{in } (0, T] \times \Omega, \\ u(x, 0) = u_0(x) & \text{in } \Omega, \\ u(x, t) = 0 & \text{on } (0, T] \times \partial\Omega, \end{cases} \quad (4.40)$$

where $\varepsilon \ll 1$ is the diffusion parameter, \mathbf{b} with $\|\mathbf{b}\| = \mathcal{O}(1)$ the given convective field, f the forcing term, $\Omega \subset \mathbb{R}^2$ the computational domain, $t \in [0, T]$, with T the final time, and $u_0(\cdot)$ the initial condition. Without loss of generality, we assume in the sequel that the boundary conditions are homogeneous Dirichlet.

The POD Galerkin truncation is the approximation $u_r \in X^r$ (1.3) of u :

$$u_r := \sum_{j=1}^r u_j \varphi_j. \quad (4.41)$$

Plugging (4.41) into (4.40) and multiplying by test functions in $X := H_0^1(\Omega)$ yield the POD-G-ROM

$$(u_{r,t}, v_r) + \varepsilon(\nabla u_r, \nabla v_r) + (\mathbf{b} \cdot \nabla u_r, v_r) + (gu_r, v_r) = (f, v_r) \quad \forall v_r \in X^r. \quad (4.42)$$

The main advantage of the POD-G-ROM (4.42) over a straightforward finite element discretization of (4.40) is clear - since $\dim X^r \ll \dim X$, the computational cost of the former is dramatically lower than that of the latter. There are, however, several well-documented disadvantages of (4.42), such as its numerical instability in convection-dominated flows [205]. To address this issue, we draw inspiration from the methodologies used in increasing the numerical stability of finite element discretizations of such flows.

The VMS method introduced by Hughes and his group [112, 113, 114, 115] has been successful in the numerical discretization of turbulent flows [92, 93, 94, 95, 106, 121, 122, 123,

124, 125, 126, 127]. The idea in VMS is straightforward: Instead of adding artificial viscosity to all resolved scales, in VMS artificial viscosity is only added to the smallest resolved scales. Thus, the small scale oscillations are eliminated without polluting the large scale components of the approximation. The VMS method has been extensively developed, and various numerical methods were used. The finite element discretization of the resulting VMS model has evolved in several directions: Hughes and his group proposed a VMS formulation for the NSE in which a Smagorinsky model [32, 211] was added only to the smallest resolved scales [112, 113, 114, 115]. A different type of VMS approach, based on the residual of the NSE, was proposed in Bazilevs et al. [26]. One of the earliest VMS ideas for convection-dominated convection-diffusion-reaction equations was proposed by Guermond in [98, 99]. In this VMS formulation, the smallest scales were modeled by using finite element spaces enriched with bubble functions. Layton proposed in [144] a VMS approach similar to that of Guermond. In this VMS approach, however, the smallest resolved scales were modeled by projection on a coarser mesh. The VMS approach proposed in [144] was extended to the NSE in a sequence of papers by John and Kaya [121, 122, 123, 124]. The variational formulation used by the finite element methodology fits very well with the VMS approach. The definition of the smallest resolved scales, however, often poses many challenges to the finite element method. Indeed, one needs to enrich the finite element spaces with bubble functions [98, 99], consider hierarchical finite element bases [113], or use a projection on a coarser mesh [144].

POD represents the perfect setting for the VMS methodology, since the hierarchy of the basis is already present. Indeed, the POD basis functions are already listed in descending order of the kinetic energy content. Based on this simple observation, we next propose a new VMS based POD-ROM. To this end, we consider the following spaces: $X^h \subset H_0^1(\Omega)$, $X^R := \text{span}\{\varphi_1, \varphi_2, \dots, \varphi_R\}$, where $R < r$, and L^R , where L^R will be defined later. Note that $X^R \subseteq X^r \subset X^h \subset X$. We also consider $P_R : L^2(\Omega) \rightarrow L^R$ be the orthogonal projection of $L^2(\Omega)$ on L^R , defined by

$$(u - P_R u, v_R) = 0 \quad \forall v_R \in L^R. \quad (4.43)$$

Let also $P'_R := \mathbb{I} - P_R$. We are now ready to define the new *Variational Multiscale Proper*

Orthogonal Decomposition model:

$$\begin{aligned} (u_{r,t}, v_r) + \varepsilon (\nabla u_r, \nabla v_r) + \alpha (P'_R \nabla u_r, P'_R \nabla v_r) \\ + (\mathbf{b} \cdot \nabla u_r, v_r) + (g u_r, v_r) = (f, v_r) \quad \forall v_r \in X^r. \end{aligned} \quad (4.44)$$

The third term on the LHS of (4.44) represents the artificial viscosity that is added only to the smallest resolved scales of the gradient. We note that the POD-ROM (4.44), which we denote *P_R-VMS-POD-ROM*, is different from VMS-POD-ROM (2.25). To our knowledge this is the first time that the VMS formulation in [144] is applied in a POD setting.

In the next two sections, we will first estimate the error made in the finite element discretization of the P_R-VMS-POD-ROM (4.44) and then use it in a numerical test.

4.3.1 Error Estimates

In this section, we prove estimates for the average error $\frac{1}{M+1} \sum_{k=0}^M \|u^k - u_r^k\|$, where the approximation u^k is the solution of (4.55) (the weak form of (4.40)) and u_r^k is the solution of (4.56) (the finite element discretization of the P_R-VMS-POD-ROM (4.44)). To this end, we follow the approach in [106] (see also [131]). We emphasize, however, that our presentation is different in that it has to include several results pertaining to the POD setting. To this end, we use some of the developments in [158] (see also [109, 137, 155, 203]).

We start by introducing some notation and we list several results that will be used throughout this section. For clarity of notation, we will denote by C a generic constant that can depend on all the parameters in the system, except on d (the rank of the snapshot set), M (the number of snapshots), r (the number of POD modes used in the POD-G-ROM (4.42)), R (the number of POD modes used in the projection operator in the P_R-VMS-POD-ROM (4.44)), h (the mesh-size in the finite element discretization), α (the artificial viscosity coefficient), and ε (the diffusion coefficient). Of particular interest is the independence of the generic constant C from ε . Indeed, we will prove estimates that are *uniform with respect to ε* , which is important when convection-dominated flows (such as the NSE) are considered.

We introduce the bilinear forms $b(u, v) := (\mathbf{b} \cdot \nabla u, v) + (g u, v)$, $a(u, v) := \varepsilon (\nabla u, \nabla v) + b(u, v)$, and $A(u, v) := a(u, v) + \alpha (P'_R \nabla u, P'_R \nabla v)$. We also consider the weighted norm $\|u\|_{a,b,\alpha}^2 := a \|u\|^2 + b \|\nabla u\|^2 + \alpha \|P'_R \nabla u\|^2$. We now make the following assumption, which is

used in proving the well-posedness of the weak formulation of (4.40).

Assumption 1 (Coercivity and Continuity)

$$g - \frac{1}{2} \nabla \cdot \mathbf{b} \geq \beta > 0 \quad \text{and} \quad \max\{\|g\|, \|\mathbf{b}\|\} = \gamma > 0. \quad (4.45)$$

For the finite element discretization of (4.40), we consider a family of finite dimensional subspaces X^h of $X = H_0^1(\Omega)$, such that, for all $v \in H^{m+1} \cap X$, the following assumption is satisfied.

Assumption 2 (Approximability)

$$\inf_{v_h \in X^h} \{\|v - v_h\| + h \|\nabla v - \nabla v_h\|\} \leq C h^{m+1} \|v\|_{m+1} \quad 1 \leq m \leq K, \quad (4.46)$$

where K is the order of accuracy of $\{X^h\}$. We also assume that the finite element spaces $\{X^h\}$ satisfy the following inverse estimate.

Assumption 3 (finite element Inverse Estimate)

$$\|\nabla v_h\| \leq C h^{-1} \|v_h\| \quad \forall v_h \in X^h. \quad (4.47)$$

A similar inverse estimate for POD is proven in [137]. For completeness, we present it below. We also include a new estimate and present its proof.

Lemma 4 (POD Inverse Estimate) *Let $M_r \in \mathbb{R}^{r \times r}$ with $M_{ij} = (\varphi_j, \varphi_i)$ be the POD mass matrix, $H_r \in \mathbb{R}^{r \times r}$ with $H_{ij} = (\nabla \varphi_j, \nabla \varphi_i)$ be the POD stiffness matrix, $S_r \in \mathbb{R}^{r \times r}$ with $S_{ij} = (\varphi_j, \varphi_i)_{H^1}$ be the POD mass matrix in the H^1 -norm, and $\|\cdot\|_2$ denote the matrix 2-norm. Then, for all $v_r \in X^r$, the following estimates hold.*

$$\|v_r\|_{L^2} \leq \sqrt{\|M_r\|_2 \|S_r^{-1}\|_2} \|v_r\|_{H^1}, \quad (4.48)$$

$$\|v_r\|_{H^1} \leq \sqrt{\|S_r\|_2 \|M_r^{-1}\|_2} \|v_r\|_{L^2}, \quad (4.49)$$

$$\|\nabla v_r\|_{L^2} \leq \sqrt{\|H_r\|_2 \|M_r^{-1}\|_2} \|v_r\|_{L^2}. \quad (4.50)$$

Proof: The proof of estimates (4.48) and (4.49) was given in [137] (see Lemma 2 and Remark 2). The proof of (4.50) follows along the same lines: Let $v_r = \sum_{i=1}^r x_j \varphi_j$ and $\mathbf{x} = (x_1, \dots, x_r)^T$. From the definition of H_r , it follows that $\|\nabla v_r\|_{L^2}^2 = \mathbf{x}^T H_r \mathbf{x}$. Since H_r is symmetric, its matrix 2-norm is equal to its Rayleigh quotient [68]: $\|H_r\|_2 = \max_{\mathbf{x} \neq \mathbf{0}} \frac{\mathbf{x}^T H_r \mathbf{x}}{\mathbf{x}^T \mathbf{x}}$. Thus, we get:

$$\|\nabla v_r\|_{L^2}^2 = \mathbf{x}^T H_r \mathbf{x} \leq \|H_r\|_2 \mathbf{x}^T \mathbf{x}. \quad (4.51)$$

Furthermore, since M_r^{-1} is also symmetric, we get $\mathbf{y}^T M_r^{-1} \mathbf{y} \leq \|M_r^{-1}\|_2 \mathbf{y}^T \mathbf{y}$ for all vectors $\mathbf{y} \in \mathbb{R}^r$. We also note that, since M_r is symmetric positive definite, we can use its Cholesky decomposition $M_r = L_r L_r^T$, where L_r is a lower triangular nonsingular matrix [68]. Thus, letting $\mathbf{y} = L_r \mathbf{x}$, we get:

$$\|M_r^{-1}\|_2 \geq \frac{\mathbf{y}^T M_r^{-1} \mathbf{y}}{\mathbf{y}^T \mathbf{y}} = \frac{\mathbf{x}^T L_r^T (L_r^{-1})^T L_r^{-1} L_r \mathbf{x}}{\mathbf{x}^T L^T L \mathbf{x}} = \frac{\mathbf{x}^T \mathbf{x}}{\mathbf{x}^T M_r \mathbf{x}}. \quad (4.52)$$

Inequalities (4.51) and (4.52) imply the following inequality, which proves (4.50): $\|\nabla v_r\|_{L^2}^2 \leq \|H_r\|_2 \|M_r^{-1}\|_2 \mathbf{x}^T M_r \mathbf{x} = \|H_r\|_2 \|M_r^{-1}\|_2 \|v_r\|_{L^2}^2$.

Remark 6 We note that, in our setting, (4.50) can be improved. Indeed, since S_r is the identity matrix when $X = H_0^1$, we get:

$$\|\nabla v_r\|_{L^2} \leq \|v_r\|_{H^1} \leq \sqrt{\|S_r\|_2 \|M_r^{-1}\|_2} \|v_r\|_{L^2} = \sqrt{\|M_r^{-1}\|_2} \|v_r\|_{L^2}. \quad (4.53)$$

We note, however, that in general (4.53) might not hold.

To prove optimal error estimates in time, we follow [137] and include the finite difference quotients $\bar{\partial}u(t_k) = \frac{u(t_k) - u(t_{k-1})}{\Delta t}$, where $k = 1, \dots, M$, in the set of snapshots $V := \text{span}\{u(t_0), \dots, u(t_M), \bar{\partial}u(t_1), \dots, \bar{\partial}u(t_M)\}$. As pointed out in [137], the error formula (1.9) becomes:

$$\begin{aligned} & \frac{1}{2M+1} \sum_{k=0}^M \left\| u(\cdot, t_k) - \sum_{j=1}^r (u(\cdot, t_k), \varphi_j(\cdot))_X \varphi_j(\cdot) \right\|_X^2 \\ & + \frac{1}{2M+1} \sum_{k=1}^M \left\| \bar{\partial}u(\cdot, t_k) - \sum_{j=1}^r (\bar{\partial}u(\cdot, t_k), \varphi_j(\cdot))_X \varphi_j(\cdot) \right\|_X^2 = \sum_{j=r+1}^d \lambda_j. \end{aligned} \quad (4.54)$$

After these preliminaries, we are ready to derive the error estimates.

The weak form of (4.40) reads:

$$(u_t, v) + a(u, v) = (f, v) \quad \forall v \in X. \quad (4.55)$$

The finite element spatial discretization and backward Euler time discretization of the P_R-VMS-POD-ROM (4.55) reads: Find $u_r^k \in X^r$ such that:

$$\frac{1}{\Delta t} (u_r^{k+1} - u_r^k, v_r) + A(u_r^{k+1}, v_r) = (f^{k+1}, v_r) \quad \forall v_r \in X^r. \quad (4.56)$$

The following stability result for u_r^k holds:

Theorem 2 *The solution u_r^k of (4.56) satisfies the following bound*

$$\|u_r^k\| \leq \|u_r^0\| + \Delta t \sum_{k=0}^{M-1} \|f^{k+1}\|. \quad (4.57)$$

Proof: Choosing $v_r := u_r^{k+1}$ in (4.56), we get:

$$\frac{1}{\Delta t} (u_r^{k+1} - u_r^k, u_r^{k+1}) + A(u_r^{k+1}, u_r^{k+1}) = (f^{k+1}, u_r^{k+1}). \quad (4.58)$$

By applying the Cauchy-Schwarz inequality on both sides of (4.58) and simplifying by $\|u_r^{k+1}\|$, we get:

$$\|u_r^{k+1}\| - \|u_r^k\| \leq \Delta t \|f^{k+1}\|. \quad (4.59)$$

Summing from 0 to $M - 1$ the inequality in (4.59), we get (4.57).

In order to prove an estimate for $\|u^k - u_r^k\|$, we will first consider the *Ritz projection* $w_r \in X^r$ of $u \in X$:

$$A(u - w_r, v_r) = 0 \quad \forall v_r \in X^r. \quad (4.60)$$

The existence and uniqueness of w_r follow from Lax-Milgram lemma. We now prove an estimate for $u^k - w_r^k$, the error in the Ritz projection.

Lemma 5 *The Ritz projection w_r^k of u^k satisfies the following error estimate:*

$$\begin{aligned} \frac{1}{M} \sum_{k=1}^M \|u^k - w_r^k\| \leq C \left\{ \left(1 + \sqrt{\|M_r^{-1}\|_2} + \alpha^{-1}\right)^{1/2} \right. \\ \left. \left(h^{m+1} \frac{1}{M} \sum_{k=1}^M \|u^k\|_{m+1} + \sqrt{\sum_{j=r+1}^d \lambda_j} \right) \right. \\ \left. + \sqrt{\varepsilon + \alpha} \left(h^m \frac{1}{M} \sum_{k=1}^M \|u^k\|_{m+1} + \sqrt{\sum_{j=r+1}^d \lambda_j} \right) \right\}. \end{aligned} \quad (4.61)$$

Proof: Setting $u := u^k$ in (4.60), we get:

$$A(u^k - w_r^k, v_r) = 0 \quad \forall v_r \in X^r. \quad (4.62)$$

We decompose the error $u^k - w_r^k$ as $u^k - w_r^k := (u^k - I_{h,r}(u^k)) - (w_r^k - I_{h,r}(u^k)) = \eta^k - \varphi_r^k$, where $I_{h,r}(u^k)$ is the interpolant of u^k in the space X^r . By the triangle inequality, we have:

$$\frac{1}{M} \sum_{k=1}^M \|u^k - w_r^k\| \leq \frac{1}{M} \sum_{k=1}^M \|\eta^k\| + \frac{1}{M} \sum_{k=1}^M \|\varphi_r^k\|. \quad (4.63)$$

We start by estimating $\|\eta^k\|$. We note that $I_{h,r}(u^k)$ consists of two parts: We first consider u_h^k , the finite element solution of (4.40), which yielded the ensemble of snapshots. Then, we interpolate u_h^k in X^r , which yields $I_{h,r}(u^k)$. Note that this is different from [106], where only the first part was present (see (8) in [106]).

$$\begin{aligned} \frac{1}{M} \sum_{k=1}^M \|\eta^k\| &= \frac{1}{M} \sum_{k=1}^M \|u^k - I_{h,r}(u^k)\| \\ &\leq \frac{1}{M} \sum_{k=1}^M \|u^k - u_h^k\| + \frac{1}{M} \sum_{k=1}^M \|u_h^k - I_{h,r}(u^k)\|. \end{aligned} \quad (4.64)$$

Using Assumption 2, it is easily shown [185] that:

$$\frac{1}{M} \sum_{k=1}^M \|u^k - u_h^k\| \leq C h^{m+1} \frac{1}{M} \sum_{k=1}^M \|u^k\|_{m+1}. \quad (4.65)$$

Picking $I_{h,r}(u^k) := \sum_{j=1}^r (u^k, \varphi_j)_X \varphi_j$ in the last term on the RHS of (4.64) and then using (4.54), we get:

$$\frac{1}{M} \sum_{k=1}^M \|u_h^k - I_{h,r}(u^k)\| \leq \sqrt{\sum_{j=r+1}^d \lambda_j}. \quad (4.66)$$

Note that we consider that the time instances $t_k = k \Delta t$ in the time discretization (4.56) are the same as the time instances at which the snapshots were taken. If this is not the case, one should use a Taylor series approach (see (4.8) in [158]).

Plugging (4.65) and (4.66) in (4.64), we get:

$$\frac{1}{M} \sum_{k=1}^M \|\eta^k\| \leq C h^{m+1} \frac{1}{M} \sum_{k=1}^M \|u^k\|_{m+1} + \sqrt{\sum_{j=r+1}^d \lambda_j}. \quad (4.67)$$

Similarly, using that $X = H_0^1$ in (4.54), we get:

$$\frac{1}{M} \sum_{k=1}^M \|\nabla \eta^k\| \leq C h^m \frac{1}{M} \sum_{k=1}^M \|u^k\|_{m+1} + \sqrt{\sum_{j=r+1}^d \lambda_j}. \quad (4.68)$$

Equation (4.62) implies:

$$A(u^k - w_r^k, v_r) = A(\eta^k - \varphi_r^k, v_r) = 0. \quad (4.69)$$

Choosing $v_r = \varphi_r^k$ in (4.69) implies:

$$A(\varphi_r^k, \varphi_r^k) = A(\eta^k, \varphi_r^k). \quad (4.70)$$

We decompose the bilinear form A into its symmetric and skew-symmetric parts: $A := A_s + A_{ss}$, where $A_s(u, v) := \alpha (P'_R \nabla u, P'_R \nabla v) + \varepsilon (\nabla u, \nabla v) + ((g - \frac{1}{2} \nabla \cdot \mathbf{b}) u, v)$, and $A_{ss}(u, v) := (\mathbf{b} \cdot \nabla u + \frac{1}{2} (\nabla \cdot \mathbf{b}) u, v)$. Equation (4.70) implies:

$$A_s(\varphi_r^k, \varphi_r^k) + \cancel{A_{ss}(\varphi_r^k, \varphi_r^k)} \xrightarrow{0} = A_s(\eta^k, \varphi_r^k) + A_{ss}(\eta^k, \varphi_r^k). \quad (4.71)$$

Assumption 1 implies that $A_s(\varphi_r^k, \varphi_r^k) \geq C \|\varphi_r^k\|_{1,\varepsilon,\alpha}^2$. Thus, using the Cauchy-Schwarz and Young's inequalities, (4.71) becomes:

$$\begin{aligned} C \|\varphi_r^k\|_{1,\varepsilon,\alpha}^2 &\leq A_s(\varphi_r^k, \varphi_r^k)^{1/2} A_s(\eta^k, \eta^k)^{1/2} + A_{ss}(\eta^k, \varphi_r^k) \\ &\leq \frac{1}{2} A_s(\varphi_r^k, \varphi_r^k) + \frac{1}{2} A_s(\eta^k, \eta^k) + (\mathbf{b} \eta^k, \nabla \varphi_r^k) + \frac{1}{2} ((\nabla \cdot \mathbf{b}) \eta^k, \varphi_r^k). \end{aligned} \quad (4.72)$$

Rearranging and using Assumption 1, (4.72) becomes:

$$C \|\varphi_r^k\|_{1,\varepsilon,\alpha}^2 \leq C \left(|A_s(\eta^k, \eta^k)| + |(\mathbf{b} \eta^k, \nabla \varphi_r^k)| + |((\nabla \cdot \mathbf{b}) \eta^k, \varphi_r^k)| \right). \quad (4.73)$$

We now estimate each term on the RHS of (4.73).

$$\begin{aligned} |A_s(\eta^k, \eta^k)| &= \varepsilon \|\nabla \eta^k\|^2 + \left(\left(g - \frac{1}{2} \nabla \cdot \mathbf{b} \right) \eta^k, \eta^k \right) \\ &\quad + \alpha \|P'_R \nabla \eta^k\|^2 \leq C \|\eta^k\|_{1,\varepsilon,\alpha}^2. \end{aligned} \quad (4.74)$$

To estimate the second term on the RHS of (4.73), we first note that $\|P_R\| \leq 1$ (since P_R is L^2 -projection) and use the inverse estimate (4.49) in Lemma 4 to obtain:

$$\|P_R(\nabla \varphi_r^k)\| \leq \|\nabla \varphi_r^k\| \leq \|\varphi_r^k\|_{H^1} \leq \sqrt{\|M_r^{-1}\|_2} \|\varphi_r^k\|. \quad (4.75)$$

Using that $(P_R u, P'_R v) = 0 \forall u, v$, the Cauchy-Schwarz and Young's inequalities, and the inverse estimate (4.53), we then get:

$$\begin{aligned} |(\mathbf{b}\eta^k, \nabla \varphi_r^k)| &\leq |(P_R(\mathbf{b}\eta^k), P_R(\nabla \varphi_r^k))| + |(P'_R(\mathbf{b}\eta^k), P'_R(\nabla \varphi_r^k))| \\ &\leq \|P_R(\mathbf{b}\eta^k)\| \|P_R(\nabla \varphi_r^k)\| + \|P'_R(\mathbf{b}\eta^k)\| \|P'_R(\nabla \varphi_r^k)\| \\ &\leq C \sqrt{\|M_r^{-1}\|_2} \|P_R(\mathbf{b}\eta^k)\| \|\varphi_r^k\| + \|P'_R(\mathbf{b}\eta^k)\| \|P'_R(\nabla \varphi_r^k)\| \\ &\leq \left(\frac{1}{\beta} C \|M_r^{-1}\|_2 \|P_R(\mathbf{b}\eta^k)\|^2 + \frac{\beta}{4} \|\varphi_r^k\|^2 \right) \\ &\quad + \left(\frac{1}{2\alpha} \|P'_R(\mathbf{b}\eta^k)\|^2 + \frac{\alpha}{2} \|P'_R(\nabla \varphi_r^k)\|^2 \right). \end{aligned} \quad (4.76)$$

We note that this is exactly why we need the inverse estimate in Lemma 4: to absorb $\|\varphi_r^k\|^2$ in the LHS of (4.73). If we had used $\|\nabla \varphi_r^k\|^2$ instead, then we would have had to absorb it in $\varepsilon \|\nabla \varphi_r^k\|^2$ on the LHS, and so the RHS would have depended on ε . Finally, by using the Cauchy-Schwarz and Young's inequalities, the third term on the RHS of (4.73) can be estimated as follows:

$$|((\nabla \cdot \mathbf{b}) \eta^k, \varphi_r^k)| \leq C \|\eta^k\| \|\varphi_r^k\| \leq C \left(\frac{1}{\beta} \|\eta^k\|^2 + \frac{\beta}{4} \|\varphi_r^k\|^2 \right). \quad (4.77)$$

Collecting estimates (4.73), (4.74), (4.76) and (4.77), we get:

$$\begin{aligned} \|\varphi_r^k\|_{1,\varepsilon,\alpha}^2 &\leq C \left(\|\eta^k\|_{1,\varepsilon,\alpha}^2 + \frac{1}{\beta} \|M_r^{-1}\|_2 \|P_R(\mathbf{b}\eta^k)\|^2 \right. \\ &\quad \left. + \frac{1}{2\alpha} \|P'_R(\mathbf{b}\eta^k)\|^2 + \frac{1}{\beta} \|\eta^k\|^2 \right). \end{aligned} \quad (4.78)$$

The last term on the RHS of (4.78), can be absorbed in $C \|\eta^k\|_{1,\varepsilon,\alpha}^2$. Since $\|P_R\| \leq 1$ (P_R is L^2 -projection) and $\|\mathbf{b}\| \leq \gamma$ (by Assumption 1), we get:

$$\frac{1}{\beta} \|M_r^{-1}\|_2 \|P_R(\mathbf{b}\eta^k)\|^2 \leq C \|M_r^{-1}\|_2 \|\eta^k\|^2. \quad (4.79)$$

Since $\|P_R\| \leq 1$ (P_R is L^2 -projection) and $\|\mathbf{b}\| \leq \gamma$ (by Assumption 1), we get:

$$\frac{1}{2\alpha} \|P'_R(\mathbf{b}\eta^k)\|^2 \leq \frac{C}{\alpha} \|\eta^k\|^2. \quad (4.80)$$

Thus, using (4.79) and (4.80) in (4.78), we get:

$$\begin{aligned} \|\varphi_r^k\|_{1,\varepsilon,\alpha}^2 &\leq C \left(\|\eta^k\|^2 + \varepsilon \|\nabla\eta^k\|^2 + \alpha \|P'_R(\nabla\eta^k)\|^2 + C \|M_r^{-1}\|_2 \|\eta^k\|^2 \right. \\ &\quad \left. + \frac{1}{2\alpha} \|P'_R(\mathbf{b}\eta^k)\|^2 + \frac{C}{\alpha} \|\eta^k\|^2 \right). \end{aligned} \quad (4.81)$$

Since P_R is L^2 -projection, $\|P'_R\| \leq 1$, and thus the second term on the RHS of (4.81) can be bounded as follows: $\alpha \|P'_R(\nabla\eta^k)\|^2 \leq \alpha \|\nabla\eta^k\|^2$. Summing in (4.81), we get:

$$\begin{aligned} \frac{1}{M} \sum_{k=1}^M \|\varphi_r^k\|_{1,\varepsilon,\alpha}^2 &\leq C (1 + \|M_r^{-1}\|_2 + \alpha^{-1}) \frac{1}{M} \sum_{k=1}^M \|\eta^k\|^2 \\ &\quad + (\varepsilon + \alpha) \frac{1}{M} \sum_{k=1}^M \|\nabla\eta^k\|^2. \end{aligned} \quad (4.82)$$

Using (4.67) and (4.68) in (4.82), we get:

$$\begin{aligned} \frac{1}{M} \sum_{k=1}^M \|\varphi_r^k\| &\leq C \left\{ (1 + \|M_r^{-1}\|_2 + \alpha^{-1})^{1/2} \right. \\ &\quad \left(h^{m+1} \frac{1}{M} \sum_{k=1}^M \|u^k\|_{m+1} + \sqrt{\sum_{j=r+1}^d \lambda_j} \right) \\ &\quad \left. + \sqrt{\varepsilon + \alpha} \left(h^m \frac{1}{M} \sum_{k=1}^M \|u^k\|_{m+1} + \sqrt{\sum_{j=r+1}^d \lambda_j} \right) \right\}. \end{aligned} \quad (4.83)$$

Using (4.63), (4.67), and (4.83), we get (4.61).

Corollary 1 *The Ritz projection w_r^k of u^k satisfies the following error estimate:*

$$\begin{aligned} \|(u^k - w_r^k)_t\| &\leq C \left\{ (1 + \|M_r^{-1}\|_2 + \alpha^{-1})^{1/2} \right. \\ &\quad \left(h^{m+1} \|u_t\|_{L^2(H^{m+1})} + \sqrt{\sum_{j=r+1}^d \lambda_j} \right) \\ &\quad \left. + \sqrt{\varepsilon + \alpha} \left(h^m \|u_t\|_{L^2(H^{m+1})} + \sqrt{\sum_{j=r+1}^d \lambda_j} \right) \right\}. \end{aligned} \quad (4.84)$$

Proof: The proof follows along the same lines as the proof of Lemma 5, with the error $u^k - w_r^k$ replaced by $(u^k - w_r^k)_t = (\eta_n - \varphi_r^k)_t$. Note that it is exactly at this point that we use the fact that the finite difference quotients $\bar{\partial}u(t_k)$ are included in the set of snapshots (see also Remark 1 in [137]). Indeed, as in the proof of Lemma 5, the error $(u^k - w_r^k)_t$ is split into two parts: $(u^k - w_r^k)_t := (u_t^k - I_{h,r}(u_t^k)) - ((w_t)_r^k - I_{h,r}(u_t^k)) = \eta_t^k - (\varphi_t)_r^k$. As in (4.64)–(4.67), η_t^k can be estimated as follows.

$$\begin{aligned} \|\eta_t^k\| &\leq \|u_t^k - u_{h,t}^k\| + \|u_{h,t}^k - I_{h,r}(u_t^k)\| \\ &\leq C \left(h^{m+1} \|u_t\|_{m+1} + \sqrt{\sum_{j=r+1}^d \lambda_j} \right), \end{aligned} \quad (4.85)$$

where in the last inequality in (4.85) we used (4.54).

We are now ready to prove the main result of this section.

Theorem 3 *Assume that*

$$L^R = \nabla X^R = \text{span}\{\nabla\varphi_1, \dots, \nabla\varphi_R\}. \quad (4.86)$$

Then the following error estimate holds:

$$\begin{aligned} \frac{1}{M+1} \sum_{k=0}^M \|u^k - u_r^k\| &\leq C \left\{ (1 + \|M_r^{-1}\|_2 + \alpha^{-1})^{1/2} \right. \\ &\quad \left(h^{m+1} \frac{1}{M} \sum_{k=1}^M (\|u^k\|_{m+1} + \|u_t\|_{L^2(H^{m+1})}) + \sqrt{\sum_{j=r+1}^d \lambda_j} \right) \\ &\quad + \sqrt{\varepsilon + \alpha} \left(h^m \frac{1}{M} \sum_{k=1}^M (\|u^k\|_{m+1} + \|u_t\|_{L^2(H^{m+1})}) + \sqrt{\sum_{j=r+1}^d \lambda_j} \right) \\ &\quad + \|u^0 - u_r^0\| + \Delta t \|u_{tt}\|_{L^2(L^2)} \\ &\quad \left. + \sqrt{\alpha} \left(h^m \frac{1}{M} \sum_{k=1}^M \|u^k\|_{m+1} + \sqrt{\sum_{j=R+1}^d \lambda_j} \right) \right\}. \end{aligned} \quad (4.87)$$

Proof: We evaluate (4.55) at t_{k+1} , we let $v = v_r$, and then we add and subtract $\left(\frac{u^{k+1} - u^k}{\Delta t}, v_r\right)$:

$$\left(u_t^{k+1} - \frac{u^{k+1} - u^k}{\Delta t}, v_r\right) + \left(\frac{u^{k+1} - u^k}{\Delta t}, v_r\right) + a(u^{k+1}, v_r) = (f^{k+1}, v_r). \quad (4.88)$$

Subtracting (4.56) from (4.88), we obtain the error equation:

$$\begin{aligned} & \left(u_t^{k+1} - \frac{u^{k+1} - u^k}{\Delta t}, v_r \right) + \left(\frac{u^{k+1} - u_r^{k+1}}{\Delta t}, v_r \right) - \left(\frac{u^k - u_r^k}{\Delta t}, v_r \right) \\ & + A(u^{k+1} - u_r^{k+1}, v_r) + (a - A)(u^{k+1}, v_r) = 0. \end{aligned} \quad (4.89)$$

We now decompose the error as $u^k - u_r^k = (u^k - w_r^k) - (u_r^k - w_r^k) = \eta^k - \varphi_r^k$, which, by the triangle inequality, implies:

$$\|u^k - u_r^k\| \leq \|\eta^k\| + \|\varphi_r^k\|. \quad (4.90)$$

We note that $\|\eta^k\|$ has already been bounded in Lemma 5. Thus, in order to estimate the error, we only need to estimate $\|\varphi_r^k\|$. The error equation (4.89) can be written as:

$$\begin{aligned} & \left(u_t^{k+1} - \frac{u^{k+1} - u^k}{\Delta t}, v_r \right) + \left(\frac{\eta^{k+1} - \eta^k}{\Delta t}, v_r \right) - \left(\frac{\varphi_r^{k+1} - \varphi_r^k}{\Delta t}, v_r \right) \\ & + A(\eta^{k+1} - \varphi_r^{k+1}, v_r) + (a - A)(u^{k+1}, v_r) = 0. \end{aligned} \quad (4.91)$$

We pick $v_r := \varphi_r^{k+1}$ in (4.91), we note that, since $\varphi_r^{k+1} \in X^r$, $A(\eta^{k+1}, \varphi_r^{k+1}) = 0$, and we get:

$$\begin{aligned} & A(\varphi_r^{k+1}, \varphi_r^{k+1}) + \frac{1}{\Delta t} (\varphi_r^{k+1} - \varphi_r^k, \varphi_r^{k+1}) = \frac{1}{\Delta t} (\eta^{k+1} - \eta^k, \varphi_r^{k+1}) \\ & + (r^k, \varphi_r^{k+1}) + (a - A)(u^{k+1}, \varphi_r^{k+1}), \end{aligned} \quad (4.92)$$

where $r^k = u_t^{k+1} - \frac{u^{k+1} - u^k}{\Delta t}$. We now start estimating all the terms in (4.92). The terms on the LHS of (4.92) are estimated as follows:

$$A(\varphi_r^{k+1}, \varphi_r^{k+1}) \geq \beta \|\varphi_r^{k+1}\|^2 + \varepsilon \|\nabla \varphi_r^{k+1}\|^2 + \alpha \|P'_R \nabla \varphi_r^{k+1}\|^2. \quad (4.93)$$

$$\frac{1}{\Delta t} (\varphi_r^{k+1} - \varphi_r^k, \varphi_r^{k+1}) \geq \frac{1}{\Delta t} (\|\varphi_r^{k+1}\|^2 - \|\varphi_r^k\| \|\varphi_r^{k+1}\|). \quad (4.94)$$

Now we estimate the RHS of (4.92) by using the Cauchy-Schwarz and Young's inequalities:

$$\begin{aligned} & \left(\frac{1}{\Delta t} (\eta^{k+1} - \eta^k) + r^k, \varphi_r^{k+1} \right) \leq \left\| \frac{1}{\Delta t} (\eta^{k+1} - \eta^k) + r^k \right\| \|\varphi_r^{k+1}\| \\ & \leq \frac{1}{2\beta} \left\| \frac{1}{\Delta t} (\eta^{k+1} - \eta^k) + r^k \right\|^2 + \frac{\beta}{2} \|\varphi_r^{k+1}\|^2. \end{aligned} \quad (4.95)$$

$$\begin{aligned} & (a - A)(u^{k+1}, \varphi_r^{k+1}) = -\alpha (P'_R \nabla u^{k+1}, P'_R \nabla \varphi_r^{k+1}) \\ & \leq \alpha \|P'_R \nabla u^{k+1}\| \|P'_R \nabla \varphi_r^{k+1}\| \leq \frac{\alpha}{2} \|P'_R \nabla u^{k+1}\|^2 + \frac{\alpha}{2} \|P'_R \nabla \varphi_r^{k+1}\|^2. \end{aligned} \quad (4.96)$$

Using (4.93)-(4.96) and absorbing RHS terms into LHS terms, (4.92) now reads:

$$\begin{aligned} & \frac{1}{\Delta t} (\|\varphi_r^{k+1}\|^2 - \|\varphi_r^k\| \|\varphi_r^{k+1}\|) + \frac{\beta}{2} \|\varphi_r^{k+1}\|^2 + \varepsilon \|\nabla \varphi_r^{k+1}\|^2 \\ & + \frac{\alpha}{2} \|P'_R \nabla \varphi_r^{k+1}\|^2 \leq \frac{1}{2\beta} \left\| \frac{1}{\Delta t} (\eta^{k+1} - \eta^k) + r^k \right\|^2 + \frac{\alpha}{2} \|P'_R \nabla u^{k+1}\|^2. \end{aligned} \quad (4.97)$$

By using Young's inequality, the first term on the LHS of (4.97) can be estimated as follows:

$$\begin{aligned} \|\varphi_r^{k+1}\|^2 - \|\varphi_r^k\| \|\varphi_r^{k+1}\| & \geq \|\varphi_r^{k+1}\|^2 - \frac{1}{2} \|\varphi_r^k\|^2 - \frac{1}{2} \|\varphi_r^{k+1}\|^2 \\ & = \frac{1}{2} \|\varphi_r^{k+1}\|^2 - \frac{1}{2} \|\varphi_r^k\|^2. \end{aligned} \quad (4.98)$$

Using (4.98) in (4.97) and multiplying by $2\Delta t$, we get:

$$\begin{aligned} & \|\varphi_r^{k+1}\|^2 - \|\varphi_r^k\|^2 + \Delta t \|\varphi_r^{k+1}\|_{1,\varepsilon,\alpha}^2 \\ & \leq C \left(\Delta t \left\| \frac{1}{\Delta t} (\eta^{k+1} - \eta^k) + r^k \right\|^2 + \alpha \Delta t \|P'_R \nabla u^{k+1}\|^2 \right) \\ & \leq C \left(\Delta t \left\| \frac{1}{\Delta t} (\eta^{k+1} - \eta^k) \right\|^2 + \Delta t \|r^k\|^2 + \alpha \Delta t \|P'_R \nabla u^{k+1}\|^2 \right). \end{aligned} \quad (4.99)$$

Summing from $k = 0$ to $k = M - 1$ in (4.99), we get:

$$\begin{aligned} \max_{0 \leq k \leq M} \|\varphi_r^k\|^2 + \sum_{k=0}^{M-1} \Delta t \|\varphi_r^{k+1}\|_{1,\varepsilon,\alpha}^2 & \leq C \left(\Delta t \sum_{k=0}^{M-1} \left\| \frac{1}{\Delta t} (\eta^{k+1} - \eta^k) \right\|^2 \right. \\ & \left. + \|\varphi_r^0\|^2 + \Delta t \sum_{k=0}^{M-1} \|r^k\|^2 + \alpha \Delta t \sum_{k=0}^{M-1} \|P'_R \nabla u^{k+1}\|^2 \right). \end{aligned} \quad (4.100)$$

Proceeding as in [214] (see also [106]), we estimate the first term on the RHS of (4.100) as follows. We start by writing:

$$\eta^{k+1} - \eta^k = \int_{t_k}^{t_{k+1}} \eta_t dt. \quad (4.101)$$

Taking the L^2 -norm in (4.101) and applying the Cauchy-Schwarz inequality, we get:

$$\begin{aligned} \|\eta^{k+1} - \eta^k\| & \leq \int_{t_k}^{t_{k+1}} 1 \|\eta_t\| dt \leq \left(\int_{t_k}^{t_{k+1}} 1^2 dt \right)^{1/2} \left(\int_{t_k}^{t_{k+1}} \|\eta_t\|^2 dt \right)^{1/2} \\ & \leq (\Delta t)^{1/2} \left(\int_{t_k}^{t_{k+1}} \|\eta_t\|^2 dt \right)^{1/2}, \end{aligned} \quad (4.102)$$

which implies $\Delta t \left\| \frac{1}{\Delta t} (\eta^{k+1} - \eta^k) \right\|^2 \leq \left(\int_{t_k}^{t_{k+1}} \|\eta_t\|^2 dt \right)^{1/2}$. Summing from $k = 0$ to $k = M - 1$, we get $\Delta t \sum_{k=0}^{M-1} \left\| \frac{1}{\Delta t} (\eta^{k+1} - \eta^k) \right\|^2 \leq \|\eta_t\|_{L^2(L^2)}$, which was bound in Corollary 1. We thus obtain:

$$\begin{aligned} \Delta t \sum_{k=0}^{M-1} \left\| \frac{1}{\Delta t} (\eta^{k+1} - \eta^k) \right\|^2 \leq C \left\{ (1 + \|M_r^{-1}\|_2 + \alpha^{-1})^{1/2} \right. \\ \left. \left(h^{m+1} \|u_t\|_{L^2(H^{m+1})} + \sqrt{\sum_{j=r+1}^d \lambda_j} \right) \right. \\ \left. + \sqrt{\varepsilon + \alpha} \left(h^m \|u_t\|_{L^2(H^{m+1})} + \sqrt{\sum_{j=r+1}^d \lambda_j} \right) \right\}. \end{aligned} \quad (4.103)$$

To estimate the third term on the RHS of (4.100), we use a Taylor series expansion of u^k around u^{k+1} :

$$u^k = u^{k+1} - u_t^{k+1} \Delta t + \int_{t_k}^{t_{k+1}} u_{tt}(s) (t_k - s) ds. \quad (4.104)$$

Taking the L^2 -norm in (4.104) and applying the Cauchy-Schwarz inequality, we get $\|r^k\| \leq \int_{t_k}^{t_{k+1}} \|u_{tt}\| ds \leq (\Delta t)^{1/2} \|u_{tt}\|_{L^2(L^2)}$. Summing from $k = 0$ to $k = M - 1$, we get:

$$\Delta t \sum_{k=0}^{M-1} \|r^k\|^2 \leq \Delta t^2 \|u_{tt}\|_{L^2(L^2)}^2. \quad (4.105)$$

To estimate the last term on the RHS of (4.100), we use the fact that $L^R = \nabla X^R$ (assumption (4.86)). We emphasize that this is the *only instance* in the proof where the assumption $L^R = \nabla X^R$ is used. Thus, we get:

$$\begin{aligned} \alpha \Delta t \sum_{k=0}^{M-1} \|P'_R \nabla u^{k+1}\|^2 &= \alpha \Delta t \sum_{k=0}^{M-1} \|\nabla u^{k+1} - P_R \nabla u^{k+1}\|^2 \\ &\stackrel{(4.86)}{\leq} C \alpha \frac{1}{M} \sum_{k=0}^{M-1} \inf_{v_R \in X^R} \|\nabla u^{k+1} - \nabla v_R\|^2 \\ &\stackrel{(1.9), (4.46)}{\leq} C \alpha \left(h^m \frac{1}{M} \sum_{k=1}^M \|u^k\|_{m+1} + \sqrt{\sum_{j=R+1}^d \lambda_j} \right)^2. \end{aligned} \quad (4.106)$$

Using (4.103), (4.105), and (4.106) in (4.100), the obvious inequality $\max_{0 \leq k \leq M} \|\varphi_r^k\| \geq \frac{1}{M+1} \sum_{k=0}^M \|\varphi_r^k\|$, inequality (4.90), and the estimates in Lemma 5, we obtain the error estimate (4.87).

4.3.2 Numerical Results

The goal of this section is twofold: (i) to show that the new P_R-VMS-POD-ROM (4.44) is significantly more stable numerically than the standard POD-G-ROM (4.42); and (ii) to illustrate numerically the theoretical error estimate (4.87). We also use Theorem 3 to provide theoretical guidance in choosing an optimal value for the artificial viscosity coefficient α and use this algorithm within our numerical framework. Finally, we show that the P_R-VMS-POD-ROM (4.44) displays a relatively low sensitivity with respect to changes in the diffusion coefficient ε . Thus, we provide numerical support for the theoretical estimate (4.87), which is *uniform* with respect to ε .

The mathematical model used for all the numerical tests in this section is the convection-dominated convection-diffusion-reaction equation (4.40) with the following parameter choices: spatial domain $\Omega = [0, 1] \times [0, 1]$, time interval $[0, T] = [0, 1]$, diffusion coefficient $\varepsilon = 1 \times 10^{-4}$, convection field $\mathbf{b} = [\cos \frac{\pi}{3}, \sin \frac{\pi}{3}]^T$, and reaction coefficient $g = 1$. The forcing term f and initial condition $u_0(x)$ are chosen to satisfy the exact solution $u(x, y, t) = 0.5 \sin(\pi x) \sin(\pi y) [\tanh(\frac{x+y-t-0.5}{0.04}) + 1]$, which is similar to that used in [98]. As in the theoretical developments in Section 4.3.1, in this section we employ the finite element method for spatial discretization and the backward Euler method for temporal discretization of all models investigated. All computations are carried out on a PC with 3.2 GHz Intel Xeon Quad-core processor.

We start by comparing the P_R-VMS-POD-ROM (4.44) to the standard POD-G-ROM (4.42). To generate the POD basis, we first run a DNS with the following parameters: piecewise quadratic finite elements, uniform triangular mesh with mesh-size $h = 0.01$, and time-step $\Delta t = 10^{-4}$. A mesh refinement study indicates that DNS mesh resolution is achieved. The average DNS error is $\frac{1}{M+1} \sum_{k=0}^M \|u^k - u_h^k\| = 2.04 \times 10^{-4}$, where $M = 1000$, and u^k and u_h^k are the exact solution and the finite element solution at $t = k\Delta t$, respectively. The

CPU time of the DNS is 9.42×10^4 s. Since the forcing term is time-dependent, the global load vectors are stored for later use in all the POD-ROMs. The POD modes are generated in H^1 by the method of snapshots; the rank of the data set is 104. For both POD-ROMs (POD-G-ROM and P_R-VMS-POD-ROM), we use the same number of POD basis functions: $r = 40$.

Table 4.3: Average errors for the POD-G-ROM (4.42) with different values of r .

r	20	40	60	80
$\frac{1}{M} \sum_{k=1}^M \ u^k - u_r^k\ $	1.25×10^{-1}	1.11×10^{-1}	9.28×10^{-2}	8.20×10^{-2}

We first test the POD-G-ROM (4.42). The CPU time for the POD-G-ROM is 96.4 s, which is three orders of magnitude lower than that of a brute force DNS. The numerical solution at $t = 1$ is shown in Figure 4.9 for both the DNS (top) and the POD-G-ROM (middle). It is clear from this figure that, although the first 40 POD modes capture 99.99% of the system's kinetic energy, the POD-G-ROM yields poor quality results and displays strong numerical oscillations. This is confirmed by the POD-G-ROM's high average error $\frac{1}{M} \sum_{k=1}^M \|u^k - u_r^k\| = 1.11 \times 10^{-1}$, where u_r^k is the POD-G-ROM's solution at $t = k\Delta t$. Indeed, the POD-G-ROM's average error is almost three orders of magnitude higher than the average error of the DNS. The average errors for different values of r listed in Table 4.3 show that increasing the number of POD modes (r) does not decrease significantly the average error. It is thus clear that the straightforward POD-G-ROM, although computationally efficient, is highly inaccurate.

Next, we investigate the P_R-VMS-POD-ROM (4.44). We make the following parameter choices: $R = 20$ and $\alpha = 4.29 \times 10^{-2}$. The motivation for this choice is given later in this section. The CPU time for the P_R-VMS-POD-ROM (4.44) is 106.2 s, which is close to the CPU time of the POD-G-ROM (4.42). The numerical solution at $t = 1$ for the P_R-VMS-POD-ROM is shown in Figure 4.9 (bottom). It is clear from this figure that the P_R-VMS-POD-ROM is much more accurate than the POD-G-ROM. Indeed, the P_R-VMS-POD-ROM results are much closer to the DNS results than the POD-G-ROM results, since the numerical oscillations displayed by the latter are dramatically decreased. This is

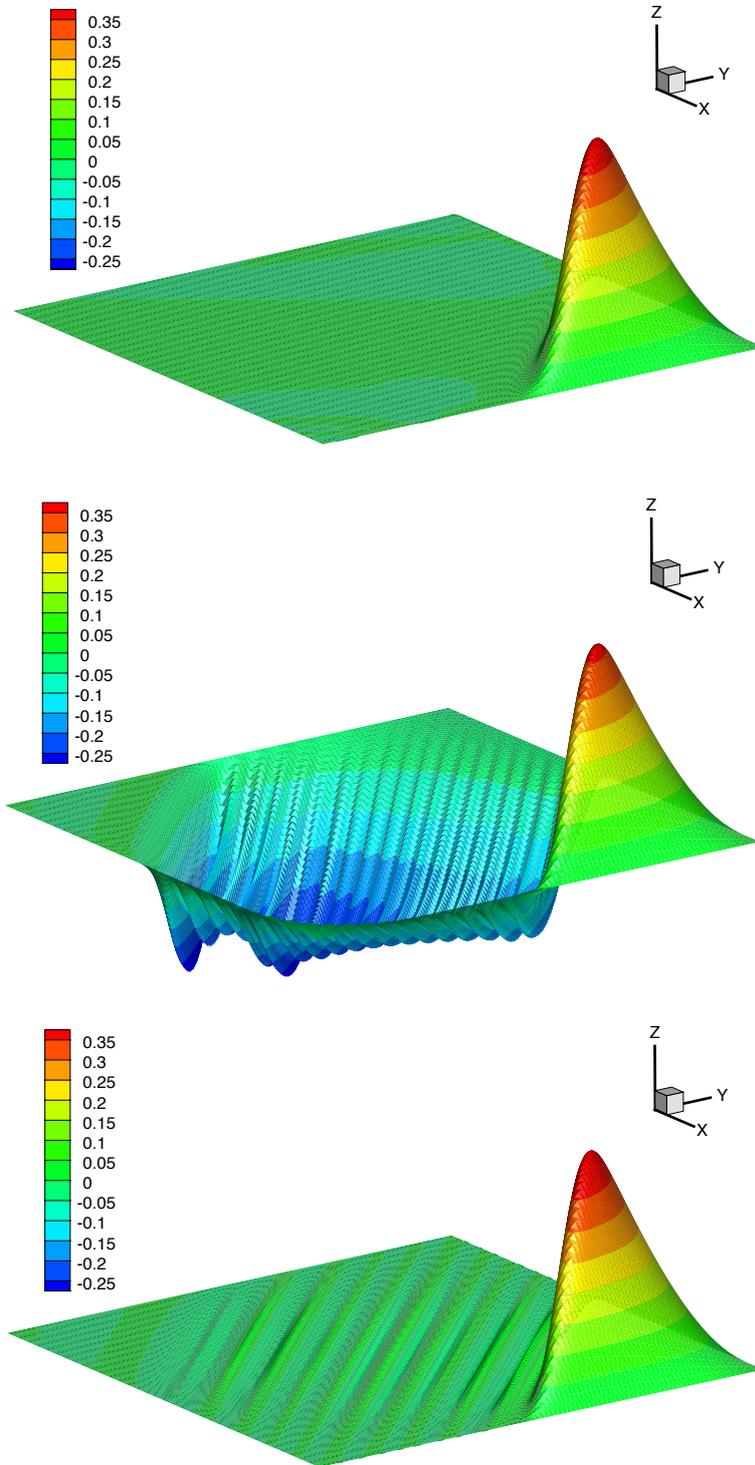
confirmed by the P_R-VMS-POD-ROM's average error $\frac{1}{M} \sum_{k=1}^M \|u^k - u_r^k\| = 4.48 \times 10^{-3}$, where u_r^k is the VMS-POD solution at $t = k\Delta t$; this error is more than 20 times lower than the error of the POD-G-ROM. In conclusion, the P_R-VMS-POD-ROM (4.44) dramatically decreases the error of the POD-G-ROM (4.42) by adding numerical stabilization, while keeping the same level of computational efficiency.

Table 4.4: P_R-VMS-POD-ROM's average error $e = \frac{1}{M} \sum_{k=1}^M \|u^k - u_r^k\|$ and its e_3 component for different values of R .

R	e_3	$\frac{1}{M} \sum_{k=1}^M \ u^k - u_r^k\ $
1	1.29×10^{-1}	2.55×10^{-2}
4	9.34×10^{-2}	1.78×10^{-2}
7	6.69×10^{-2}	1.37×10^{-2}
10	4.68×10^{-2}	9.80×10^{-3}
13	3.20×10^{-2}	6.99×10^{-3}

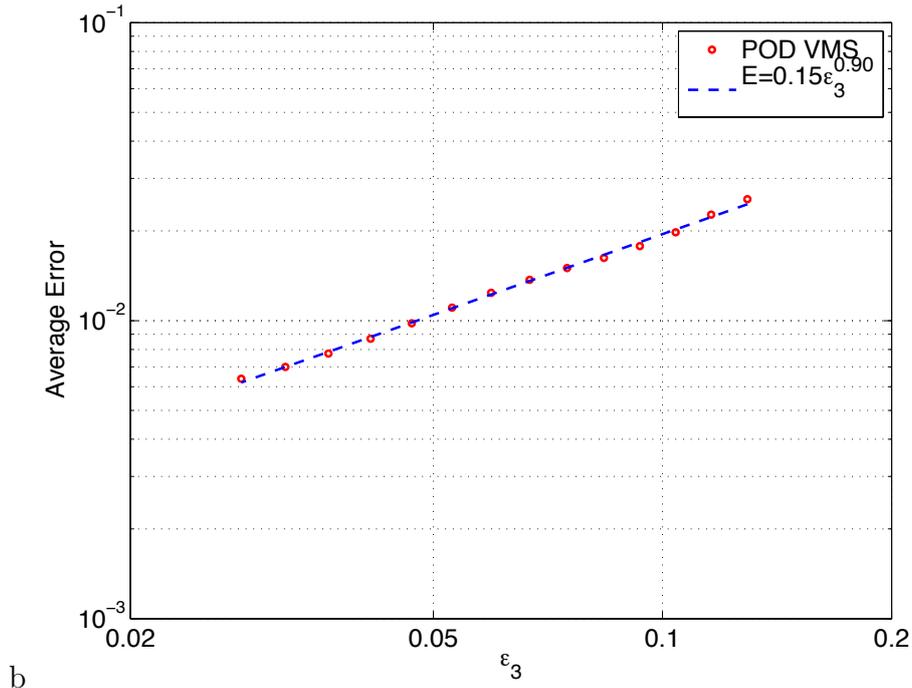
We now turn our attention to the second major goal of this section - the numerical illustration of the theoretical error estimate (4.87). Specifically, we investigate whether the asymptotic behavior of the RHS of estimate (4.87) with respect to R is reflected in the numerical results. We focus on the asymptotic behavior with respect to R since this is the main parameter introduced by the VMS formulation; the asymptotic behavior with respect to r was investigated in [38], whereas the asymptotic behavior with respect to h and Δt is standard [41, 214]. To investigate the asymptotic behavior with respect to R , we have to ensure that $\sqrt{\alpha} \sqrt{\sum_{j=R+1}^d \lambda_j}$ (the only term that depends on R) dominates all the other terms on the RHS of (4.87). To this end, we start collecting all the terms that depend on the exact solution u and we include them in the generic constant C . Next, we assume that the POD interpolation error in the initial condition $\|u^0 - u_r^0\|$ is negligible. We also assume that the time-step is small enough to neglect $\Delta t \|u_{tt}\|_{L^2(L^2)}$. With these assumptions, the error estimate (4.87) can now be written as $e \leq C(e_1 + e_2 + e_3)$, where e is the P_R-VMS-POD-ROM's average error, C a generic constant independent of $r, R, h, \Delta t$

Figure 4.9: Numerical solution at $t = 1$: DNS (top), POD-G-ROM (4.42) (middle), and P_R -VMS-POD-ROM (4.44) (bottom).



and α , $e_1 = \|M_r^{-1}\|_2^{\frac{1}{2}} h^{m+1}$, $e_2 = \|M_r^{-1}\|_2^{\frac{1}{2}} \sqrt{\sum_{j=r+1}^d \lambda_j}$, and $e_3 = \sqrt{\alpha} \sqrt{\sum_{j=R+1}^d \lambda_j}$. To ensure that e_3 dominates the other terms, we choose $r = 100$ and consider relatively low values for R . This choice for r , which is not optimal for practical computations, ensures, however, that e_3 dominates e_2 . We also note that, when h is small, e_3 dominates e_1 , too. Thus, to investigate the asymptotic behavior with respect to R of the RHS of (4.87), we fix $\alpha = 5 \times 10^{-3}$, vary R from 1 to 14, and monitor the changes in e_3 . We restrict R to this parameter range to ensure that $\sqrt{\sum_{j=R+1}^d \lambda_j}$ (and thus e_3) dominates e_2 and e_1 . Table 4.4 lists the P_R-VMS-POD-ROM's average error $e = \frac{1}{M} \sum_{k=1}^M \|u^k - u_r^k\|$ and its e_3 component for different values of R . We emphasize that, in this case, e_3 dominates the other two error components $e_1 = 3.81 \times 10^{-3}$ and $e_2 = 2.87 \times 10^{-3}$. To see whether the theoretical linear dependency predicted by the theoretical error estimate (4.87) is recovered in the numerical results in Table 4.4, we utilize a linear regression analysis in Figure 4.10. This plot shows that the rate of convergence of e with respect to e_3 is 0.9, which is close to the theoretical value of 1 predicted by (4.87). We believe that this slight discrepancy is due to the fact that the mesh-size $h = 0.01$ that we employed in this numerical investigation is not small enough for our asymptotic study.

Figure 4.10: Linear regression of P_R-VMS-POD-ROM's average error with respect to e_3 .



Summarizing the results above, we conclude that the theoretical error estimate in (4.87) is recovered asymptotically (with respect to R) in our numerical experiments.

Next, we use Theorem 3 to provide theoretical guidance in choosing an optimal value for the artificial viscosity coefficient α . The main challenge is that the theoretical error estimate (4.87) is *asymptotic* with respect to $h, \Delta t$ and r , while in practical computations we are using small, yet non-negligible values for these parameters. Furthermore, the generic constant C is problem-dependent and can play a significant role in practical computations. Notwithstanding these hurdles, we choose a value for α that minimizes the RHS of (4.87):

$\tilde{\alpha} = \frac{h^{m+1} + \sqrt{\sum_{j=r+1}^d \lambda_j}}{2h^m + \sqrt{\sum_{j=r+1}^d \lambda_j} + \sqrt{\sum_{j=R+1}^d \lambda_j}}$. In the derivation of this formula, we made the same assumptions as those made in the numerical investigation of the asymptotic behavior of the P_R-VMS-POD-ROM's error and we again considered that (4.87) can be written as $e \leq C(e_1 + e_2 + e_3)$.

We note that, if $\sqrt{\sum_{j=r+1}^d \lambda_j} \ll \sqrt{\sum_{j=R+1}^d \lambda_j}$ and $h^m \ll \sqrt{\sum_{j=R+1}^d \lambda_j}$, then $\tilde{\alpha}$ becomes too small in practical computations and the P_R-VMS-POD-ROM becomes similar to the inaccurate POD-G-ROM. To circumvent this, we use in our numerical tests a ‘‘clipping’’ procedure (see, e.g., [37]) by setting $\alpha^* = \max\{\tilde{\alpha}, \frac{h}{2}\}$.

Table 4.5 lists the P_R-VMS-POD-ROM's average error $e = \frac{1}{M} \sum_{k=1}^M \|u^k - u_r^k\|$ for the following values of r, R and α : $r = 20, 40$ and 60 ; R from 5 to $r - 5$ in increments of 5 ; and $\alpha = 0.01\alpha^*, \alpha^*$, and $100\alpha^*$. Note that the P_R-VMS-POD-ROM consistently performs best for $\alpha = \alpha^*$. The only two slight deviations from this rule are for $r = 60$ ($R = 20$ and $R = 30$); we again believe that this is due to the mesh-size $h = 0.01$, which is not small enough for the asymptotic regime in Theorem 3.

Finally, we investigate numerically the P_R-VMS-POD-ROM's sensitivity with respect to changes in the diffusion coefficient ε . To this end, we run the P_R-VMS-POD-ROM (4.44) with the same parameters as above ($r = 40, R = 20$ and $\alpha = \alpha^*$) for different values of the diffusion coefficient: $\varepsilon = 10^{-2}, 10^{-4}$ and 10^{-6} . Table 4.6 lists the average errors for DNS, POD-G-ROM and P_R-VMS-POD-ROMs for different values of ε . It is clear from this table that the POD-G-ROM's average error is significantly higher than the error of the DNS. The P_R-VMS-POD-ROM, however, performs well for all values of ε and displays a low sensitivity

Table 4.5: P_R-VMS-POD-ROM's average error $e = \frac{1}{M} \sum_{k=1}^M \|u^k - u_r^k\|$ for different values of r and R , and $\alpha = 0.01\alpha^*$, α^* , and $100\alpha^*$.

r	R	$0.01\alpha^*$	e	α^*	e	$100\alpha^*$	e
20	5	1.2×10^{-3}	1.0×10^{-1}	1.2×10^{-1}	5.8×10^{-2}	1.2×10^1	7.8×10^{-2}
	10	2.0×10^{-3}	9.5×10^{-2}	2.0×10^{-1}	2.4×10^{-2}	2.04×10^1	2.6×10^{-2}
	15	3.3×10^{-3}	8.2×10^{-2}	3.3×10^{-1}	2.0×10^{-2}	3.3×10^1	2.5×10^{-2}
40	5	6.4×10^{-5}	1.09×10^{-1}	6.4×10^{-3}	3.0×10^{-2}	6.4×10^{-1}	7.2×10^{-2}
	10	1.1×10^{-4}	1.0×10^{-1}	1.1×10^{-2}	1.8×10^{-2}	1.1×10^0	2.5×10^{-2}
	20	4.2×10^{-4}	9.7×10^{-2}	4.2×10^{-2}	4.4×10^{-3}	4.2×10^0	4.1×10^{-3}
	30	1.7×10^{-3}	6.8×10^{-2}	1.7×10^{-1}	8.1×10^{-3}	1.7×10^1	1.0×10^{-2}
	35	3.0×10^{-3}	4.9×10^{-2}	3.0×10^{-1}	2.1×10^{-2}	3.0×10^1	2.4×10^{-2}
60	5	5.0×10^{-5}	8.7×10^{-2}	5.0×10^{-3}	1.8×10^{-2}	5.0×10^{-1}	7.0×10^{-2}
	10	5.0×10^{-5}	8.7×10^{-2}	5.0×10^{-3}	1.3×10^{-2}	5.0×10^{-1}	2.4×10^{-2}
	20	5.0×10^{-5}	8.7×10^{-2}	5.0×10^{-3}	1.0×10^{-2}	5.0×10^{-1}	3.9×10^{-3}
	30	1.2×10^{-4}	8.0×10^{-2}	1.2×10^{-2}	4.4×10^{-3}	1.2×10^0	7.4×10^{-4}
	40	5.4×10^{-4}	5.5×10^{-2}	5.4×10^{-2}	1.2×10^{-3}	5.4×10^0	2.4×10^{-3}
	50	1.8×10^{-3}	2.6×10^{-2}	1.8×10^{-1}	1.3×10^{-2}	1.8×10^1	1.4×10^{-2}
	55	2.9×10^{-3}	2.2×10^{-2}	2.9×10^{-1}	1.1×10^{-2}	2.9×10^1	1.2×10^{-2}

with respect to changes in the diffusion coefficient. Thus, we provide numerical support for the theoretical estimate (4.87), which is *uniform* with respect to ε .

4.4 Summary

In this chapter, we have developed, analyzed and tested numerically two new POD-ROMs. One is the AV-POD-ROM (4.9), which introduces an artificial viscosity term to account for the loss of information in the POD-G-ROM. The other is the P_R-VMS-POD-ROM (4.44), which uses a new VMS closure modeling strategy for POD-ROMs of convection-dominated equations. Although similar models were proposed in one of our earlier papers [35], this is the first time to our knowledge that both models are analyzed and tested numerically.

Theoretical and numerical results for the AV-POD-ROM (4.9) have been developed for

Table 4.6: Average errors of DNS, POD-G-ROM and P_R-VMS-POD-ROM for different values of the diffusion coefficient ε .

ε	DNS	POD-G	VMS-POD	
	$\frac{1}{M+1} \sum_{k=0}^M \ u^k - u_h^k\ $	$\frac{1}{M} \sum_{k=1}^M \ u^k - u_r^k\ $	α	$\frac{1}{M} \sum_{k=1}^M \ u^k - u_r^k\ $
10^{-2}	1.10×10^{-4}	1.10×10^{-2}	4.05×10^{-2}	4.27×10^{-3}
10^{-4}	2.04×10^{-4}	1.11×10^{-1}	4.29×10^{-2}	4.48×10^{-3}
10^{-6}	1.88×10^{-4}	1.17×10^{-1}	9.65×10^{-2}	4.05×10^{-3}
10^{-8}	2.46×10^{-4}	1.17×10^{-1}	1.01×10^{-1}	4.05×10^{-3}

the Burgers equation (4.1). This choice allowed us a clear presentation of our results. We emphasize, however, that the new AV-POD-ROM is by no means restricted to this one-dimensional setting. It is, in fact, developed for realistic turbulent flows.

Following [137], we presented a thorough numerical analysis for the discretization of the new AV-POD-ROM. We proved the specific dependence of the total error on the temporal discretization and, more importantly, the dependence on the POD truncation. To test the new AV-POD-ROM numerically, we used two test problems, both displaying shock-like behavior. In both tests, the AV-POD-ROM clearly outperformed the standard POD-G-ROM by damping the numerical oscillations produced by the latter. The improved performance of the AV-POD-ROM was displayed graphically as well as numerically.

The new P_R-VMS-POD-ROM (4.44) also introduces an artificial viscosity term for numerical stabilization of POD-ROMs. Following the guiding principle of the VMS methodology, we only add artificial viscosity to the small resolved scales. Thus, no artificial viscosity is used for the large resolved scales. The POD setting represents an ideal framework for the VMS approach, since the POD modes are listed in descending order of their kinetic energy content.

A thorough numerical analysis for the finite element discretization of the new P_R-VMS-POD-ROM was presented. The numerical tests showed the increased numerical stability of the new P_R-VMS-POD-ROM and illustrated the theoretical error estimates. We also employed the theoretical error estimates to provide guidance in choosing the artificial viscosity coefficient in practical computations. We emphasize that the theoretical error estimates

were uniform with respect to ε , the diffusion coefficient. The numerical tests confirmed the theoretical results: The average error of the P_R-VMS-POD-ROM showed a low sensitivity with respect to changes in ε .

Although the new P_R-VMS-POD-ROM targets general convection-dominated problems, it was analyzed theoretically and tested numerically by using the convection-dominated convection-diffusion-reaction equations. We chose this simplified mathematical and numerical setting as a first step in a thorough investigation of the new P_R-VMS-POD-ROM.

These steps in the assessment of the new POD closure models are encouraging. In Chapter 5, we test these models in a realistic turbulent flow: a 3D flow past a circular cylinder at $\text{Re} = 1000$ [223]. We also believe that our new methodology could have a major impact in the reduced-order modeling and data assimilation of oceanic and atmospheric flows [77]. We will pursue these research directions in the future (see Section 6.4).

Chapter 5

Numerical Experiments of POD Closure Models in Turbulence [†]

5.1 Introduction

In this chapter, we use a structurally dominated 3D turbulent flow problem to test the four POD-ROMs described in Section 2.3: (i) the ML-POD-ROM (2.14)-(2.15); (ii) the S-POD-ROM (2.17)-(2.18); (iii) the new VMS-POD-ROM (2.25)-(2.34); and (iv) the new DS-POD-ROM (2.52)-(2.53). We also include results for the POD-G-ROM (1.12) (i.e., a POD-ROM without any closure model). A successful POD closure model should at least perform better than the POD-G-ROM (1.12). Finally, a DNS projection of the evolution of the POD modes served as benchmark for our numerical simulations: The closeness to the DNS data was used as a criterion for the success of the POD closure model.

The qualitative behavior of all POD-ROMs is judged according to the following five criteria: (i) the kinetic energy spectrum; (ii) the mean velocity; (iii) the Reynolds stresses; (iv) the root mean square (rms) values of the velocity fluctuations; and (v) the time evolution of the POD coefficients. The first four criteria measure the temporal and spatial average behavior of the POD-ROMs, whereas the last criterion measures the instantaneous behavior

[†]The material in this chapter is based on the following report:

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.*, in press, 2012.

of the POD-ROMs. We also include a computational efficiency assessment for all four POD-ROMs as well as a sensitivity study to measure the robustness of the numerical results with respect to changes in r , the number of POD modes retained in the POD-ROMs. In Section 5.2, details of the numerical methods and parameter choices are given. In Section 5.3, numerical results are presented and discussed. Several conclusions are drawn in Section 5.4.

5.2 Numerical Methods and Parameter Choices

We investigate all four POD-ROMs in the numerical simulation of a 3D flow past a circular cylinder at $\text{Re} = 1000$. The wake of the flow is fully turbulent. The cylinder is parallel to the z -axis and the flow is along the positive direction of the x -axis (see Figure 5.1). In this section, u denotes the streamwise velocity component (pointing in the positive direction of the x -axis), v denotes the spanwise velocity component (pointing in the positive direction of the y -axis), and w denotes the normal velocity component (pointing in the positive direction of the z -axis). A parallel CFD solver is employed on the time interval $[0, 300]$ to generate the DNS data [4]. Details on the numerical discretization are presented in the Appendix of [224].

Collecting 1000 snapshots of the velocity field (u, v, w) over the time interval $[0, 75]$ and applying the method of snapshots developed in [206, 207, 208], we obtain the POD basis $\{\varphi_1, \dots, \varphi_d\}$ with $d = 1000$. These POD modes are then interpolated onto a structured quadratic finite element mesh with nodes coinciding with the nodes used in the original DNS finite volume discretization. The first $r = 6$ POD modes capture 84% the system's kinetic energy. These modes are used in all POD-ROMs that we investigate. For all the POD-ROMs, the time discretization was effected by using the explicit Euler method with $\Delta t = 7.5 \times 10^{-4}$.

POD-ROMs for the pressure field have also been developed [8, 172]. The advantage of these POD-ROMs is that the computation of hydrodynamic forces on structures becomes possible. A pressure-Poisson based POD-ROM that requires the POD modes of the pressure field in addition to those for the velocity field was proposed in [8]. This model predicted the lift and drag forces on a cylinder at $\text{Re} = 100$. In this dissertation, however, we only

develop POD-ROMs of the velocity field. Thus, lift and drag forces cannot be computed and compared with those from DNS.

It is important to note that the quadratic nonlinearity in the NSE (1.10) allows for easy precomputation of the vector \mathbf{b} , the matrix \mathbf{A} and the tensor \mathbf{B} in the POD-G-ROM (1.13). For the general nonlinear EV POD closure model (2.11), however, the vector $\tilde{\mathbf{b}}(\mathbf{a})$ and the matrix $\tilde{\mathbf{A}}(\mathbf{a})$ that correspond to the additional closure terms have to be recomputed (re-assembled) at each time step. Since the POD basis functions are global, although only a few are used in POD-ROMs ($r \ll N$), reassembling $\tilde{\mathbf{b}}(\mathbf{a})$ and $\tilde{\mathbf{A}}(\mathbf{a})$ at each time step would dramatically increase the CPU time of the corresponding POD-ROM. Thus, the major advantage of POD-ROMs (the dramatic decrease of computational time), would be completely lost.

To ensure a high computational efficiency of the POD-ROMs, we utilize two approaches: (1) Instead of updating the closure terms in the POD-ROMs every time step, we recompute them every 1.5 time units (i.e., every 20,000 time steps). The previous numerical investigations in [223] showed that this approach does not compromise the numerical accuracy of the S-POD-ROM (2.17)-(2.18). (2) We employ the two-level algorithm introduced in [223] and also described in Section 3.2 to discretize the nonlinear closure models. Before briefly describing the two-level algorithm, we emphasize that, in order to maintain a fair numerical comparison of the four POD-ROMs, we used both algorithmic choices listed above in *all* four POD-ROMs. Therefore, the success or failure of the POD-ROM can solely be attributed to the closure term, which is the only distinguishing feature among all POD-ROMs, and not to the specific algorithmic choices, which are the same for all POD-ROMs.

The two-level algorithm used in all four POD-ROMs is shown in (3.3). In all four POD-ROMs, we apply the two-level algorithm with a coarsening factor $R_c = 4$ in both radial and azimuthal directions. Thus, the vectors and matrices related to the nonlinear closure terms are computed on a coarse finite element mesh with $37 \times 49 \times 17$ grid points.

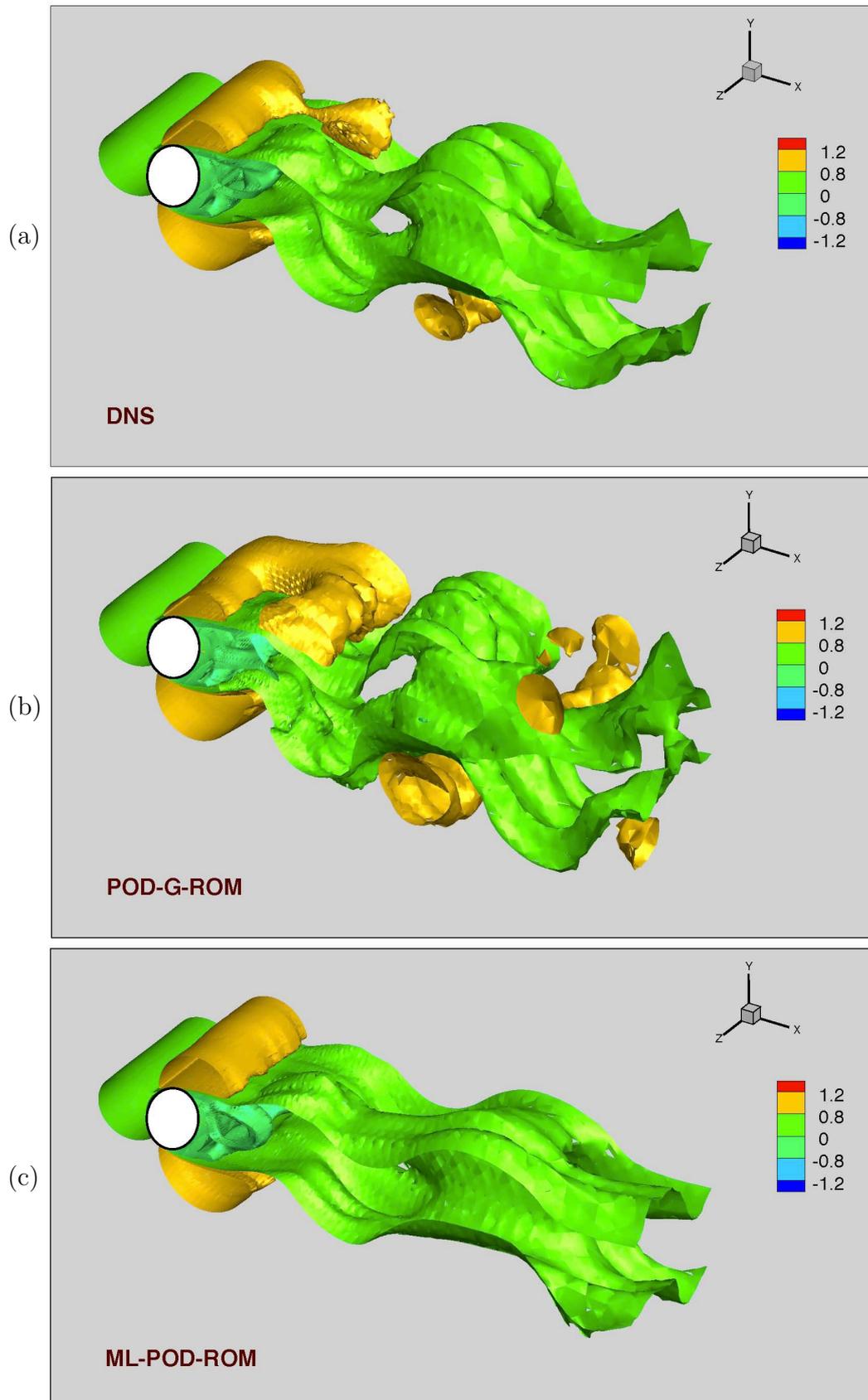
In Section 2.2, we proposed two definitions for the POD lengthscale δ . Since in the finite element discretization that we employ, definition (2.10) is harder to implement than (2.9), we use the latter. Thus, using definition (2.9) with $r = 6$, we obtain $\delta = 0.1179$, which is the POD lengthscale that we will use in all four POD-ROMs. For the DS-POD-ROM (2.52)-

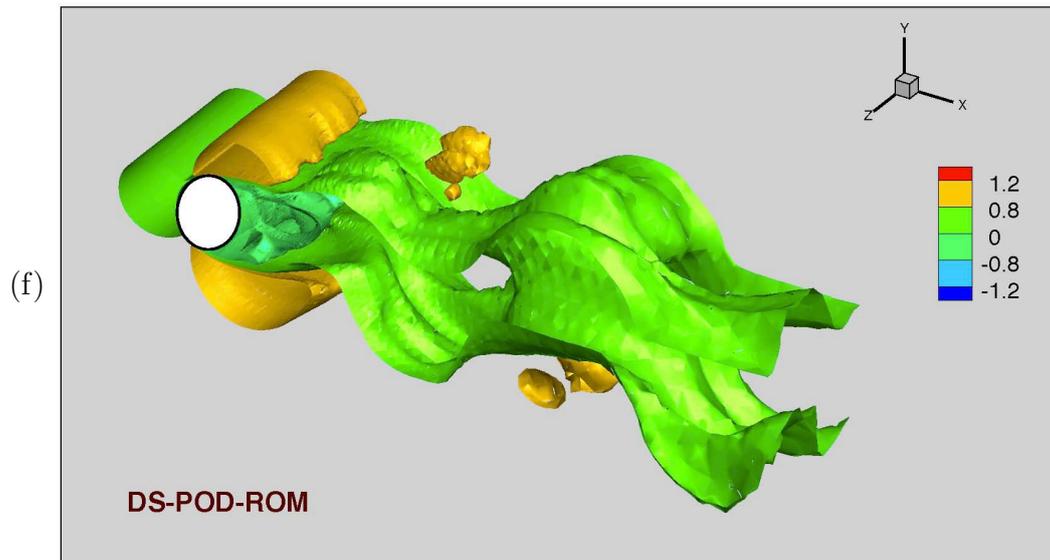
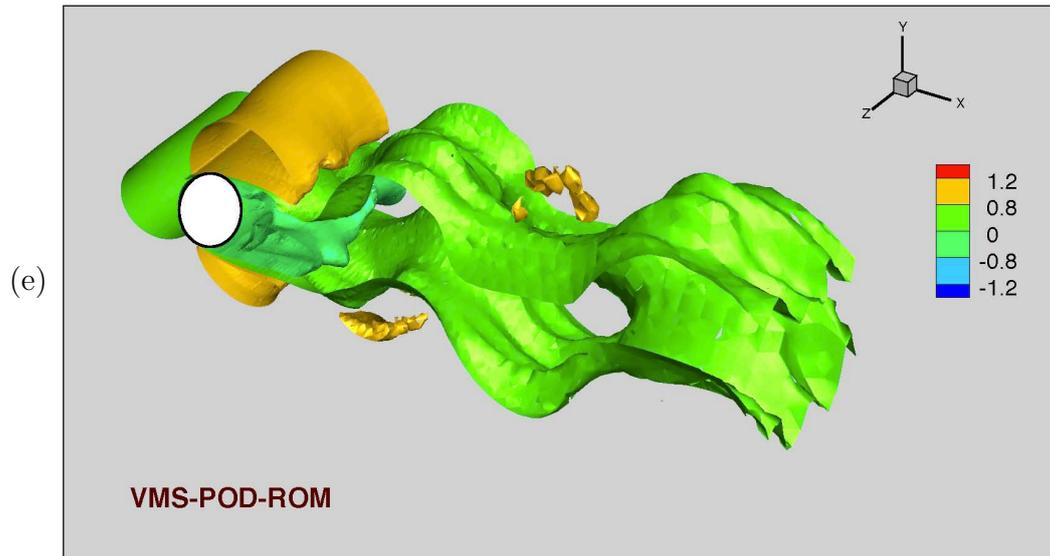
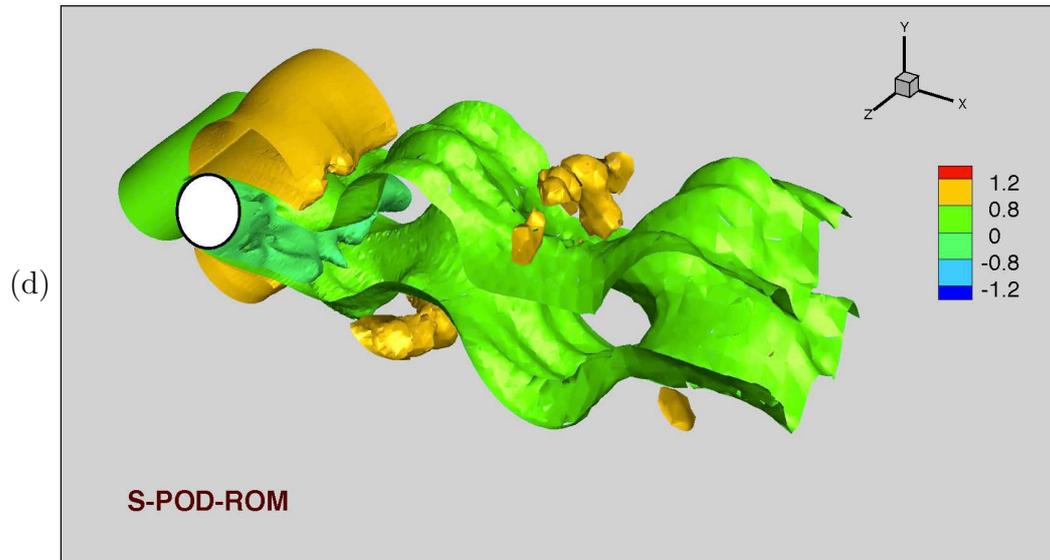
(2.53), we need to define the second (test) Galerkin projection (2.41) and the corresponding filter radius $\tilde{\delta}$. Choosing $R = 1$ in (2.41) and using (2.9), we obtain $\tilde{\delta} = 0.1769$.

The constants in EV LES models are determined in a straightforward fashion, utilizing scaling laws satisfied by general 3D turbulent flows (see for example, [199]). Although the energy cascade concept in a POD context was verified numerically in [60], there are no general scaling laws available in this setting. Thus, to our knowledge, the correct values for the EV constants α in the ML-POD-ROM (2.14)-(2.15) and C_S in the S-POD-ROM (2.17)-(2.18) and the new VMS-POD-ROM (2.25)-(2.34) are still not known. To determine these EV constants, we run the corresponding POD-ROM on the short time interval $[0, 15]$ with several different values for the EV constants and choose the value that yields the results that are closest to the DNS results. This approach yields the following values for the EV constants: $\alpha = 3 \times 10^{-3}$ for the ML-POD-ROM, $C_S = 0.1426$ for the S-POD-ROM, and $C_S = 0.1897$ for the VMS-POD-ROM. We emphasize that these EV constant values are optimal only on the short time interval tested, and they might actually be non-optimal on the entire time interval $[0, 300]$ on which the POD-ROMs are tested. Thus, this heuristic procedure ensures some fairness in the numerical comparison of the four POD-ROMs.

In the VMS-POD-ROM, only the first POD mode is considered as the large resolved POD mode, that is, $r_L = 1$ in (2.20). In the DS-POD-ROM, since ν_{DS} can be negative, we use a standard “clipping” procedure to ensure the numerical stability of the discretization (see for example, [37, 199]). Specifically, we let $C_S(\mathbf{x}, t) = \max\{C_S(\mathbf{x}, t), -0.2\}$. The value -0.2 is determined as follows: We first run the DS-POD-ROM without “clipping” for the time interval $[0, 15]$ and record $C_{S,ave}^-$, the average negative value of $C_S(\mathbf{x}, t)$. We then run on the entire time interval $[0, 300]$ the DS-POD-ROM with a “clipping” value $C_{S,ave}^-/2 = -0.2$. We note that there are alternative procedures to deal with the same issue in LES, such as VDSMwc [166]. We utilized the standard “clipping” procedure described above as a first step in the numerical investigation of the DS-POD-ROM.

Figure 5.1: (Continued on next page.) Snapshots of horizontal velocity at $t = 142.5 s$ for: (a) DNS; (b) the POD-G-ROM (1.12); (c) the ML-POD-ROM (2.14)-(2.15); (d) the S-POD-ROM (2.17)-(2.18); (e) the new VMS-POD-ROM (2.25)-(2.34); and (f) the new DS-POD-ROM (2.52)-(2.53). Five iso-surfaces are plotted.





5.3 Numerical Results

Before starting the quantitative comparison of the POD-ROMs, we first give a flavor of the topology of the resulting flow fields. Figure 3.10 presents the first streamwise POD mode (top left), the first normal POD mode (top right), the third streamwise POD mode (bottom left), and the third normal POD mode (bottom right). Figure 5.1 presents snapshots of horizontal velocity at $t = 142.4$ s for DNS, POD-G-ROM, ML-POD-ROM, S-POD-ROM, VMS-POD-ROM, and DS-POD-ROM. For clarity, only five iso-surfaces are drawn. Taking the DNS results as a benchmark, the POD-G-ROM seems to add unphysical structures. The ML-POD-ROM, on the other hand, appears to add too much numerical dissipation to the system and thus eliminates some of the vortical structures in the wake. The S-POD-ROM, VMS-POD-ROM, and DS-POD-ROM perform well, capturing a similar amount of structure as the DNS. It also seems that there is some phase shift for all these POD-ROMs. Due to space limitations, only one time instance snapshot is shown for the POD-ROMs. The general behavior over the entire time interval is similar; it can be found at http://www.math.vt.edu/people/wangzhu/POD_3DNumComp.html.

Physical Accuracy Figure 5.2 presents the energy spectra of the four POD-ROMs and, for comparison purposes, of the POD-G-ROM. The five energy spectra are compared with the DNS energy spectrum. All energy spectra are calculated from the average kinetic energy of the nodes in the cube with side 0.1 centered at the probe (0.9992, 0.3575, 1.0625). It is clear that the energy spectrum of the POD-G-ROM overestimates the energy spectrum of the DNS. The energy spectrum of the ML-POD-ROM, on the other hand, underestimates the the energy spectrum of the DNS, especially at the higher frequencies. The S-POD-ROM has a more accurate spectrum than the ML-POD-ROM, but it displays high oscillations at the higher frequencies. The VMS-POD-ROM is a clear improvement over the S-POD-ROM, with smaller oscillations at the higher frequencies. The energy spectrum of the DS-POD-ROM is qualitatively similar to that of the VMS-POD-ROM. The DS-POD-ROM spectrum decreases the amplitude of the high frequency oscillations of the VMS-POD-ROM even further, although it introduces some sporadic large amplitude oscillations at high frequencies. To summarize, the DS-POD-ROM and the VMS-POD-ROM yield the most accurate energy

spectra, i.e., the closest to the DNS energy spectrum. On the average, the DS-POD-ROM performs slightly better than the VMS-POD-ROM.

The second criterion in the comparison of the four POD-ROMs is represented by the mean velocity components: $\langle u \rangle$ (the mean streamwise velocity), $\langle v \rangle$ (the mean spanwise velocity), and $\langle w \rangle$ (the mean normal velocity). The time averaging is effected on the interval $[0, 300]$. Since the topology of the velocity field is markedly different in the x -, y -, and z -directions (see Figures 3.10 and 5.1), two types of spatial averaging are considered. In Figure 5.3, we consider averaging in the yz -direction ($\langle \cdot \rangle = \langle \cdot \rangle_{tyz}$) and plot the mean velocity components for different values of x , whereas in Figure 5.4, we consider averaging in the xz -direction ($\langle \cdot \rangle = \langle \cdot \rangle_{txz}$) and plot the mean velocity components for different values of y . Both figures yield the same conclusions: First, the mean streamwise velocity is computed accurately by all POD-ROMs. Second, the POD-G-ROM yields inaccurate results for the mean spanwise velocity; all the other POD-ROMs perform significantly better than the POD-G-ROM. Third, the mean normal velocity results are similar for the ML-POD-ROM, the S-POD-ROM, the VMS-POD-ROM and the DS-POD-ROM; the POD-G-ROM performs better than all these POD-ROMs over certain regions, and worse over other regions, especially when the averaging in the yz -direction is considered.

As a third criterion in the comparison of the POD-ROMs, we utilize the Reynolds stresses: $\langle u - \langle u \rangle, v - \langle v \rangle \rangle$ (the xy -component of the Reynolds stress), $\langle u - \langle u \rangle, w - \langle w \rangle \rangle$ (the xz -component of the Reynolds stress), and $\langle v - \langle v \rangle, w - \langle w \rangle \rangle$ (the yz -component of the Reynolds stress), where $\langle \cdot \rangle$ represents the temporal and spatial averaging operator. The time averaging is effected on the interval $[0, 300]$. Two types of spatial averaging are considered. In Figure 5.5, we consider averaging in the yz -direction ($\langle \cdot \rangle = \langle \cdot \rangle_{tyz}$) and plot the Reynolds stresses for different values of x , whereas in Figure 5.6, we consider averaging in the xz -direction ($\langle \cdot \rangle = \langle \cdot \rangle_{txz}$) and plot the Reynolds stresses for different values of y . Both figures yield the same two conclusions: First, the POD-G-ROM Reynolds stresses are consistently the most inaccurate (i.e., the farthest from the DNS Reynolds stresses). The second conclusion is that the ML-POD-ROM, the S-POD-ROM, the VMS-POD-ROM and the DS-POD-ROM have all similar behaviors. Indeed, different POD-ROMs might outperform the others over different spatial regions, but there is no clear “winner” over the entire spatial

interval for any of the Reynolds stresses and spatial averaging used.

The fourth criterion in the POD-ROM comparison is represented by the rms values of the velocity fluctuations: $\langle u \rangle_{rms} = \langle u - \langle u \rangle, u - \langle u \rangle \rangle$ (the rms of the streamwise velocity fluctuations), $\langle v \rangle_{rms} = \langle v - \langle v \rangle, v - \langle v \rangle \rangle$ (the rms of the spanwise velocity fluctuations), and $\langle w \rangle_{rms} = \langle w - \langle w \rangle, w - \langle w \rangle \rangle$ (the rms of the normal velocity fluctuations). The time averaging is effected on the interval $[0, 300]$. Two types of spatial averaging are considered. In Figure 5.7, we consider averaging in the yz -direction ($\langle \cdot \rangle = \langle \cdot \rangle_{tyz}$) and plot the rms values for different values of x , whereas in Figure 5.8, we consider averaging in the xz -direction ($\langle \cdot \rangle = \langle \cdot \rangle_{txz}$) and plot the rms values for different values of y . Similar to the Reynolds stresses case, the POD-G-ROM rms values of the velocity fluctuations are consistently the most inaccurate (i.e., the farthest from the DNS rms values). The rms plots corresponding to the four POD-ROMs, however, display a clear, consistent ordering this time. Indeed, the DS-POD-ROM and the VMS-POD-ROM consistently outperformed the other two POD-ROMs (the S-POD-ROM and the ML-POD-ROM), especially when the $\langle v \rangle_{rms}$ and the $\langle w \rangle_{rms}$ plots are considered. The S-POD-ROM is consistently performing worse than the DS-POD-ROM and the VMS-POD-ROM, but is clearly more accurate than the ML-POD-ROM.

As the fifth criterion in judging the performance of the POD-ROMs, the time evolutions of the POD basis coefficients $a_1(\cdot)$ and $a_4(\cdot)$ on the entire time interval $[0, 300]$ are shown in Figures 5.9-5.10. We note that the other POD coefficients have a similar behavior. Thus, for clarity of exposition, we include only $a_1(\cdot)$ and $a_4(\cdot)$. The POD-G-ROM's time evolutions of a_1 and a_4 are clearly inaccurate. Indeed, the magnitude of a_4 is nine times larger than that of the DNS projection, which indicates the need for closure modeling. The ML-POD-ROM's time evolutions of a_1 and a_4 are also inaccurate. Specifically, although the time evolution at the beginning of the simulation (where the EV constant α was chosen) is relatively accurate, the accuracy significantly degrades toward the end of the simulation. For example, the magnitude of a_4 at the end of the simulation is only one eighth of that of the DNS. The S-POD-ROM yields more accurate time evolutions than the ML-POD-ROM for both a_1 and a_4 , although the magnitude of the POD coefficients stays almost constant at a high level. The VMS-POD-ROM's time evolutions of a_1 and a_4 are better than those of the S-POD-ROM, since the magnitudes of the POD coefficients are closer to those of the DNS. Finally,

the DS-POD-ROM also yields accurate results. We note that the DS-POD-ROM's a_1 and a_4 coefficients have significantly more variability than the corresponding coefficients of the VMS-POD-ROM. This is a consequence of the fact that the EV coefficient C_S varies in time and space for the DS-POD-ROM, whereas it is constant for the VMS-POD-ROM. To summarize, the DS-POD-ROM and the VMS-POD-ROM perform the best. On the average, the DS-POD-ROM performs slightly better than the VMS-POD-ROM.

Based on the overall results (the kinetic energy spectrum, the mean velocity, the Reynolds stresses, the rms values of the velocity fluctuations, and the time evolutions of the POD basis coefficients $a_1(\cdot)$ and $a_4(\cdot)$), the DS-POD-ROM and the VMS-POD-ROM outperform the ML-POD-ROM and the S-POD-ROM. To determine which one of the DS-POD-ROM and the VMS-POD-ROM performs best, we collected the results in Figures 5.10(d)–5.10(e) (corresponding to the time evolution of the POD basis coefficient $a_4(\cdot)$ for the DNS projection, the VMS-POD-ROM and the DS-POD-ROM) and we displayed them in the same plot in Figure 5.11. Since it is difficult to distinguish between the results from the VMS-POD-ROM and the DS-POD-ROM, we zoomed in on the POD basis coefficient a_4 over the time interval [266, 282]. Based on the plot in the inset, it is clear that, for this time interval, the DS-POD-ROM performs better than the VMS-POD-ROM. More importantly, it appears that the magnitude of a_4 in the DS-POD-ROM displays some of the variability displayed by the DNS; the magnitude of the VMS-POD-ROM's a_4 coefficient, on the other hand, displays an almost periodic behavior. We believe that the variation of the DS-POD-ROM's a_4 coefficient is due to the dynamical computation of the EV coefficient, which changes both in space and time; the EV coefficient of the VMS-POD-ROM, however, is constant and is computed at the beginning of the simulation.

To summarize, the VMS and DS approaches, which are state-of-the-art closure modeling strategies in LES, yield the most accurate POD closure models for the 3D turbulent flow that we investigated. Indeed, the DS-POD-ROM and the VMS-POD-ROM clearly yield: (i) the best energy spectra (with a plus for the former, see Figure 5.2); (ii) the best rms values (see Figures 5.7 and 5.8); and (iii) the best time evolutions of the POD coefficients a_1 and a_4 (with a plus for the DS-POD-ROM, see Figures 5.9–5.11). Furthermore, with respect to the other two criteria (the mean velocity components in Figures 5.3 and 5.4 and the Reynolds

Table 5.1: Speed-up factors of POD-ROMs.

	POD-G-ROM	ML-POD-ROM	S-POD-ROM	VMS-POD-ROM	DS-POD-ROM
S_f	665	659	36	41	23

stresses in Figures 5.5 and 5.6), the DS-POD-ROM and the VMS-POD-ROM perform at least as well as the other POD-ROMs. Thus, we conclude that the DS-POD-ROM and the VMS-POD-ROM yield the most accurate average and instantaneous numerical results.

Computational Efficiency A natural question, however, is whether the new POD closure modeling strategies that we proposed display a high level of computational efficiency, which is one of the trademarks of a successful POD-ROM. To answer this question, we computed the CPU times for all four POD-ROMs and compared them with those of the DNS and the POD-G-ROM.

To make such a comparison, however, we first need to address the numerical differences between the DNS and the POD-ROMs. First, the discretizations used in the two approaches are completely different. Indeed, the spatial discretization used in the DNS was the finite volume method, whereas for the POD-ROMs we used a finite element method. Furthermore, the time-discretization used in the DNS was second-order (Crank-Nicolson and Adams-Bashforth), whereas in the POD-ROMs we used a first-order time discretization (explicit Euler). The time steps employed were also different: $\Delta t = 2 \times 10^{-3}$ in the DNS and $\Delta t = 7.5 \times 10^{-4}$ in the POD-ROM. Most importantly, the DNS was performed on a parallel machine (on 16 processors), whereas all the POD-ROM runs were carried out on a single-processor machine. Thus, to ensure a more realistic comparison between the CPU times of the DNS and the POD-ROMs, we multiplied the CPU time of the DNS by a factor of 16.

To measure the computational efficiency of the four POD-ROMs, we define the speed-up factor

$$S_f = \frac{\text{CPU time of DNS}}{\text{CPU time of POD-ROM}} \quad (5.1)$$

and list results in Table 5.1. The most efficient model is the POD-G-ROM. This is not

surprising, since no closure model is used in POD-G-ROM and thus no CPU time is spent computing an additional nonlinear term at each time step. The second most efficient model is the ML-POD-ROM. This is again natural, since only a linear closure model is employed in the ML-POD-ROM and thus the computational overhead is minimal. The speed-up factors for the S-POD-ROM, the VMS-POD-ROM and the DS-POD-ROM are one order of magnitude lower than those for the ML-POD-ROM and the POD-G-ROM. The reason is that the former use nonlinear closure models, which increase significantly the computational time. Note, however, that the S-POD-ROM, the VMS-POD-ROM and the DS-POD-ROM are still significantly more efficient than the DNS.

Sensitivity Analysis Finally, to assess the robustness of the POD-ROMs, we conduct a sensitivity study with respect to r , the number of POD modes retained in the models. To this end, we repeat the above numerical investigation of the POD-ROMs, this time, however, with *fewer* POD modes: We use $r = 4$ instead of $r = 6$, as we have used so far. The question we are trying to address is whether the conclusions drawn from the previous numerical results remain valid when r is reduced from 6 to 4. We note that, since the numerical results with spatial averaging in the xz -direction were qualitatively similar to those with spatial averaging in the yz -direction, they were not included in this dissertation.

Figure 5.12 presents the energy spectra of the POD-ROMs and the DNS. The DS-POD-ROM is clearly the most accurate model, just as it was the case for $r = 6$ (see Figure 5.2).

Figure 5.13 presents the mean velocity components of the POD-ROMs and the DNS. The same conclusions as in the $r = 6$ (see Figure 5.3) can be drawn: (i) all the POD-ROMs predict the mean streamwise velocity accurately; and (ii) all the POD-ROMs (except the POD-G-ROM) predict well the spanwise velocity component. In contrast with the $r = 6$ case, however, the POD-G-ROM is much worse than the other POD-ROMs in predicting the normal velocity component.

Figure 5.14 presents the Reynolds stresses of the POD-ROMs and the DNS. The same conclusions as in the $r = 6$ (see Figure 5.5) can be drawn: (i) the POD-G-ROM yields inaccurate results; and (ii) all other POD-ROMs perform similarly.

Figure 5.15 presents the rms values of the velocity fluctuations of the POD-ROMs and

the DNS. As in the $r = 6$ case (see Figure 5.7), the POD-G-ROM yields very inaccurate results. There is, however, a difference between the results for $r = 4$ and those for $r = 6$. Indeed, the most accurate results in Figure 5.15 are those for the DS-POD-ROM (as in the $r = 6$ case), but the ML-POD-ROM performs better than the VMS-POD-ROM this time. The S-POD-ROM performs badly, just as in the $r = 6$ case.

Figures 5.16 and 5.17 present the time evolutions of the POD basis coefficients $a_1(\cdot)$ and $a_4(\cdot)$ on the entire time interval $[0, 300]$. As in the $r = 6$ case (see Figure 5.9), the DS-POD-ROM is the most accurate model in predicting the time evolution of $a_1(\cdot)$, followed (in this order) by the ML-POD-ROM, the VMS-POD-ROM, the S-POD-ROM, and the POD-G-ROM (which is the most inaccurate model). The ordering for the time evolution of $a_4(\cdot)$, however, is different: The most accurate models seem to be the VMS-POD-ROM and the S-POD-ROM, followed by the DS-POD-ROM, the ML-POD-ROM, and the POD-G-ROM (which is again the most inaccurate model).

To summarize, the same general conclusions as in the $r = 6$ case can be drawn in the $r = 4$ case: Overall, the DS-POD-ROM and the VMS-POD-ROM yield the most accurate average and instantaneous numerical results. Furthermore, with a few exceptions, the POD-ROMs results display a relatively low sensitivity with respect to changes in r , the number of POD modes employed in the models.

5.4 Summary

To assess the performance of four POD closure models, a structurally dominated 3D turbulent flow problem at $\text{Re} = 1000$ was considered. The POD basis ($r = 4, 6$) was generated from the snapshots on the time interval $[0, 75]$, however, numerical simulations of the POD-ROMs were implemented over a larger time interval $[0, 300]$ (i.e., $t_0 = 0$, $\Delta_t = 75$, and $C_1 = 4$ in the “perfect” POD-ROM defined in Section 1.6, see Figure 1.1). Based on the numerical results, the following general conclusions were drawn: Both the POD-G-ROM and the ML-POD-ROM yielded inaccurate results. The DS-POD-ROM and the VMS-POD-ROM clearly outperformed these two models, yielding more accurate results. The DS-POD-ROM generally performed slightly better than the VMS-POD-ROM and seemed to display more

adaptivity in terms of adjusting the magnitude of the POD basis coefficients. Overall, however, the two models yielded similar qualitative results. This seems to reflect the LES setting, where both the DS and the VMS closure modeling strategies are considered state-of-the-art [114, 115]. The DS-POD-ROM and the VMS-POD-ROM, although not as computationally efficient as the POD-G-ROM or the ML-POD-ROM, significantly decreased the CPU time of the DNS. Finally, to assess the robustness of the POD-ROMs, we conducted a sensitivity study with respect to r , the number of POD modes retained in the models. To this end, we repeated the above numerical investigation of the POD-ROMs, this time, however, using $r = 4$ instead of $r = 6$. The same general conclusions as in the $r = 6$ were drawn in the $r = 4$ case: Overall, the DS-POD-ROM and the VMS-POD-ROM yielded the most accurate average and instantaneous numerical results. Furthermore, with a few exceptions, the POD-ROMs results displayed a relatively low sensitivity with respect to changes in r .

To summarize, for the 3D turbulent flow that we investigated, the DS-POD-ROM and the VMS-POD-ROM were found to perform the best among the POD-ROMs investigated, combining a relative high numerical accuracy with a high level of computational efficiency.

Figure 5.2: Kinetic energy spectrum of the DNS (blue) and the POD-ROMs (red) for $r = 6$: (a) the POD-G-ROM (1.12); (b) the ML-POD-ROM (2.14)-(2.15); (c) the S-POD-ROM (2.17)-(2.18); (d) the new VMS-POD-ROM (2.25)-(2.34); and (e) the new DS-POD-ROM (2.52)-(2.53).

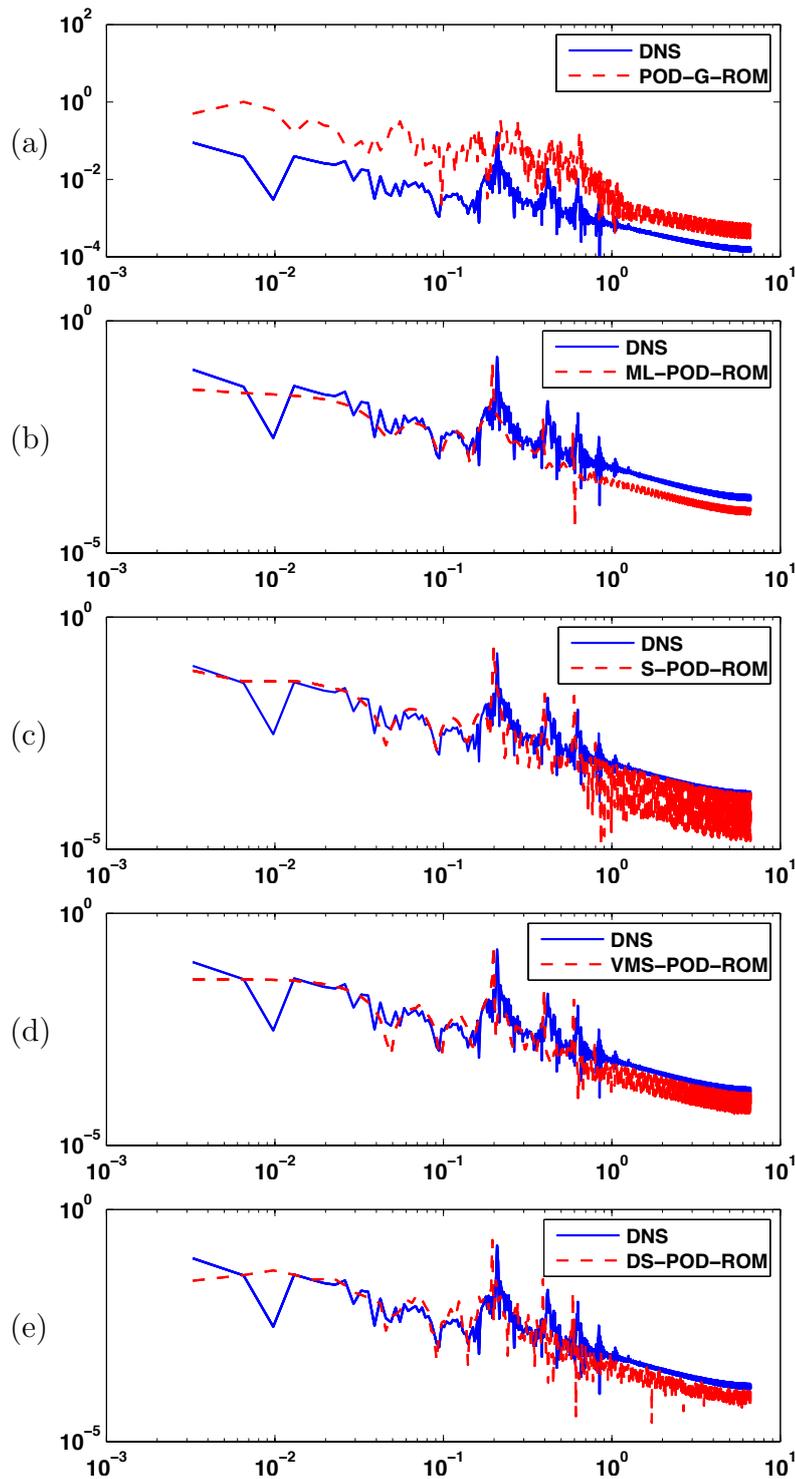


Figure 5.3: Mean velocity components for DNS and POD-ROMs when $r = 6$: (a) $\langle u \rangle$, (b) $\langle v \rangle$, and (c) $\langle w \rangle$, where $\langle \cdot \rangle = \langle \cdot \rangle_{tyz}$.

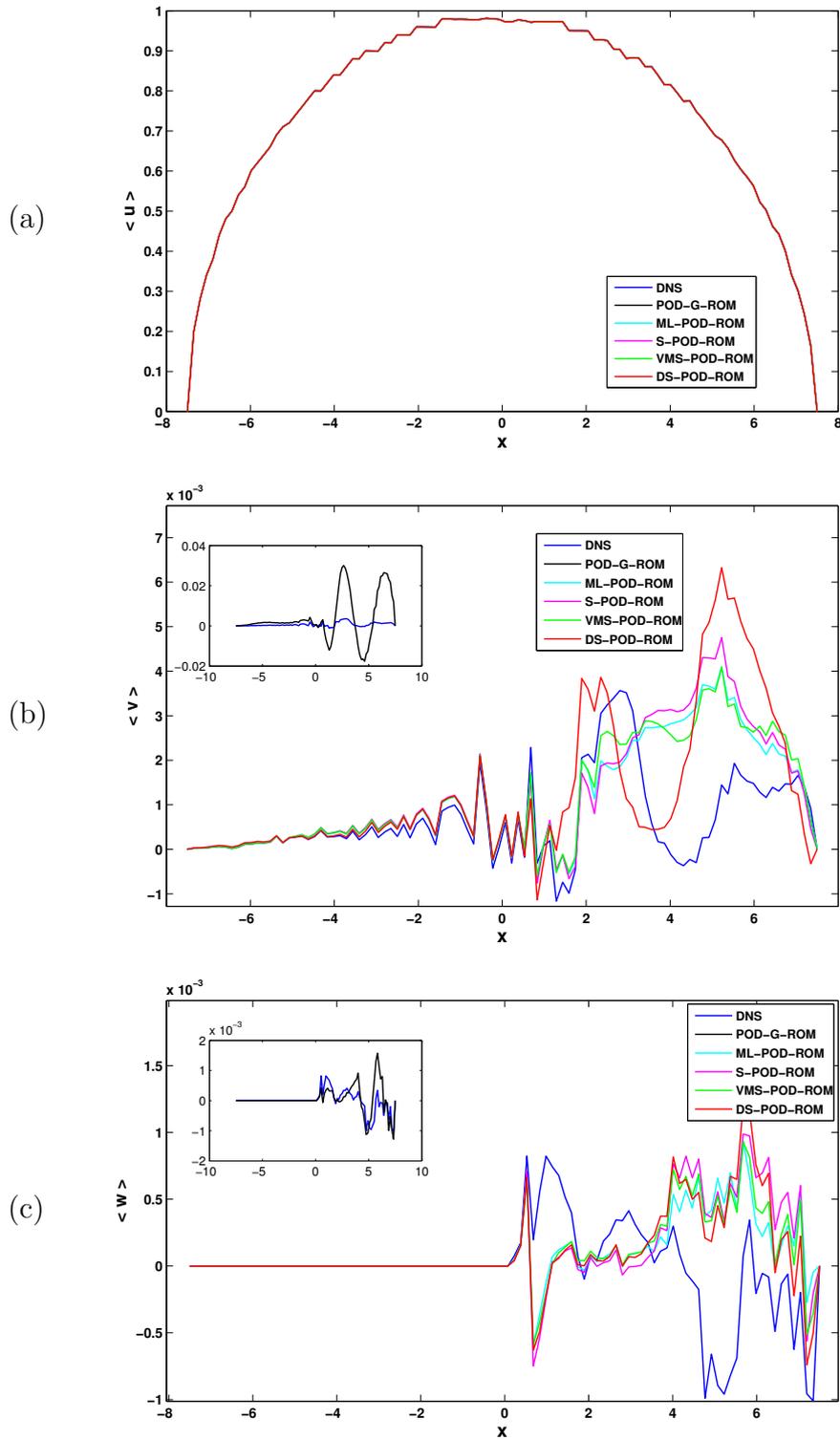


Figure 5.4: Mean velocity components for DNS and POD-ROMs when $r = 6$: (a) $\langle u \rangle$, (b) $\langle v \rangle$, and (c) $\langle w \rangle$, where $\langle \cdot \rangle = \langle \cdot \rangle_{txz}$.

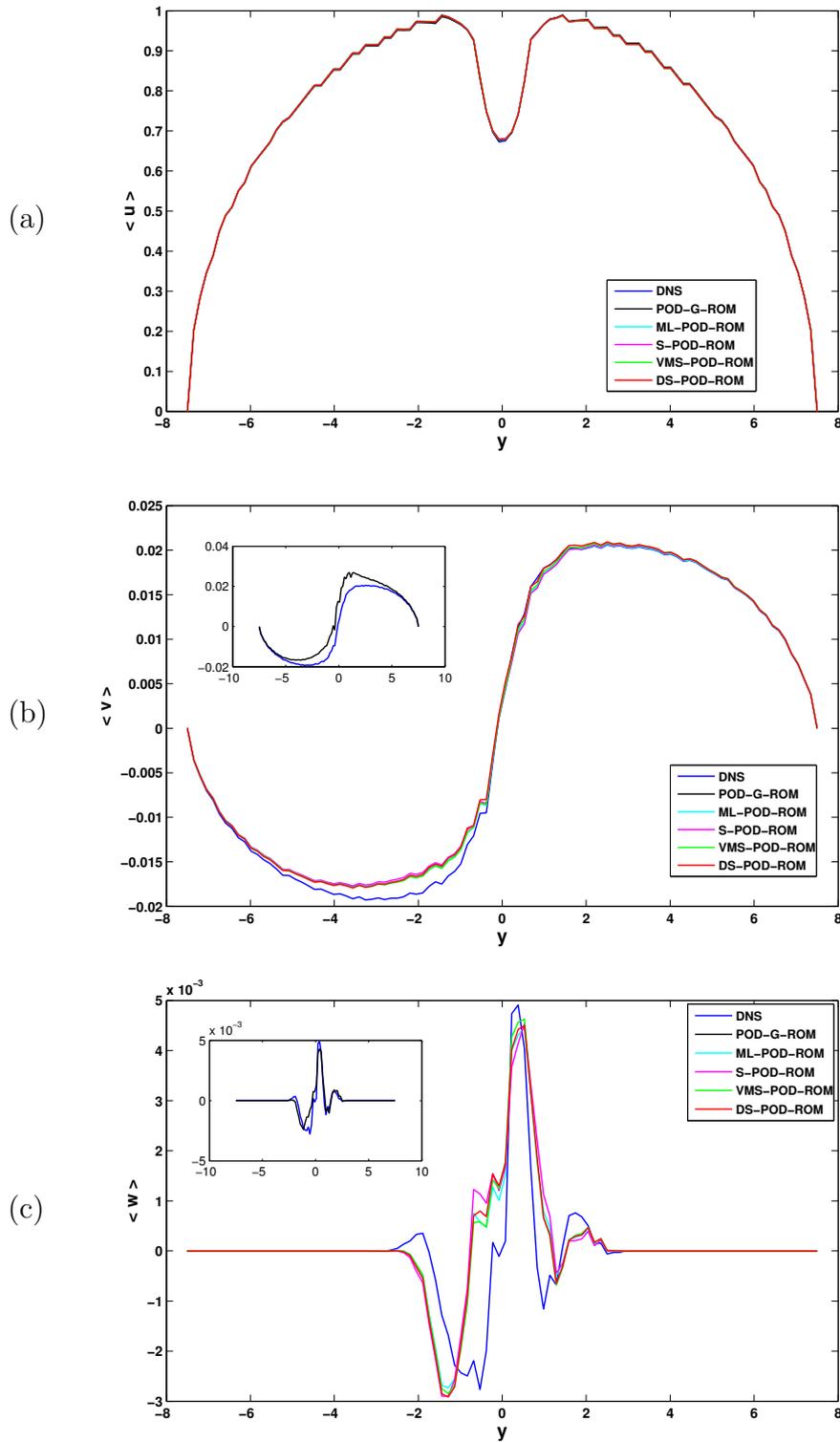


Figure 5.5: Reynolds stresses for DNS and POD-ROMs when $r = 6$: (a) $\langle u - \langle u \rangle, v - \langle v \rangle \rangle$, (b) $\langle u - \langle u \rangle, w - \langle w \rangle \rangle$, and (c) $\langle v - \langle v \rangle, w - \langle w \rangle \rangle$, where $\langle \cdot \rangle = \langle \cdot \rangle_{tyz}$.

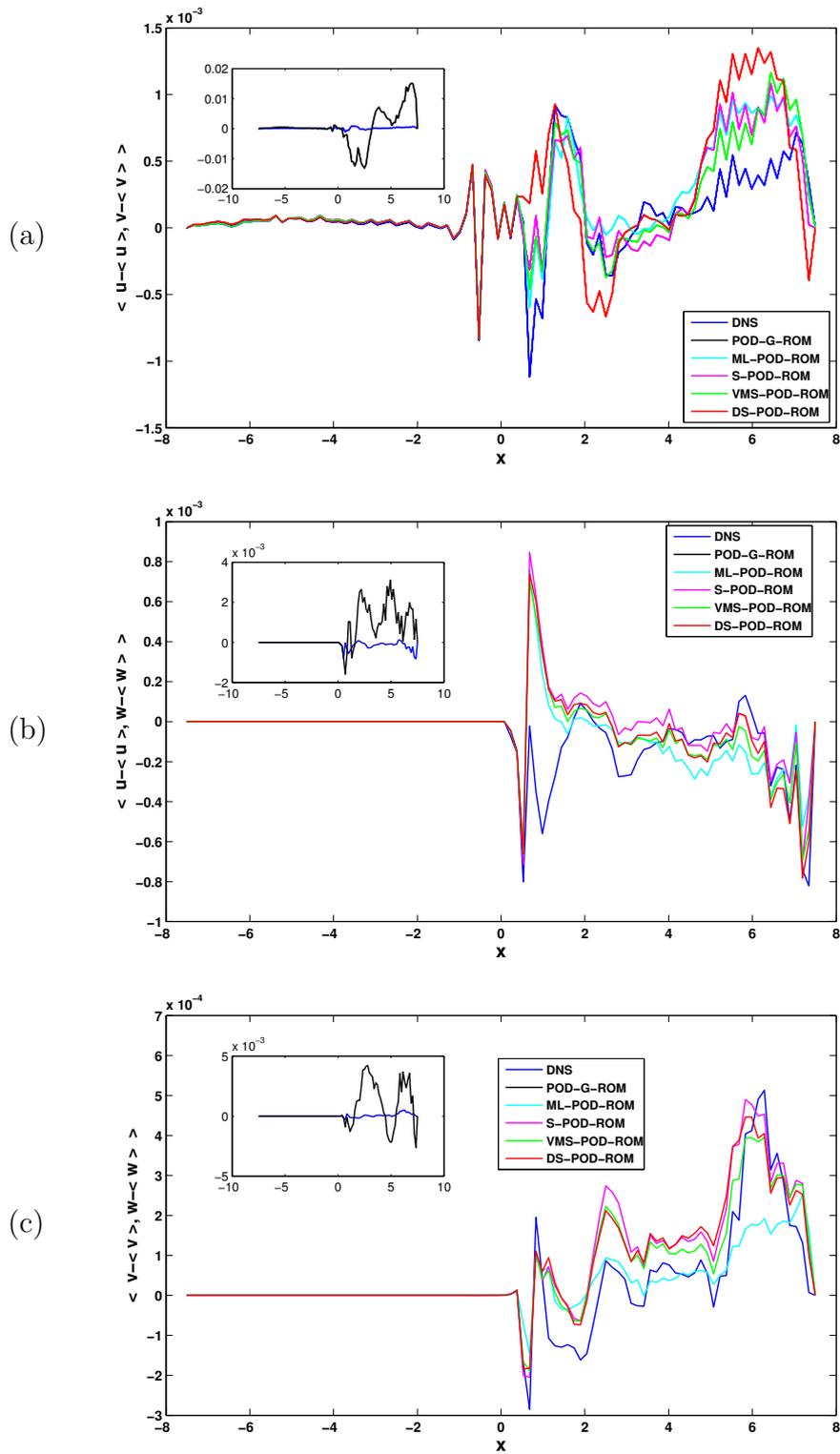


Figure 5.6: Reynolds stresses for DNS and POD-ROMs when $r = 6$: (a) $\langle u - \langle u \rangle, v - \langle v \rangle \rangle$, (b) $\langle u - \langle u \rangle, w - \langle w \rangle \rangle$, and (c) $\langle v - \langle v \rangle, w - \langle w \rangle \rangle$, where $\langle \cdot \rangle = \langle \cdot \rangle_{txz}$.

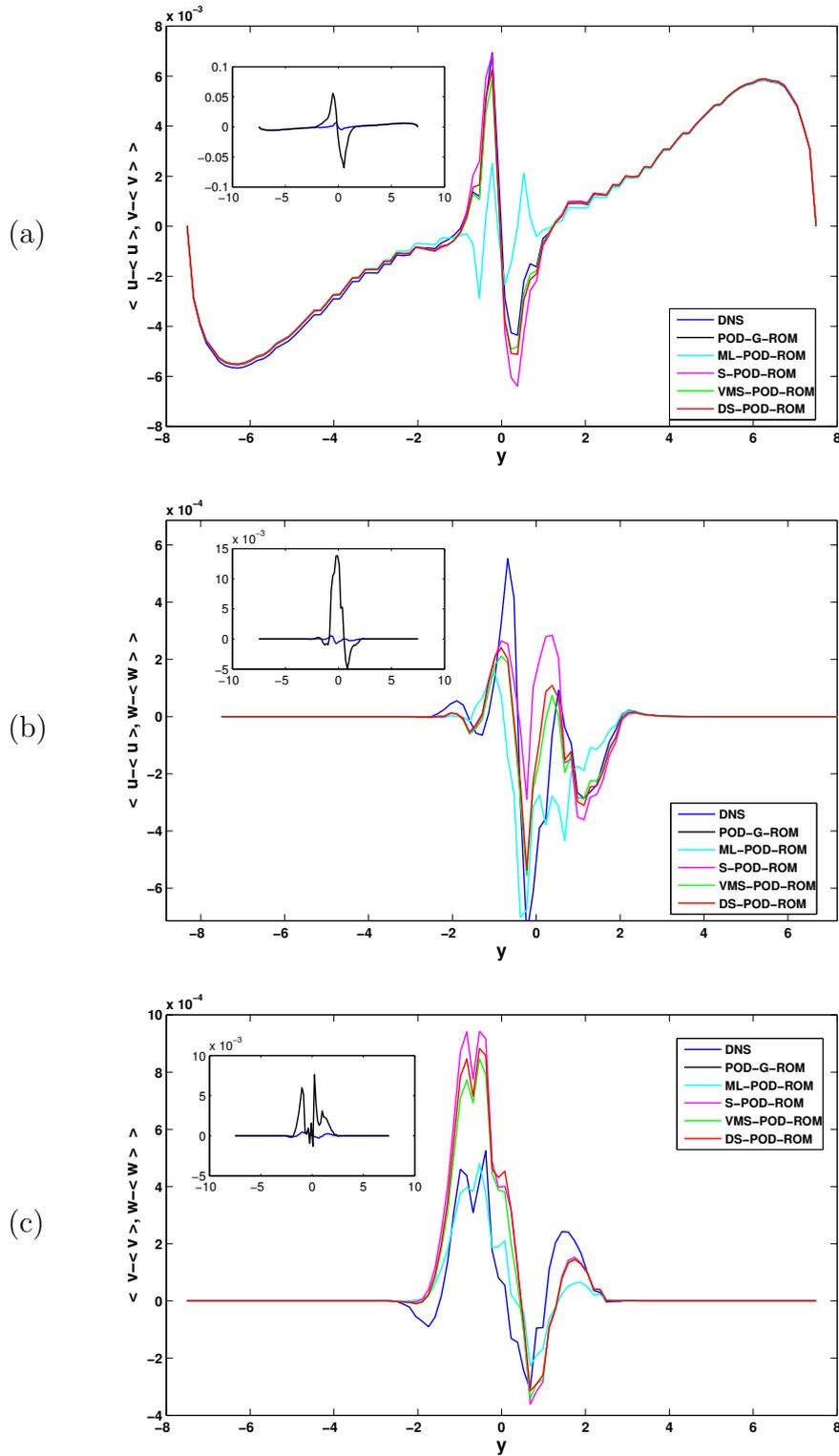


Figure 5.7: Rms values of the velocity fluctuations for DNS and POD-ROMs when $r = 6$: (a) $\langle u \rangle_{rms} = \langle u - \langle u \rangle, u - \langle u \rangle \rangle$, (b) $\langle v \rangle_{rms} = \langle v - \langle v \rangle, v - \langle v \rangle \rangle$, and (c) $\langle w \rangle_{rms} = \langle w - \langle w \rangle, w - \langle w \rangle \rangle$, where $\langle \cdot \rangle = \langle \cdot \rangle_{tyz}$.

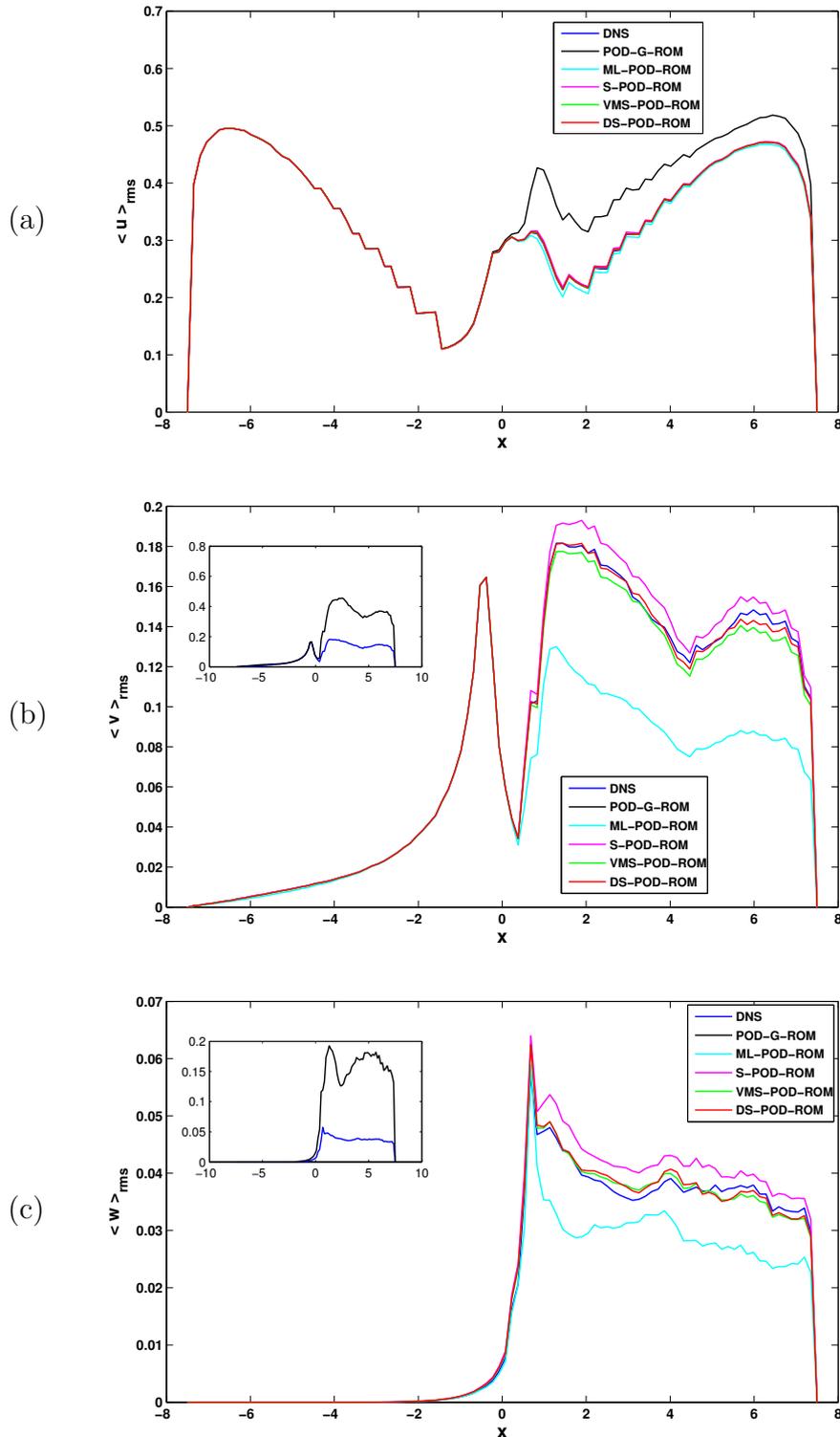


Figure 5.8: Rms values of the velocity fluctuations for DNS and POD-ROMs when $r = 6$: (a) $\langle u \rangle_{rms} = \langle u - \langle u \rangle, u - \langle u \rangle \rangle$, (b) $\langle v \rangle_{rms} = \langle v - \langle v \rangle, v - \langle v \rangle \rangle$, and (c) $\langle w \rangle_{rms} = \langle w - \langle w \rangle, w - \langle w \rangle \rangle$, where $\langle \cdot \rangle = \langle \cdot \rangle_{txz}$.

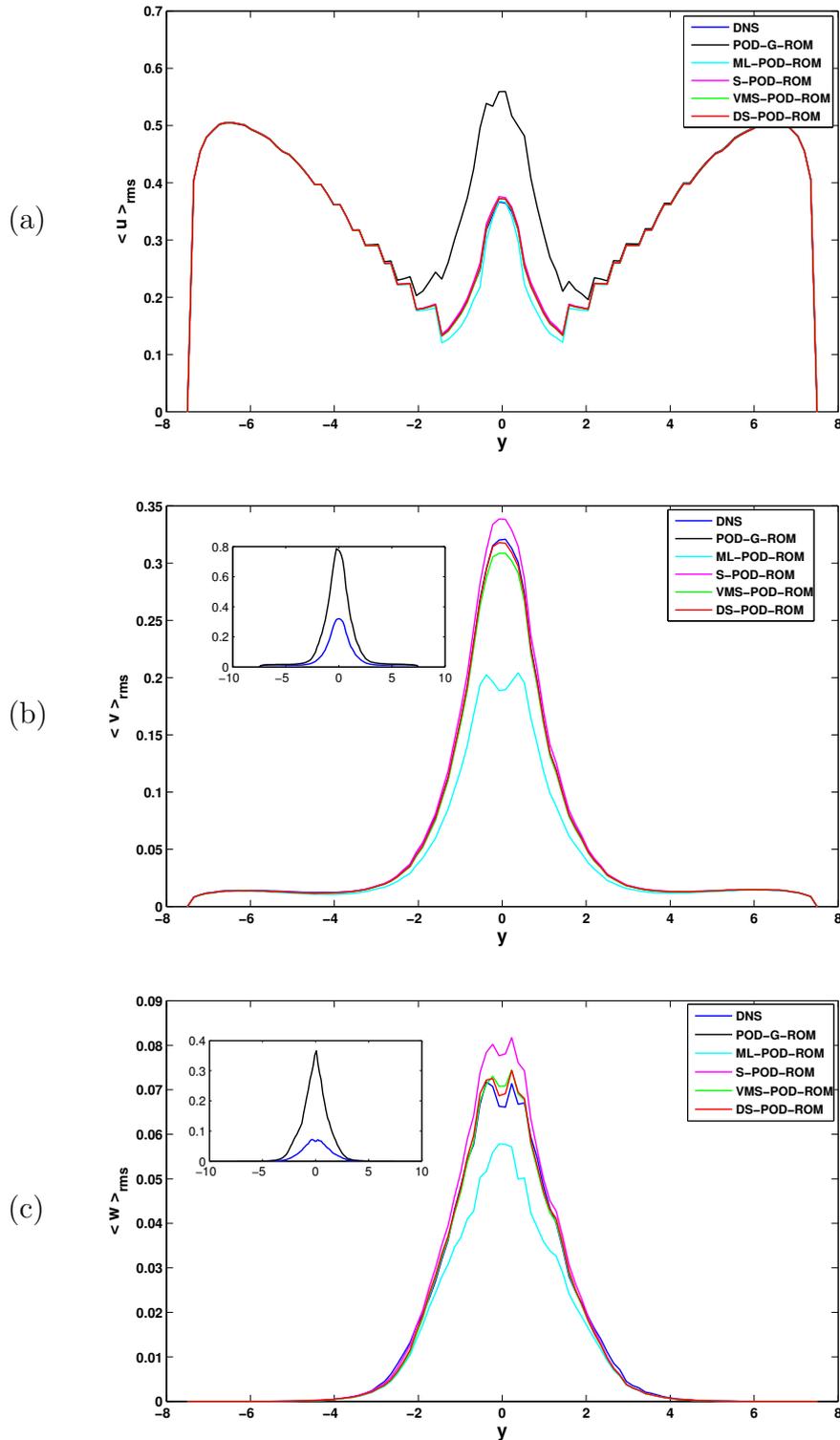


Figure 5.9: Time evolutions of the POD basis coefficient a_1 of the DNS (blue) and the POD-ROMs (red) for $r = 6$: (a) the POD-G-ROM (1.12); (b) the ML-POD-ROM (2.14)-(2.15); (c) the S-POD-ROM (2.17)-(2.18); (d) the new VMS-POD-ROM (2.25)-(2.34); and (e) the new DS-POD-ROM (2.52)-(2.53).

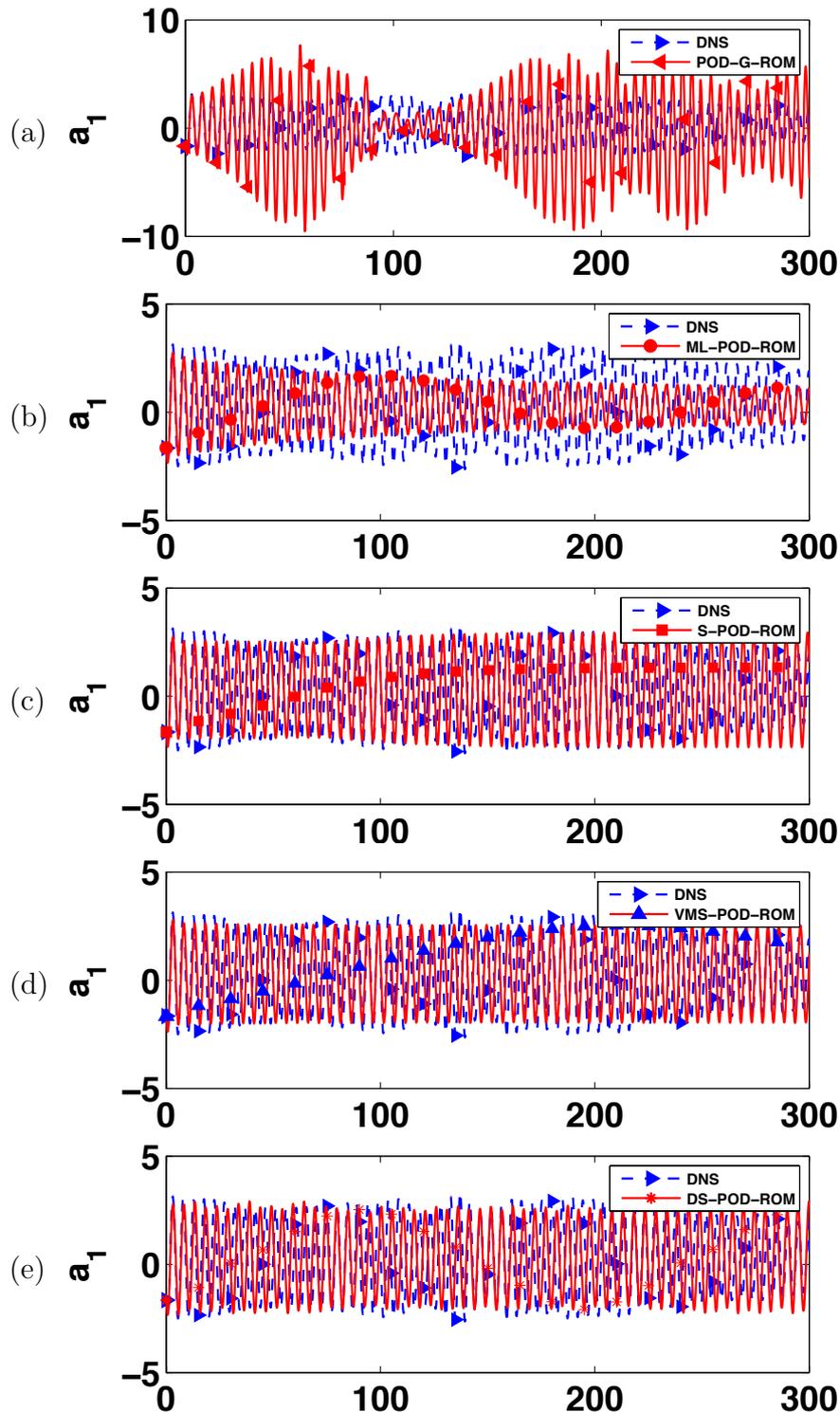


Figure 5.10: Time evolutions of the POD basis coefficient a_4 of the DNS (blue) and the POD-ROMs (red) for $r = 6$: (a) the POD-G-ROM (1.12); (b) the ML-POD-ROM (2.14)-(2.15); (c) the S-POD-ROM (2.17)-(2.18); (d) the new VMS-POD-ROM (2.25)-(2.34); and (e) the new DS-POD-ROM (2.52)-(2.53).

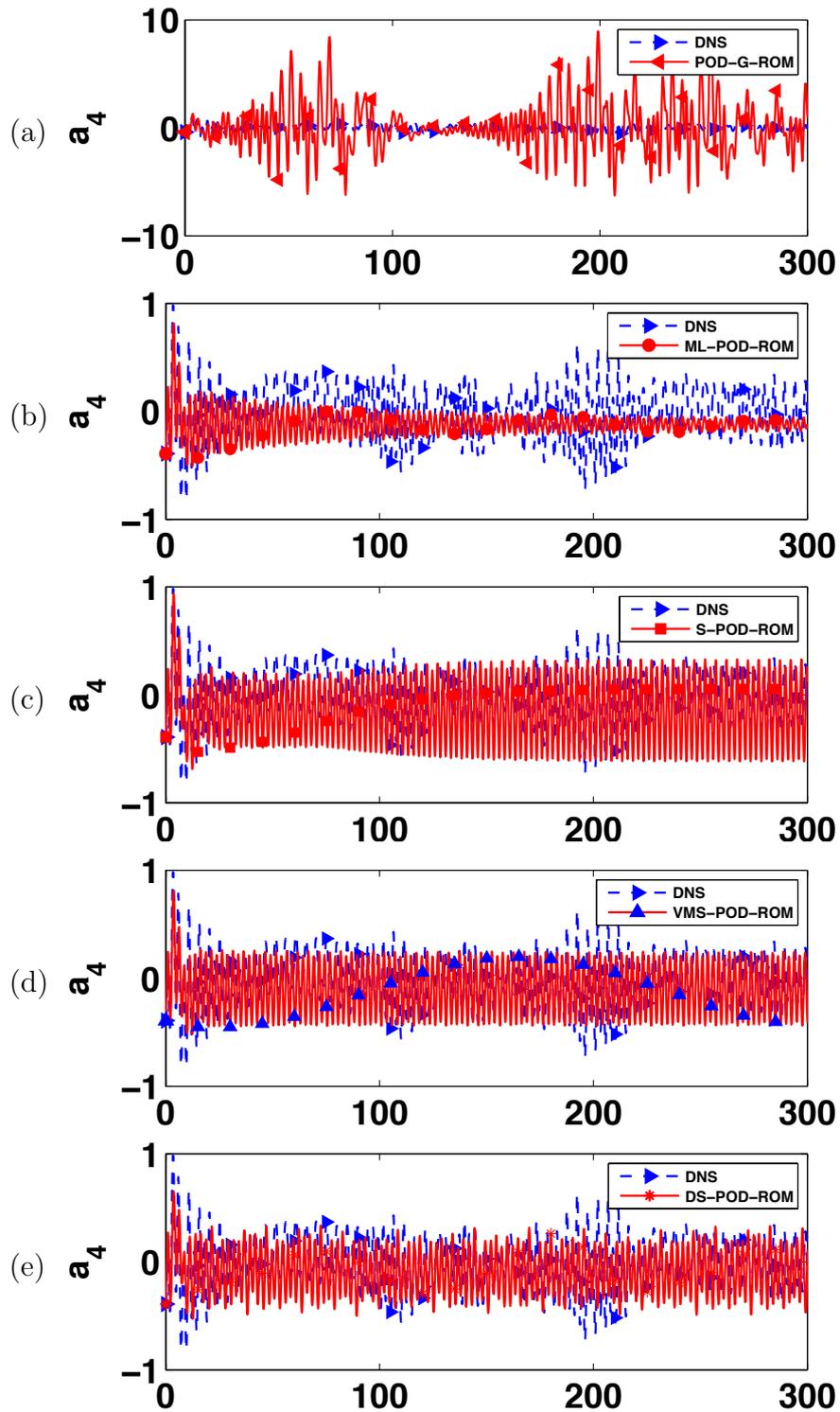


Figure 5.11: Time evolutions of the POD basis coefficient a_4 of the DNS (green), the new VMS-POD-ROM (2.25)-(2.34) (blue), and the new DS-POD-ROM (2.52)-(2.53) (red) when $r = 6$.

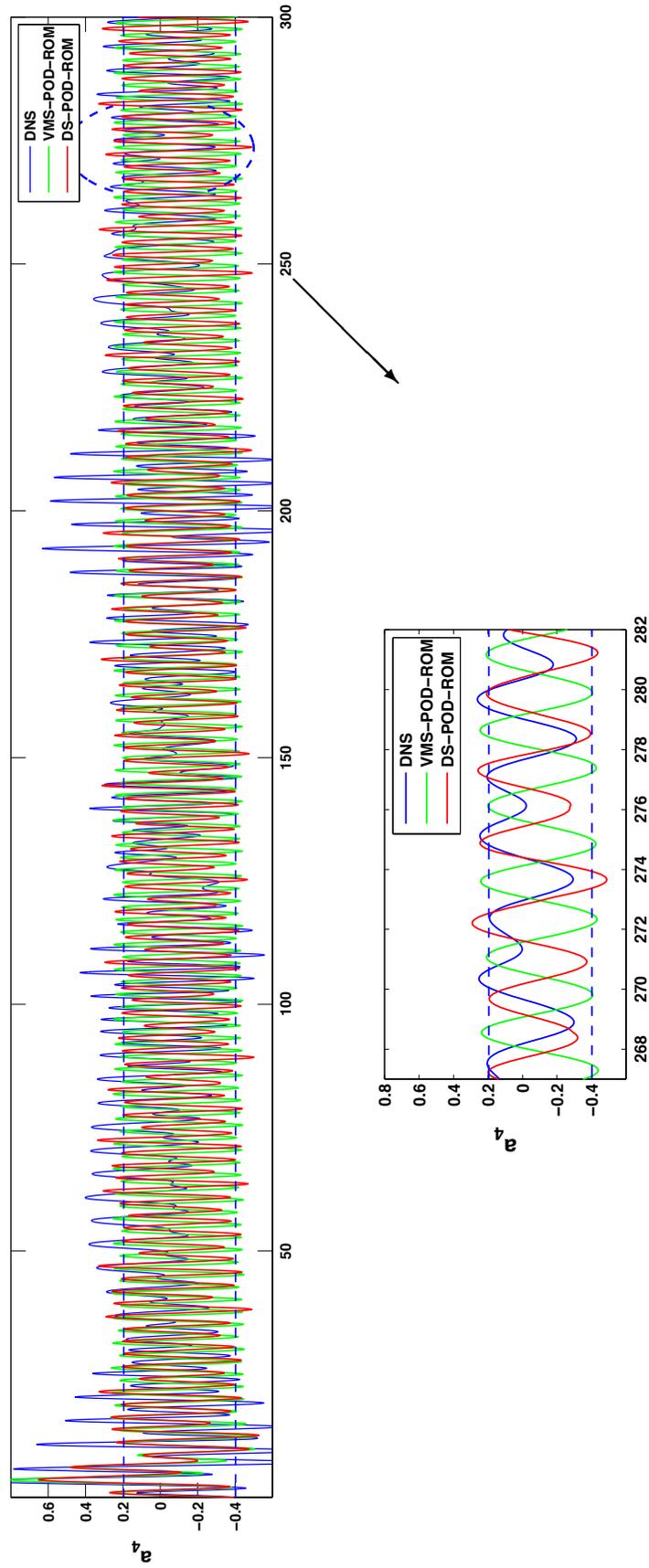


Figure 5.12: Kinetic energy spectrum of the DNS (blue) and the POD-ROMs (red) for $r = 4$.

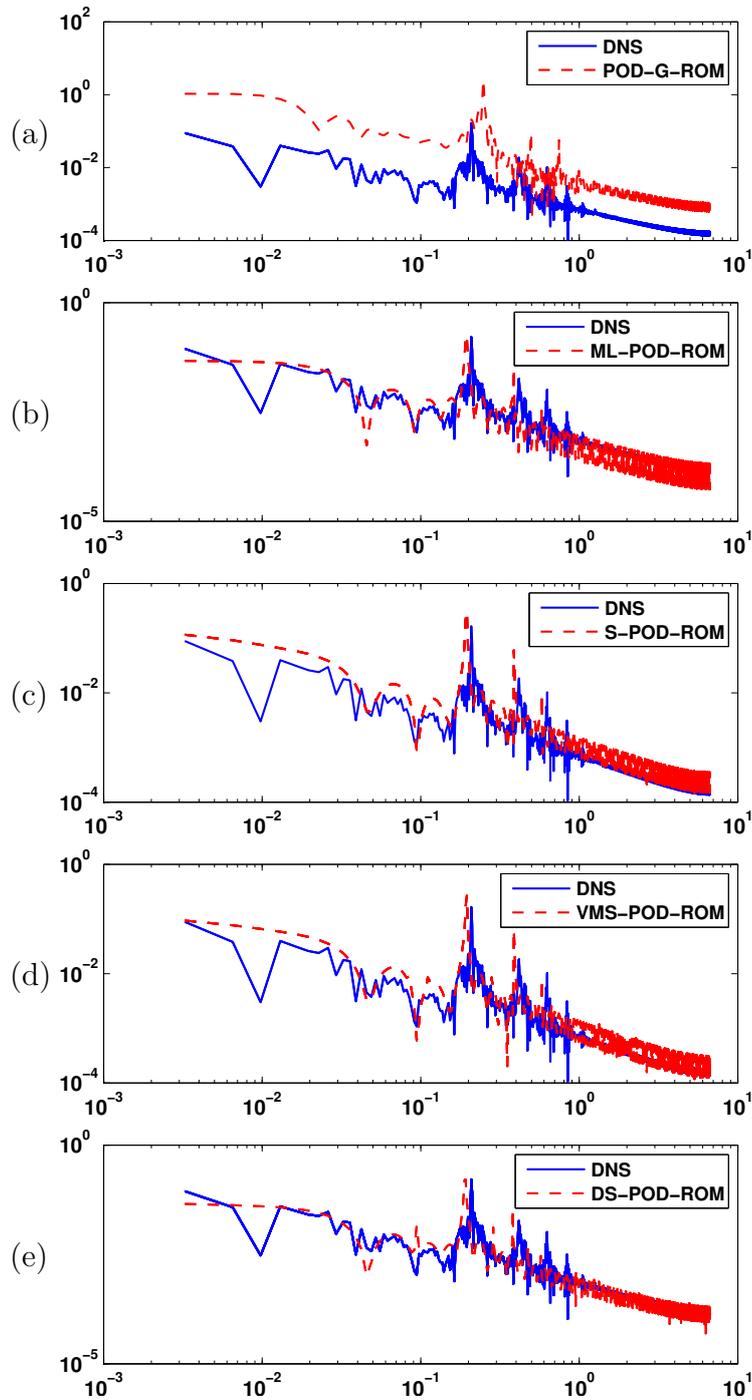


Figure 5.13: Mean velocity components for DNS and POD-ROMs when $r = 4$.

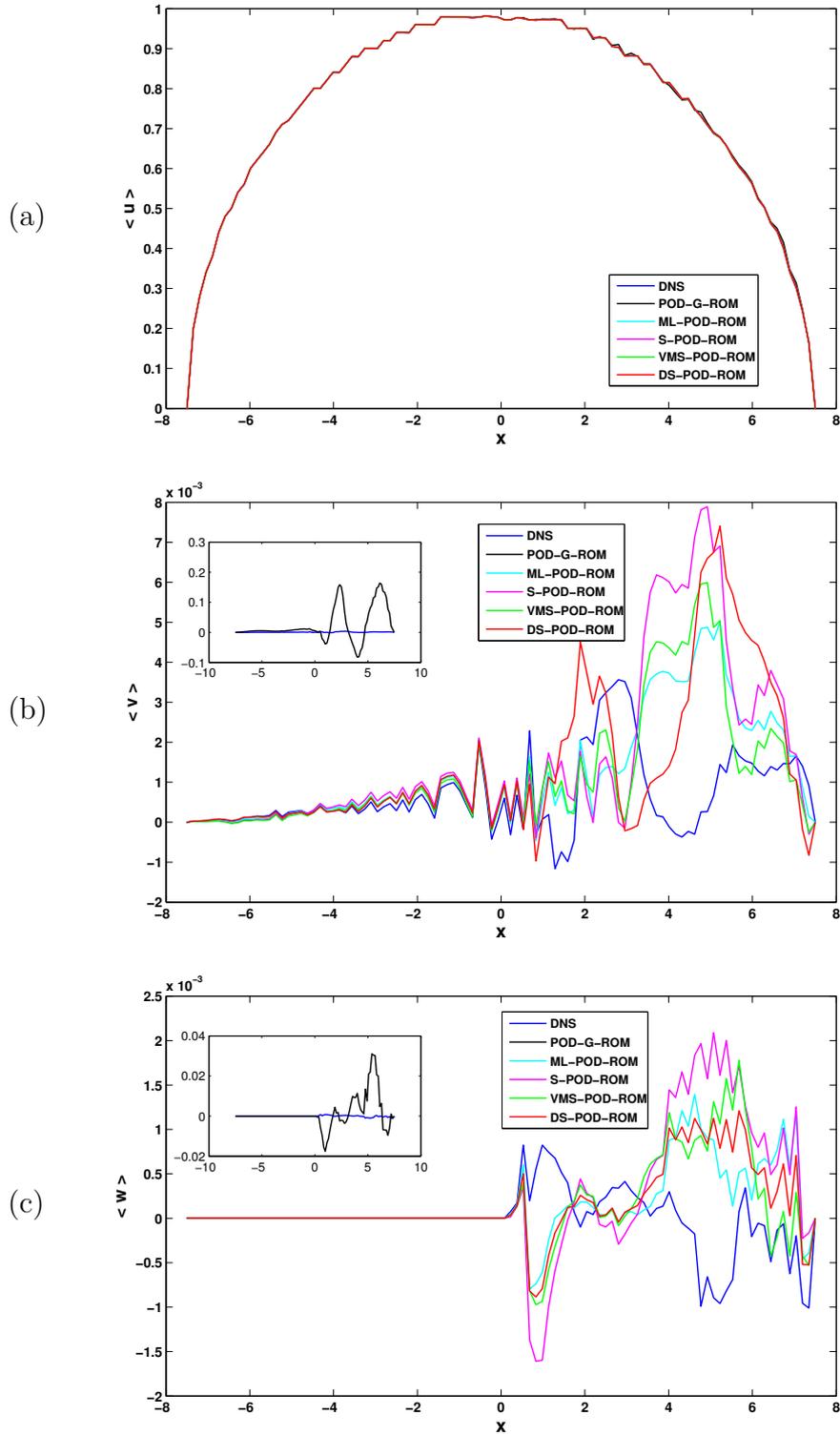


Figure 5.14: Reynolds stresses for DNS and POD-ROMs when $r = 4$.

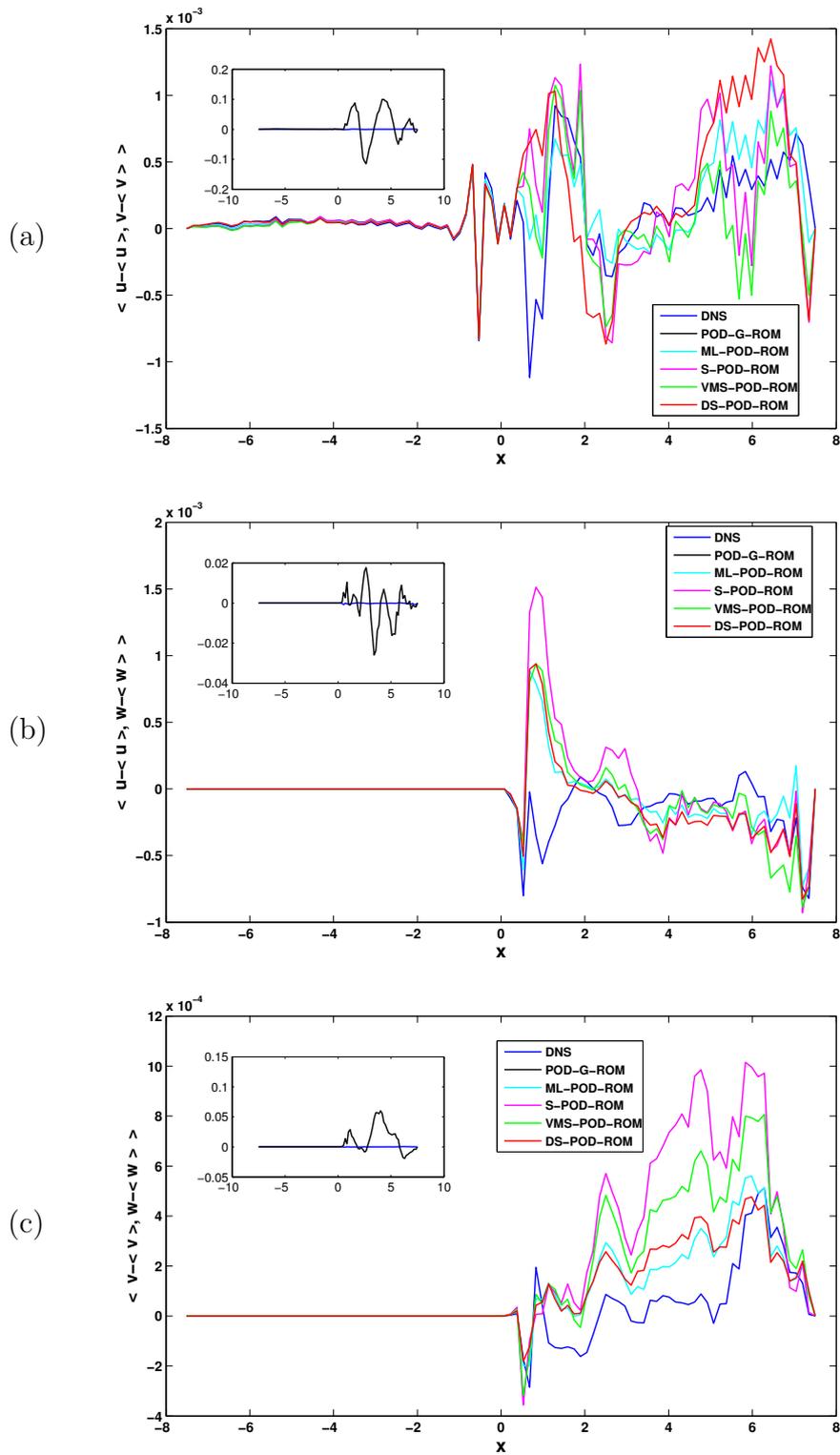


Figure 5.15: Rms values of the velocity fluctuations for DNS and POD-ROMs when $r = 4$.

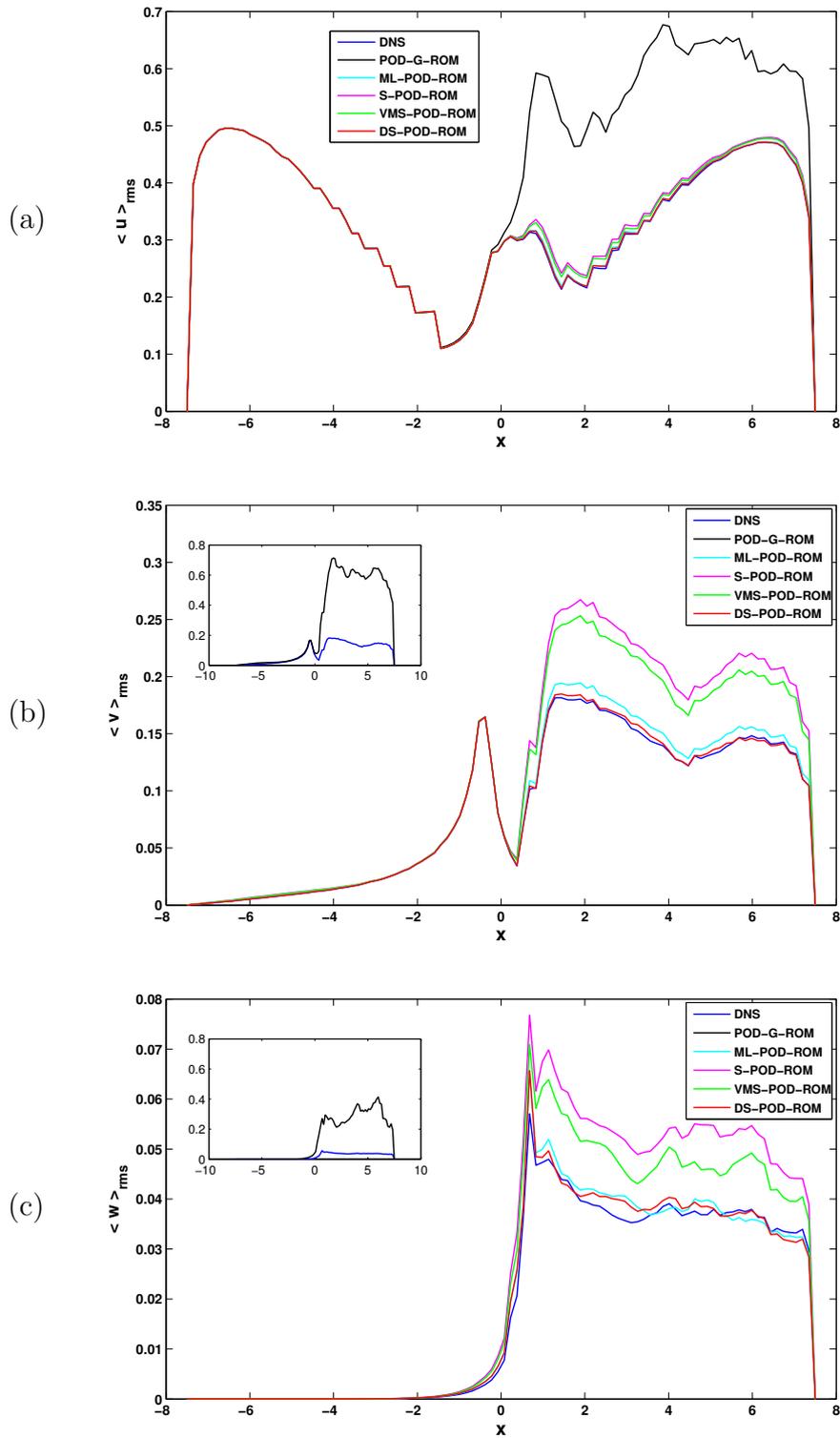


Figure 5.16: Time evolutions of the POD basis coefficient a_1 of the DNS (blue) and the POD-ROMs (red) for $r = 4$.

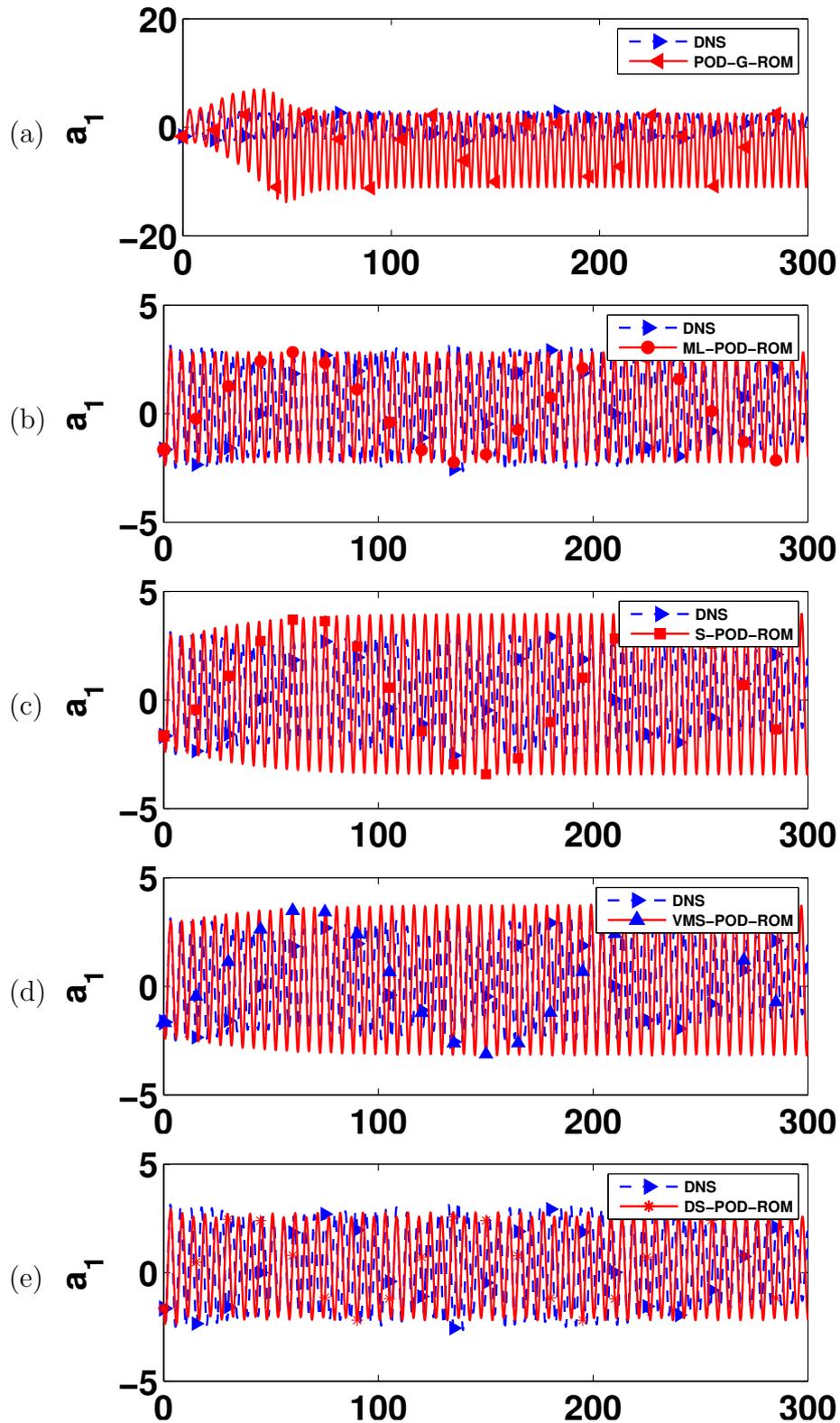
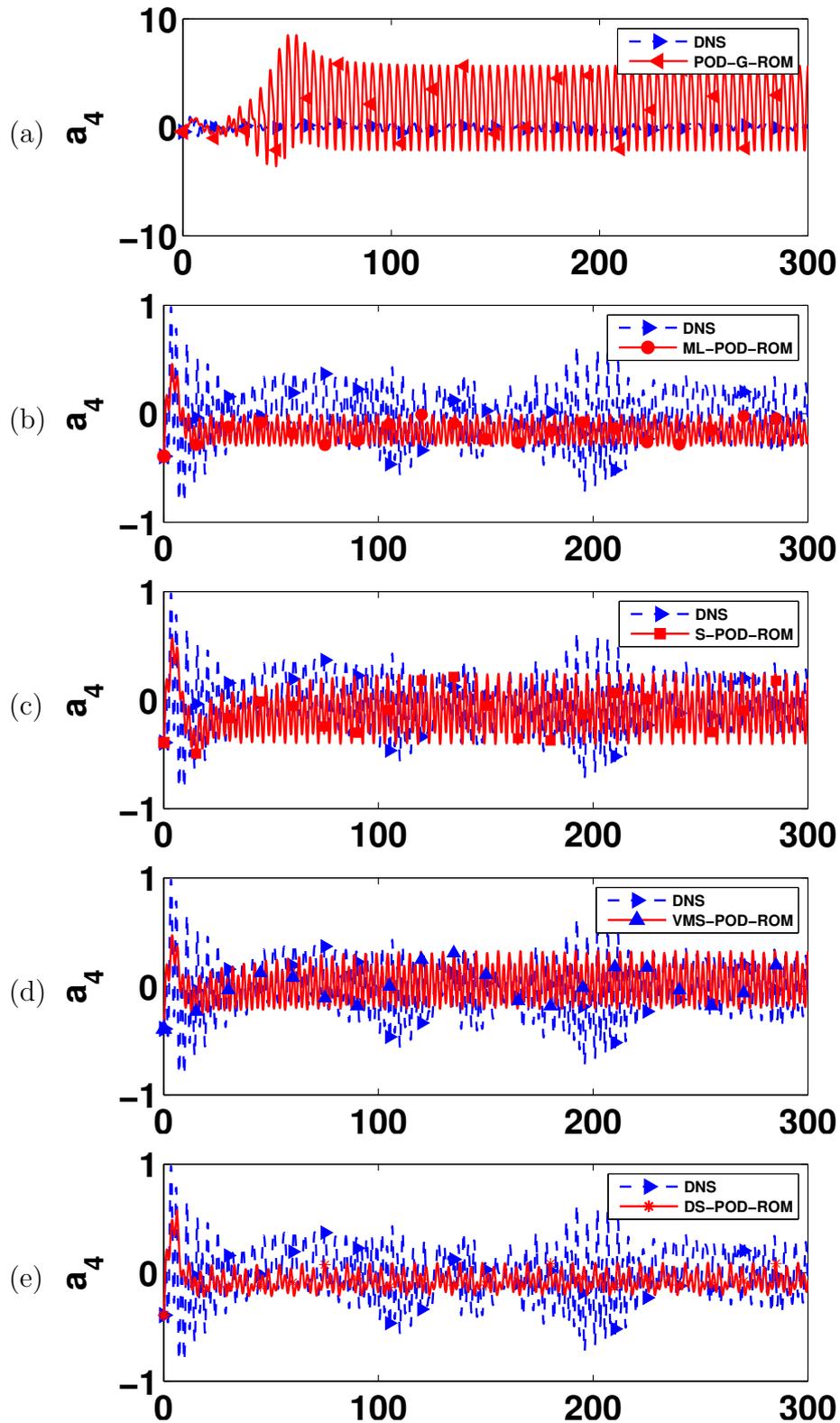


Figure 5.17: Time evolutions of the POD basis coefficient a_4 of the DNS (blue) and the POD-ROMs (red) for $r = 4$.



Chapter 6

Future Work [†]

6.1 Introduction

Climate modeling as well as optimization and control of airflow in energy efficient buildings represent computational grand challenges at the frontier of science and engineering. Indeed, a brute force computational approach is clearly impractical for both applications when realistic parameters are used. POD-ROMs represent a natural alternative, since they combine computational efficiency with physical fidelity. The current limitations of this approach are, of course, developing appropriate closure models for the POD-ROMs, efficient numerical discretizations for these inherently nonlinear models, and updating POD bases when needed.

In this chapter, we present our first steps in these directions. Our preliminary results suggest that our approach is promising, but much more remains to be done. The rest of this

[†]The material in this chapter is based on the following reports:

O. San, A. E. Staples, Z. Wang and T. Iliescu. Approximate Deconvolution Large Eddy Simulation of a Barotropic Ocean Circulation Model. *Ocean Modelling*, 40, 2011, pp. 120-132.

E. Foster, T. Iliescu, and Z. Wang. A Finite Element Discretization of the Streamfunction Formulation of the Stationary Quasi-Geostrophic Equations of the Ocean. Submitted, 2012.

Z. Wang and T. Iliescu. A Mixed Finite Element Method for the Time-Dependent Quasi-Geostrophic Equations of the Ocean. Submitted, 2012.

Z. Wang, J. Borggaard and T. Iliescu. Reduced-Order Modeling of Airflow in Energy-Efficient Buildings. In preparation, 2012.

chapter is organized as follows: In Section 6.2, we discuss energy efficient building airflow simulation problems. In Section 6.3, we consider large-scale ocean currents simulation in climate modeling. Both represent applications of our ongoing research for complex flows. Several research avenues we plan to pursue in the future are discussed in Section 6.4.

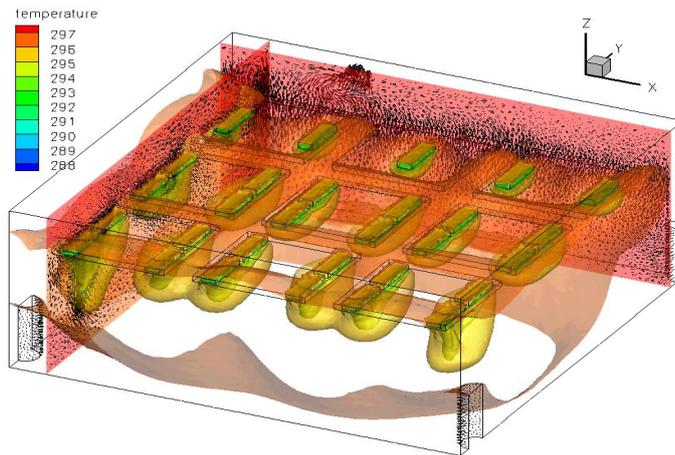


Figure 6.1: Reduced-order modeling of airflow in the DOE energy-efficient buildings system design hub.

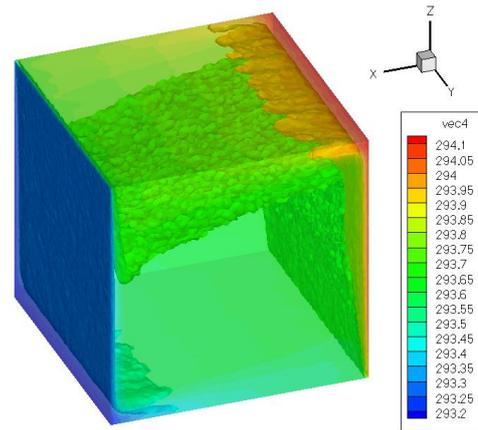


Figure 6.2: Mean temperature of airflow in a simple geometry.

6.2 Energy Efficient Building Airflow Simulations

Motivation Buildings consume 39% of total energy in U.S. (source: U.S. Department of Energy (DOE), 2008 Buildings Energy Data Book [73], Section 1.1.1, 2008). Thus they have a significant impact on the national energy future. To achieve a significant reduction (e.g., 70-80%) in energy consumption, a modern mathematical and computational framework needs to be employed in the building design. One natural avenue is to utilize advanced optimization and control tools to achieve the same room temperature while consuming much less heating and cooling energy. However, the structural, spatial and temporal complexity of the system makes a brute force computational approach in the optimization and control impractical. Therefore, simplified models are needed. POD-ROMs are considered here as a way to significantly reduce the dimension of the system, and, thus, to make the real time dynamic control of airflow for energy efficient buildings feasible.

POD-G-ROM for the Boussinesq Equations A commonly used mathematical model for the airflows is the *Boussinesq equations* [25, 74]:

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\nabla p + \text{Re}^{-1} \Delta \mathbf{u} + \text{Gr} \text{Re}^{-2} \mathbf{g}(T - T_\infty) \\ \nabla \cdot \mathbf{u} = 0 \\ \frac{\partial T}{\partial t} + (\mathbf{u} \cdot \nabla) T = (\text{Re Pr})^{-1} \Delta T \end{cases} . \quad (6.1)$$

Note that this is a PDE system in which the NSE and a thermal energy equation are coupled. In (6.1), \mathbf{u} is the velocity, p is the pressure, T is the temperature, T_∞ is the reference temperature, Re is the Reynolds number, Gr is the Grashof number, Pr is the Prandtl number and \mathbf{g} is the gravitational acceleration vector. The Grashof number measures the ratio of buoyancy to the viscous force. For natural convection from a vertical flat plate, the transition to turbulent flow occurs in the range $10^8 < \text{Gr} < 10^9$. At a Grashof number within or beyond this range, the boundary layer becomes turbulent. The Prandtl number measures the viscous diffusion effect with respect to the thermal diffusion effect. A low value of Prandtl number implies a strong conductive transfer. For airflow, Pr is around $0.7 - 0.8$.

A POD-G-ROM of the airflow is constructed from the POD basis by writing

$$\begin{cases} \mathbf{u}(x, t) \approx \mathbf{u}_r(x, t) = \mathbf{U} + \sum_{j=1}^r a_j(t) \boldsymbol{\varphi}_j(x) \\ T(x, t) \approx T_r(x, t) = T_\infty + \sum_{j=1}^r b_j(t) \psi_j(x) \end{cases} , \quad (6.2)$$

where $\mathbf{X}_u^r = \{\boldsymbol{\varphi}_1(x), \dots, \boldsymbol{\varphi}_r(x)\}$ and $X_T^r = \{\psi_1(x), \dots, \psi_r(x)\}$ are the POD basis functions for the velocity \mathbf{u} and temperature T , respectively, \mathbf{U} is the central trajectory for the velocity, and $\{a_j(t)\}_{j=1}^r$ and $\{b_j(t)\}_{j=1}^r$ are coefficient functions to be computed. In general, the number of POD modes in \mathbf{X}_u^r and X_T^r could be different. For simplicity, we just choose them to be same.

The POD-G-ROM is obtained by replacing the primitive variables in (6.1) with their POD approximations \mathbf{u}_r and T_r , then projecting (6.1) onto the POD basis sets:

$$\begin{cases} \left(\frac{\partial \mathbf{u}_r}{\partial t}, \boldsymbol{\varphi} \right) = - \left((\mathbf{u}_r \cdot \nabla) \mathbf{u}_r, \boldsymbol{\varphi} \right) - \text{Re}^{-1} \left(\nabla \mathbf{u}_r, \nabla \boldsymbol{\varphi} \right) + \text{Gr} \text{Re}^{-2} \mathbf{g} (T_r - T_\infty, \boldsymbol{\varphi}) \quad \forall \boldsymbol{\varphi} \in \mathbf{X}_u^r \\ \left(\frac{\partial T_r}{\partial t}, \psi \right) = - \left((\mathbf{u}_r \cdot \nabla) T_r, \psi \right) - \left((\text{Re Pr})^{-1} \nabla T_r, \psi \right) \quad \forall \psi \in X_T^r \end{cases} . \quad (6.3)$$

As a first step in our investigation, we test the numerical behavior of this POD-G-ROM.

Preliminary Results We avoid the structural complexity of the realistic airflow by testing the model (6.3) on a case with a simple geometry — a cube (see, for example, Figure 6.2 where the mean temperature is shown). However, the physical parameter values are chosen to be close to those used in realistic airflows: $Re = 6618.1$, $Pr = 0.712$, $Gr = 1.46 \times 10^8$, and $T_\infty = 293.65$. Note that the high values of Re and Gr fall in the turbulent range. The DNS utilizes 163609 degrees of freedom.

The nine leading POD modes ($r = 9$) are used in the POD-G-ROM (6.3). Figure 6.3 shows the first nine temperature POD modes. The DNS at time $t = 10$ is shown at the top of Figure 6.4 while the numerical result of the POD-G-ROM (6.3) at the same time instance is shown in the middle of Figure 6.4. Only three iso-surfaces ($T = 293.2, 293.65, 294.1$) are plotted for a clear visualization. Comparing these iso-surfaces at the same time instance clearly shows that the POD-G-ROM generates unphysical structures near the top and bottom of the cube.

To improve the standard POD-G-ROM, we first employed a simple mixing-length model (2.13) as a closure in the moment equation with eddy viscosity coefficient $\nu_{ML} = Re^{-1}$. After applying this model, the accuracy of POD-G-ROM was greatly improved (see the bottom of Figure 6.4). Indeed, the unphysical structures produced by the POD-G-ROM are removed and the simulation results are qualitatively close to those from the DNS. These numerical results suggest that the state-of-the-art POD closure models introduced in this dissertation can be utilized in the Boussinesq equations and can enable efficient numerical airflow simulation models for energy efficient building design and control. We are currently investigating the new POD-ROMs within a more realistic geometry setting (see Figure 6.1 for a snapshot showing temperature contours and the velocity vector field).

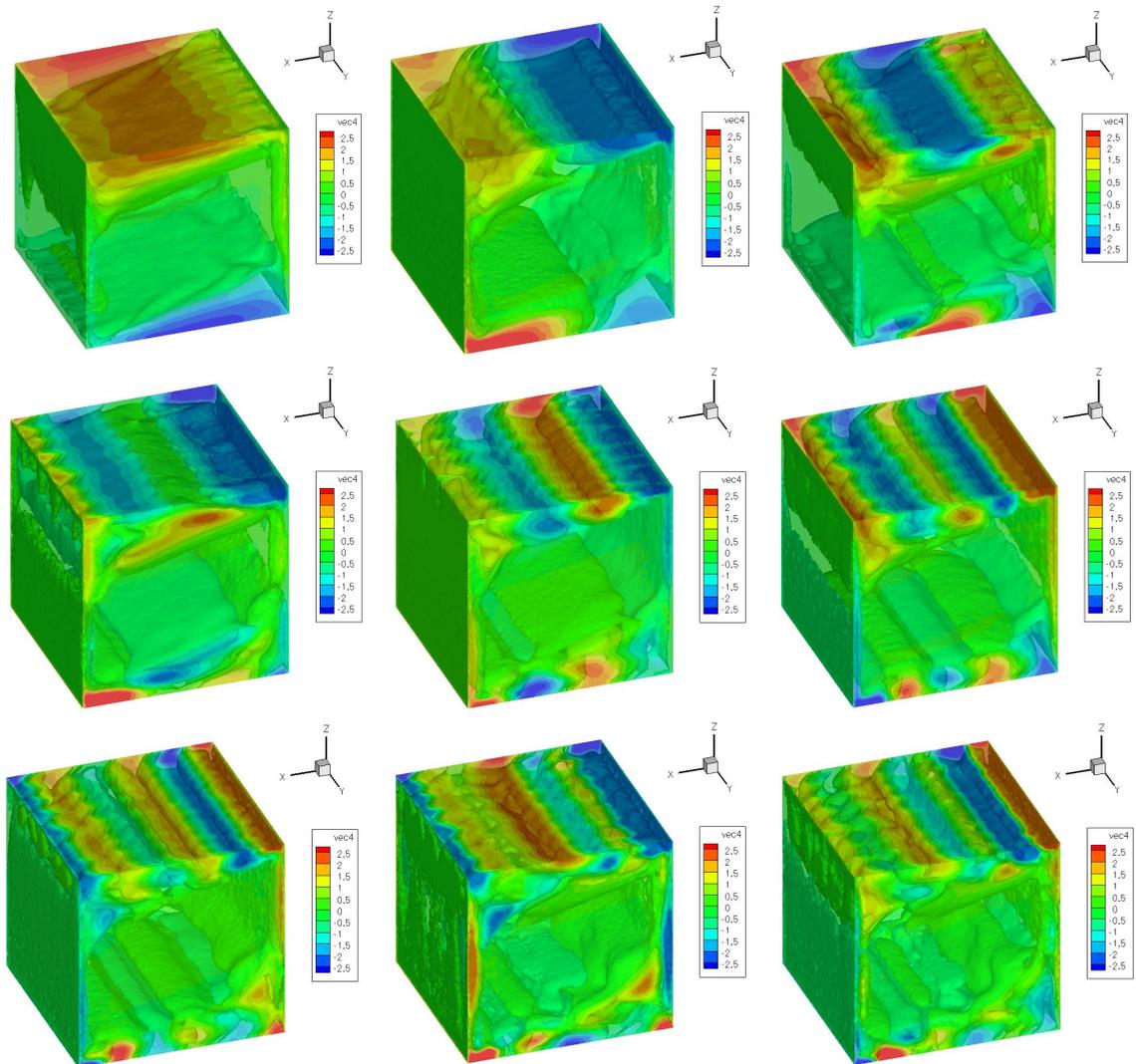
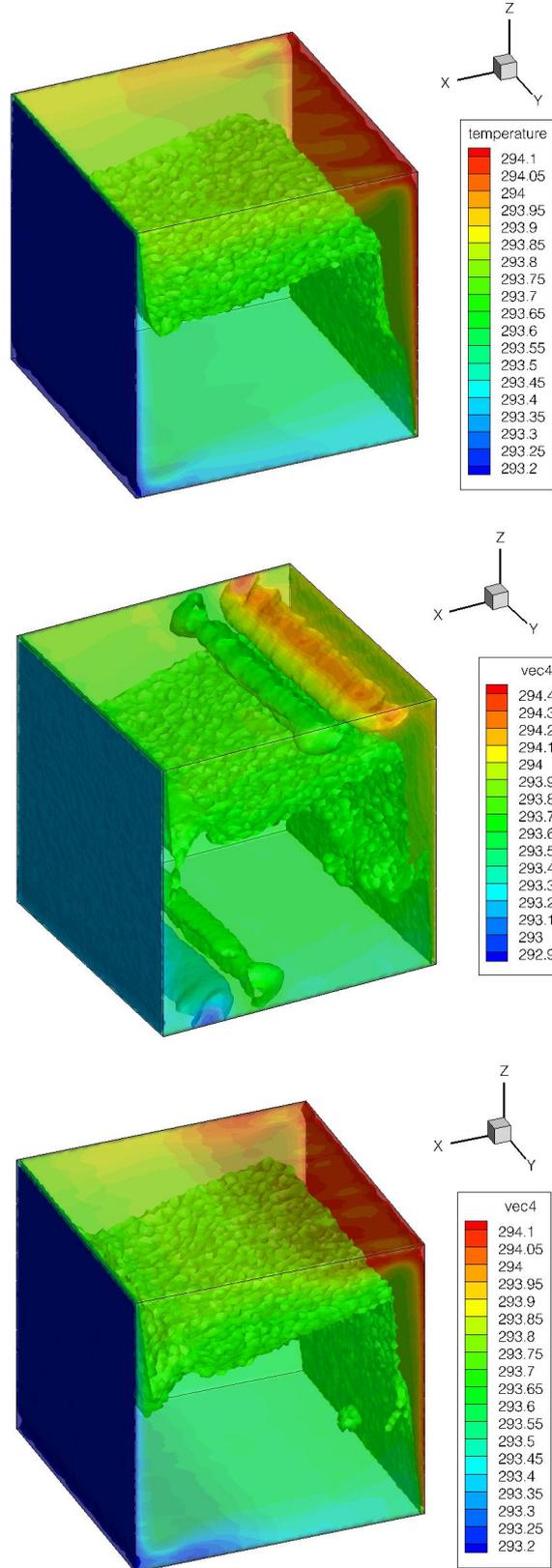
Figure 6.3: The first nine POD modes $\{\psi_1(x), \dots, \psi_9(x)\}$ in the temperature field.

Figure 6.4: Three temperature iso-surfaces (at $T=293.2$, 293.65 and 294.1) for the time instance $t = 10$. Top: DNS; Middle: POD-G-ROM; Bottom: ML-POD-ROM.



6.3 Climate Modeling

Motivation Climate modeling is a *computational grand challenge*, a research topic at the frontier of computational science and scientific computing. Indeed, using standard mathematical models, the currently available computational resources are clearly insufficient for accurate climate modeling. Factoring in the uncertainty in the system makes the problem even more daunting.

This is why simplified models for geophysical flows have been developed: these models aim at capturing the important geophysical structures, while keeping the computational cost at a minimum. Although successful in numerical weather prediction, these models have a prohibitively high computational cost in climate modeling, where different physical processes and *long time integration* need to be considered. Thus,

there is a clear need for computationally efficient mathematical models and their numerical discretizations to enable accurate climate modeling.

The main challenge is the tradeoff between computational efficiency and physical accuracy. Since 2D turbulence is fundamentally different from 3D turbulence, the ROMs developed for engineering flows cannot be directly used in climate modeling. Another challenge is the need to use numerical discretizations that accommodate complex geometries.

Quasi-Geostrophic Equations: As a first step in developing computationally efficient numerical discretizations for the ocean and the atmosphere, we considered the *quasi-geostrophic equations (QGE)* [69, 179, 216], which describe wind-driven circulations characterizing mid-latitude ocean basins (see Figure 6.5). For the QGE, we considered both the streamfunction-vorticity formulation:

$$\begin{cases} \frac{\partial q}{\partial t} + J(q, \psi) = \text{Re}^{-1} \Delta q + F & \text{in } \Omega, \\ q = -\text{Ro} \Delta \psi + y & \text{in } \Omega, \end{cases} \quad (6.4)$$

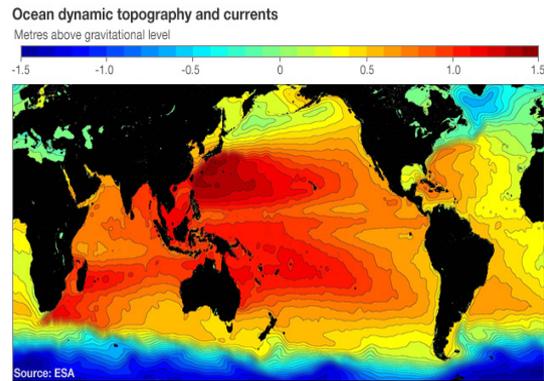


Figure 6.5: Ocean circulation traced by the GOCE gravity mission (source from <http://earth.esa.int/GOCE/>).

and the pure streamfunction formulation:

$$-\frac{\partial}{\partial t}(\Delta\psi) + \text{Re}^{-1}\Delta^2\psi - J(\Delta\psi, \psi) - \text{Ro}^{-1}\frac{\partial\psi}{\partial x} = \text{Ro}^{-1}F, \quad (6.5)$$

with the boundary conditions $\psi = \frac{\partial\psi}{\partial n} = 0$ on $\partial\Omega$. In (6.4) and (6.5), $\text{Ro} = \frac{U}{\beta L^2}$ is the Rossby number, $\text{Re} = \frac{UL}{A}$ is the Reynolds number, ψ is the streamfunction, q is the potential vorticity, $J(q, \psi) = \frac{\partial q}{\partial x}\frac{\partial\psi}{\partial y} - \frac{\partial q}{\partial y}\frac{\partial\psi}{\partial x}$ is the Jacobian, F is the external forcing, U and L represent the velocity and length scale of the geophysical flows in the large scale ocean motions, and βy measures the beta-plane effect from the Coriolis force due to rotation.

For a conforming finite element discretization of the streamfunction-vorticity formulation (6.4), a C^0 element can be utilized. Let $W_\psi = H_0^1(\Omega) \cap W^{1,4}(\Omega)$, $W_q = H^1(\Omega)$. Then the finite element discretization of (6.4) reads: Find $\psi \in W_\psi$ and $q \in W_q$ satisfying

$$\begin{cases} \left(\frac{\partial q}{\partial t}, \varphi \right) + (J(q, \psi), \varphi) = -\text{Re}^{-1}(\nabla q, \nabla \varphi) + (F, \varphi) & \forall \varphi \in W_\psi \\ (q, \omega) = \text{Ro}(\nabla \psi, \nabla \omega) + (y, \omega) & \forall \omega \in W_q \end{cases}. \quad (6.6)$$

This method was analyzed in [225].

However, for a conforming finite element discretization of the streamfunction formulation (6.5), a C^1 element is an appropriate choice. Let $W = H_0^2(\Omega)$. Then the finite element discretization of (6.5) reads: Find $\psi \in W$ satisfying

$$\left(\frac{\partial}{\partial t} \nabla \psi, \nabla \varphi \right) + \text{Re}^{-1}(\Delta \psi, \Delta \varphi) + (J(\psi, \Delta \psi), \varphi) - \text{Ro}^{-1}(\psi_x, \varphi) = \text{Ro}^{-1}(F, \varphi), \forall \varphi \in W. \quad (6.7)$$

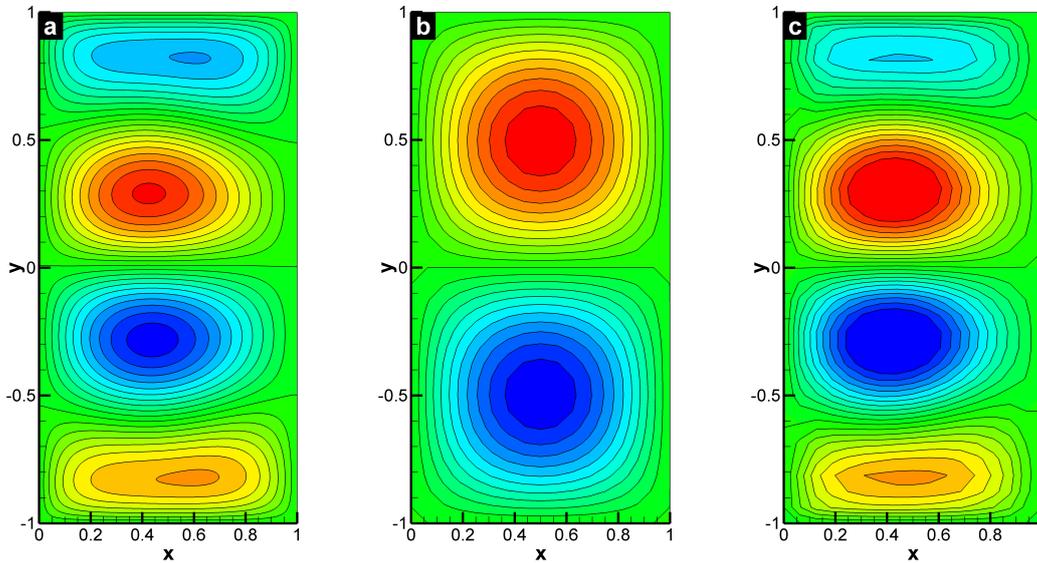
A careful error analysis for the finite element discretization of (6.7) was presented in [83]. Several numerical tests, commonly employed in the geophysical literature, showed the accuracy of the finite element discretization and illustrated the theoretical estimates.

Although a simplified model, due to the strong requisite for long-term integration in climate modeling, efficient numerical algorithms for QGE are needed. We developed a LES *approximate deconvolution (AD)* model that decreased the computational cost of the DNS simulation by three orders of magnitude without compromising the physical accuracy. The QGE-AD model reads

$$\begin{cases} \frac{\partial \bar{q}}{\partial t} + \overline{J(q^*, \psi^*)} = \text{Re}^{-1}\Delta \bar{q} + \bar{F} \\ \bar{q} = -\text{Ro}\Delta \bar{\psi} + y \end{cases}.$$

The goal in AD is to obtain approximations of the unfiltered flow variables by using repeated filtering when approximations of the filtered flow variables are available [32, 145]. These approximations of the unfiltered flow variables are then used in the subfilter-scale terms to close the LES system. Let G be any spatial filter and \mathbb{I} be the identity operator. The inverse filtering operator $G_N^{-1} := \sum_{i=0}^N (\mathbb{I} - G)^i$ is a truncation of $G^{-1} \sim \sum_{i=0}^{\infty} (\mathbb{I} - G)^i$. Then any unfiltered quantity $q = G^{-1}\bar{q}$ can be approximated by $q \approx q^* = G_N^{-1}\bar{q}$. Therefore, $\overline{J(\psi, q)} \approx \overline{J(\psi^*, q^*)}$. The temporal average of the streamfunction in Figure 6.6 clearly shows that the QGE on a coarse mesh leads to nonphysical results, whereas the AD model obtains results that are qualitatively close to those from the DNS. This is the first time that the AD methodology, which has a sound mathematical foundation, has been employed in the numerical simulation of realistic large scale geophysical flows. For details of this approach, see [200].

Figure 6.6: Double-gyre oceanic circulation modeled by the QGE. Time-averaged streamfunction for $\text{Re} = 450$ and $\text{Ro} = 0.0036$: (a) DNS (512×256); (b) under-resolved QGE (32×16); (c) AD model (32×16). The new AD model yields accurate results on a much coarser mesh.



POD-G-ROM for Quasi-Geostrophic Equations An alternative approach for deriving efficient numerical algorithms for the QGE is the POD method. In fact, from the direct observation of the time average of streamfunction in Figure 6.5, the oceanic flows are domi-

nated by large scale structures, which suggests that POD is a perfect fit in the QGE setting.

For clarity, we only consider the streamfunction-vorticity formulation (6.4). A POD-ROM is constructed from the POD basis by writing

$$\begin{cases} \psi(x, t) \approx \psi_r(x, t) = \Psi + \sum_{j=1}^r a_j(t) \xi_j(x) \\ q(x, t) \approx q_r(x, t) = Q + \sum_{j=1}^r b_j(t) \eta_j(x) \end{cases}, \quad (6.8)$$

where $X_\psi^r = \{\xi_1(x), \dots, \xi_r(x)\}$ and $X_q^r = \{\eta_1(x), \dots, \eta_r(x)\}$ are the POD basis functions for streamfunction ψ and potential vorticity q respectively, Ψ and Q are the time average for streamfunction and potential vorticity, respectively, and $\{a_j(t)\}_{j=1}^r$ and $\{b_j(t)\}_{j=1}^r$ are the coefficient functions to be computed. The number of POD modes in X_ψ^r and X_q^r can be different. Again, for simplicity, we choose them to be same.

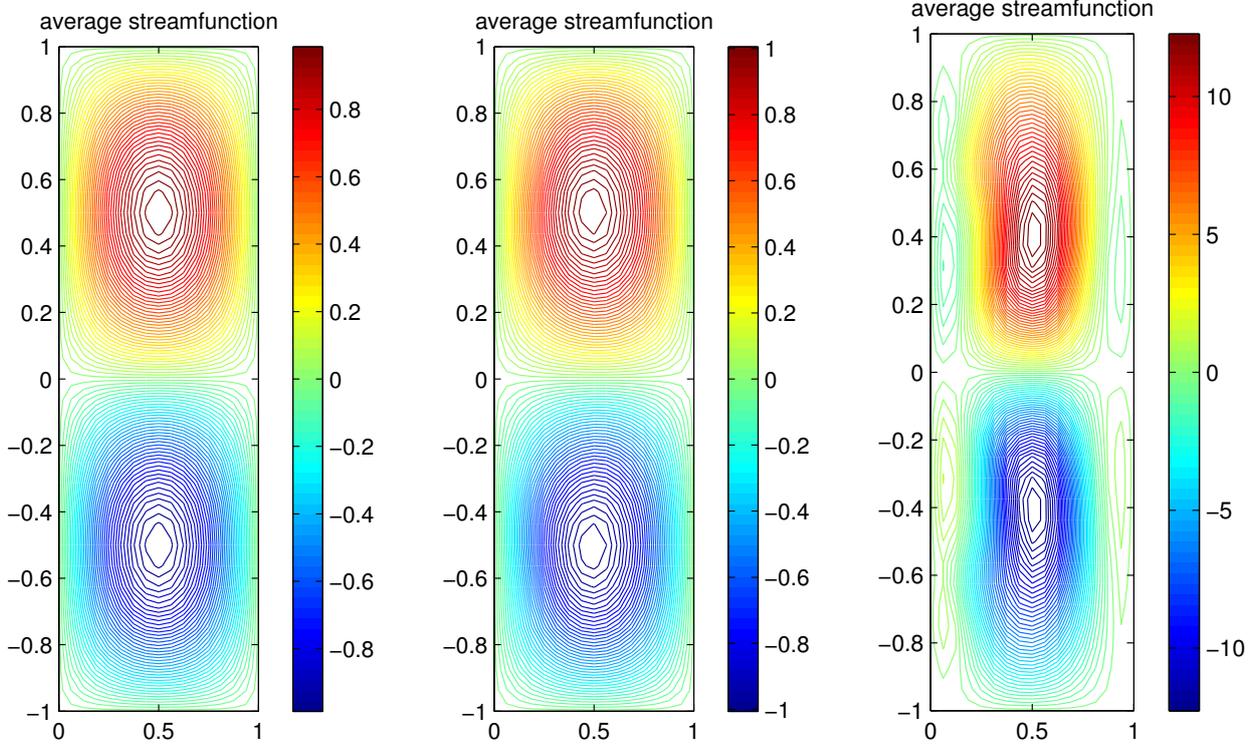
The POD-G-ROM is obtained by replacing the primitive variables in (6.4) with their POD approximations ψ_r and q_r , then projecting (6.4) onto the POD basis sets.

$$\begin{cases} \left(\frac{\partial q_r}{\partial t}, \xi \right) + (J(q_r, \psi_r), \xi) = -\text{Re}^{-1} (\nabla q_r, \nabla \xi) + (F, \xi) & \forall \xi \in X_\psi^r \\ (q_r, \eta) = \text{Ro} (\nabla \psi_r, \nabla \eta) + (y, \eta) & \forall \eta \in X_q^r \end{cases}. \quad (6.9)$$

Preliminary Results As a first step in our investigation, we consider a steady state Taylor-Green vortex flow: $\psi = \sin(\pi x) \sin(\pi y)$, with $\text{Ro} = 0.0016$ and $\text{Re} = 200$. The forcing term F and potential vorticity q are determined by using the analytic formula for ψ in (6.4). The results from POD-G-ROMs with different number of modes $r = 10, 5, 3$ are shown in Figure 6.7. The energy evolution is shown in Figure 6.8. It is seen that $r = 10$ yields accurate results close to the exact solution. However, the accuracy degrades as we decrease the size of POD basis set. In particular, when $r = 3$, the magnitude of the mean streamfunction is 10 times larger than that of the exact solution. Furthermore, the magnitude of the energy is more than 160 times higher than the correct value (see Figure 6.8).

To obtain improved accuracy with a small POD basis set (for example, $r = 3$ in this test case), appropriate closure models should be employed. Due to the fundamentally different physical characteristics of geophysical flows from three dimensional engineering flows, the direct use of LES inspired EV-type POD closure models developed in Chapter 2 is not readily

Figure 6.7: Mean streamfunction from POD-G-ROMs with $r=10$, 5, and 3 (from left to right).



applicable. In fact, the EV-type closure is based on the forward energy cascade. However, in geophysical flows, the main energy transfer is backward, from the small scales to large scales [216]. Therefore, we propose an LES AD-type POD-ROM, based on a mathematical closure modeling strategy that does not rely on the concept of an energy cascade. We first define a POD filter:

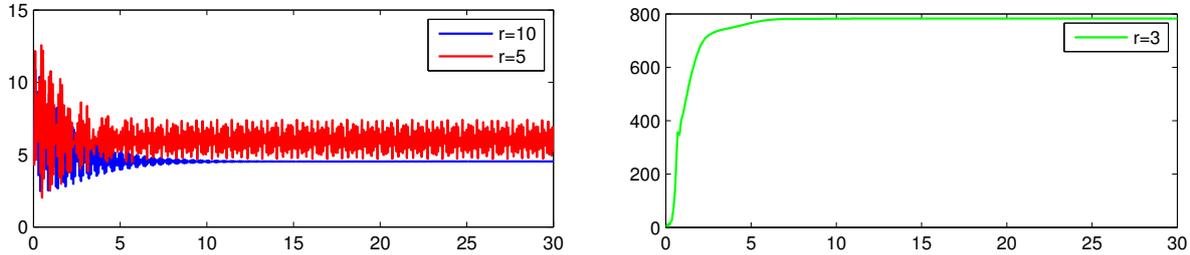
$$(v - (\mathbb{I} - \delta^2 \Delta) \bar{v}, \chi) = 0, \forall \chi \in X^r, \quad (6.10)$$

where \bar{v} is filtered quantity of v , X^r is spanned by a POD basis set of either streamfunction or potential vorticity. The new *POD approximate deconvolution reduced-order model (POD-AD-ROM)* reads

$$\begin{cases} \left(\frac{\partial \bar{q}_r}{\partial t}, \xi \right) + \left(\overline{J(q_r^*, \psi_r^*)}, \xi \right) = \text{Re}^{-1} (\nabla \bar{q}_r, \nabla \xi) + (\bar{F}, \xi) & \forall \xi \in X_\psi^r \\ (\bar{q}_r, \eta) = \text{Ro} (\nabla \bar{\psi}_r, \nabla \eta) + (y, \eta) & \forall \eta \in X_q^r \end{cases}. \quad (6.11)$$

The numerical investigation of this new model for realistic geophysical flows is ongoing.

Figure 6.8: Energy evolution of POD-G-ROMs with $r = 10, 5,$ and 3 (from left to right).



6.4 Future Research Directions

We plan to further investigate several other research avenues.

Snapshots from LES To test the robustness of the new POD-ROMs with respect to parameter changes, we will gradually increase the Reynolds number in the underlying turbulent flows. To obtain the snapshot data for the POD-ROM in these high Reynolds number flows, we plan to employ modern numerical simulation strategies, such as LES. Indeed, with the same computational resources as those used in DNS, with LES we can tackle a more complex system. For example, we can consider Reynolds numbers over a larger interval. Equipped with the closure models proposed in this dissertation, the POD-ROMs could then be tested in the numerical simulation of turbulent flows dominated by coherent structures for a wide range of Reynolds numbers and for time intervals that are longer than those on which the POD snapshot matrix was computed.

Efficiency Improvement of POD-ROMs We plan to develop discretization strategies for reducing the computational cost of the two-level algorithm even further. To this end, we plan to develop more appropriate interpolation strategies, based on DEIM ([54, 55]) rather than simply skipping points.

We also plan to investigate more efficient time-discretization approaches. Higher-order time discretization strategies will provide significant further savings in computational time. Finally, we will take advantage of parallel computing in order to further decrease the computational cost and, at the same time, increase the dimension (and thus physical accuracy) of the POD-ROMs.

Hybrid Approach and A Posteriori Estimate Since the linear closure model (ML-POD-ROM) is computationally efficient, but only works on a relative short time interval if the appropriate EV coefficient α is chosen, we will investigate a hybrid approach: We will use the DS approach to calculate α only when the flow displays a high level of variability, and then use this value in the ML-POD-ROM as long as the flow does not experience sudden transitions. The success of this approach also relies on the design of an accurate a posteriori error estimator. Since the dimension of the POD-ROM is low, the evaluation of a posteriori error bounds should be fast.

Stochastic Calibration Uncertainty quantification of realistic complex flows is important, for example, in nuclear engineering. Since the DNS of turbulent flows is computationally expensive, uncertainty analysis based on a few full model runs or *even one model run* is highly desirable. Therefore, the POD-ROMs are a natural choice for surrogate models. However, since the POD basis is extracted from a data set which corresponds to certain deterministic physical parameters, the POD-ROM becomes inaccurate once the parameters deviate under uncertainty effects. Thus, a new closure strategy for uncertainty analysis has been investigated based on a stochastic calibration approach of the POD-ROM. This is joint work with Dr. Anitescu and Dr. Roderick (Mathematics and Computer Science Division, Argonne National Laboratory).

For climate modeling, when stochastic wind forcing is considered, uncertainties should also be considered in the POD-ROMs. We will use the stochastic calibration method to include the uncertainty effect.

Finite Element Discretization We plan to continue developing and analyzing computationally efficient and numerically accurate finite element discretizations for POD-ROMs and climate modeling. We will derive rigorous error estimates for the Navier-Stokes equations endowed with the new VMS-POD-ROM proposed in [116] and address new challenges, such as the treatment of the strongly nonlinear terms. We will also prove a priori error estimates for the EV type of closure models used in the POD-ROMs of the Boussinesq equations and the AD closure models employed in the POD-AD-ROM (6.11) of the QGE. The extension

of these methods to other simplified models of the ocean and the atmosphere, such as the shallow-water equations and the primitive equations will also be investigated.

Bibliography

- [1] S. Acharjee and N. Zabaras. A proper orthogonal decomposition approach to microstructure model reduction in rodrigues space with applications to optimal control of microstructure-sensitive properties. *Acta materialia*, 51(18):5627–5646, 2003.
- [2] R. A. Adams. *Sobolev spaces*, volume 65 of *Pure and Applied Mathematics*. Academic Press, New York-London, 1975.
- [3] S. Ahuja and C. W. Rowley. Feedback control of unstable steady states of flow past a flat plate using reduced-order estimators. *Journal of Fluid Mechanics*, 645:447–478, 2010.
- [4] I. Akhtar. *Parallel simulations, reduced-order modeling, and feedback control of vortex shedding using fluidic actuators*. PhD thesis, Virginia Tech, 2008.
- [5] I. Akhtar, J. Borggaard, and A. Hay. Shape sensitivity analysis in flow models using a finite-difference approach. *Mathematical Problems in Engineering*, 2010:1–22, 2010.
- [6] I. Akhtar, O. A. Marzouk, and A. H. Nayfeh. A van der Pol-Duffing oscillator model of hydrodynamic forces on canonical structures. *Journal of Computational and Nonlinear Dynamics*, 4:041006, 2009.
- [7] I. Akhtar and A. H. Nayfeh. Model based control of laminar wake using fluidic actuation. *Journal of Computational and Nonlinear Dynamics*, 5:041015, 2010.
- [8] I. Akhtar, A. H. Nayfeh, and C. J. Ribbens. On the stability and extension of reduced-order Galerkin models in incompressible flows. *Theoretical and Computational Fluid Dynamics*, 23(3):213–237, 2009.

-
- [9] A. K. Alekseev and I. M. Navon. The analysis of an ill-posed problem using multi-scale resolution and second-order adjoint techniques. *Computer Methods in Applied Mechanics and Engineering*, 190(15):1937–1953, 2001.
- [10] D. Amsallem, J. Cortial, K. Carlberg, and C. Farhat. A method for interpolating on manifolds structural dynamics reduced-order models. *International Journal for Numerical Methods in Engineering*, 80(9):1241–1258, 2009.
- [11] D. Amsallem, J. Cortial, and C. Farhat. Towards real-time CFD-based aeroelastic computations using a database of reduced-order models. *AIAA Journal*, 48:2029–2037, 2010.
- [12] D. Amsallem and C. Farhat. Interpolation method for adapting reduced-order models and application to aeroelasticity. *AIAA Journal*, 46(7):1803, 2008.
- [13] D. Amsallem and C. Farhat. An online method for interpolating linear parametric reduced-order models. *SIAM Journal on Scientific Computing*, 33(5):2169–2198, 2011.
- [14] D. Amsallem and C. Farhat. Stabilization of projection-based reduced-order models. *International Journal for Numerical Methods in Engineering*, 2012. In press.
- [15] A. C. Antoulas. *Approximation of large-scale dynamical systems*, volume 6. Society for Industrial Mathematics, 2005.
- [16] E. Arian, M. Fahl, and E. W. Sachs. Trust-region proper orthogonal decomposition for flow control. Technical report, DTIC Document, 2000.
- [17] E. Arian, M. Fahl, and E. W. Sachs. Managing POD models by optimization methods. In *Proceedings of 41st IEEE Conference on Decision and Control*, pages 3300–3305, 2002. ThP04-6.
- [18] P. Astrid. *Reduction of process simulation models: a proper orthogonal decomposition approach*. PhD thesis, Technische Universiteit Eindhoven, 2004.

-
- [19] P. Astrid, S. Weiland, K. Willcox, and T. Backx. Missing point estimation in models described by proper orthogonal decomposition. *Automatic Control, IEEE Transactions on*, 53(10):2237–2251, 2008.
- [20] N. Aubry, P. Holmes, J. L. Lumley, , and E. Stone. The dynamics of coherent structures in the wall region of a turbulent boundary layer. *Journal of Fluid Mechanics*, 192:115–173, 1988.
- [21] N. Aubry, W. Y. Lian, and E. S. Titi. Preserving symmetries in the proper orthogonal decomposition. *SIAM Journal on Scientific Computing*, 14:483–505, 1993.
- [22] S. Bagheri, L. Brandt, and D. S. Henningson. Input–output analysis, model reduction and control of the flat-plate boundary layer. *Journal of Fluid Mechanics*, 620:263–298, 2009.
- [23] A. Barbagallo, D. Sipp, and P. J. Schmid. Closed-loop control of an open cavity flow using reduced-order models. *Journal of Fluid Mechanics*, 641:1–50, 2009.
- [24] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera. An ‘empirical interpolation’ method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathematique*, 339(9):667–672, 2004.
- [25] M. Bartak, I. Beausoleil-Morrison, J. A. Clarke, J. Denev, F. Drkal, M. Lain, I. A. Macdonald, A. Melikov, Z. Popiolek, and P. Stankov. Integrating CFD and building simulation. *Building and Environment*, 37(8):865–871, 2002.
- [26] Y. Bazilevs, V. M. Calo, J. A. Cottrell, T. J. R. Hughes, A. Reali, and G. Scovazzi. Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. *Computational Methods in Applied Mechanical Engineering*, 197(1-4):173–201, 2007.
- [27] P. Benner, M. Hinze, and E. J. W. ter Maten. *Model Reduction for Circuit Simulation*, volume 74. Springer, 2011.

-
- [28] M. Bergmann. *Optimisation aérodynamique par réduction de modèle POD et contrôle optimal: application au sillage laminaire d'un cylindre circulaire*. PhD thesis, Institut National Polytechnique de Lorraine, 2004.
- [29] M. Bergmann, C. H. Bruneau, and A. Iollo. Enablers for robust POD models. *Journal of Computational Physics*, 228(2):516–538, 2009.
- [30] M. Bergmann, L. Cordier, and J. P. Brancher. Optimal rotary control of the cylinder wake using proper orthogonal decomposition reduced-order model. *Physics of Fluids*, 17:097101, 2005.
- [31] G. Berkooz, P. Holmes, and J. L. Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. *Annual Review of Fluid Mechanics*, 53:321–575, 1993.
- [32] L. C. Berselli, T. Iliescu, and W. J. Layton. *Mathematics of large eddy simulation of turbulent flows*. Scientific Computation. Springer-Verlag, Berlin, 2006.
- [33] E. Biglieri and K. Yao. Some properties of singular value decomposition and their applications to digital signal processing. *Signal Processing*, 18(3):277–289, 1989.
- [34] P. Binev, A. Cohen, W. Dahmen, R. DeVore, G. Petrova, and P. Wojtaszczyk. Convergence rates for greedy algorithms in reduced basis methods. *Preprint*, 47, 2010.
- [35] J. Borggaard, A. Dugleby, A. Hay, T. Iliescu, and Z. Wang. Reduced-order modeling of turbulent flows. In *Proceedings of MTNS 2008*, 2008.
- [36] J. Borggaard, A. Hay, and D. Pelletier. Interval-based reduced order models for unsteady fluid flow. *International Journal of Numerical Analysis & Modeling*, 2006.
- [37] J. Borggaard, T. Iliescu, and J. P. Roop. A bounded artificial viscosity large eddy simulation model. *SIAM Journal on Numerical Analysis*, 47:622–645, 2009.
- [38] J. Borggaard, T. Iliescu, and Z. Wang. Artificial viscosity proper orthogonal decomposition. *Mathematical and Computer Modelling*, 53(1-2):269–279, 2011.

-
- [39] J. Borggaard, M. Stoyanov, and L. Zietsman. Linear feedback control of a von Kármán street by cylinder rotation. In *American Control Conference (ACC), 2010*, pages 5674–5681. IEEE, 2010.
- [40] M. Bozkurttas, R. Mittal, H. Dong, G. V. Lauder, and P. Madden. Low-dimensional models and performance scaling of a highly deformable fish pectoral fin. *Journal of Fluid Mechanics*, 631:311, 2010.
- [41] S. C. Brenner and L. R. Scott. *The mathematical theory of finite element methods*, volume 15 of *Texts in Applied Mathematics*. Springer-Verlag, New York, 1994.
- [42] A. Buffa, Y. Maday, A. Patera, C. Prudhomme, and G. Turinici. A priori convergence of the greedy algorithm for the parametrized reduced basis. *Preprint*, 2009.
- [43] M. Buffoni, S. Camarri, A. Iollo, and M. V. Salvetti. Low-dimensional modelling of a confined three-dimensional wake flow. *Journal of Fluid Mechanics*, 569:141–150, 2006.
- [44] T. Bui-Thanh, M. Damodaran, and K. Willcox. Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition. *AIAA Journal*, 42(8):1505–1516, 2004.
- [45] J. Burkardt, M. Gunzburger, and H. C. Lee. POD and CVT-based reduced-order modeling of Navier–Stokes flows. *Computer Methods in Applied Mechanics and Engineering*, 196(1-3):337–355, 2006.
- [46] J. Burkardt, M. Gunzburger, and C. Webster. Reduced order modeling of some nonlinear stochastic partial differential equations. *International Journal of Numerical Analysis and Modeling*, 4(3-4):368–391, 2007.
- [47] J. A. Burns and B. B. King. A reduced basis approach to the design of low-order feedback controllers for nonlinear continuous systems. *Journal of Vibration and Control*, 4:297–323, 1998.
- [48] Y. Cao, J. Zhu, Z. Luo, and I. M. Navon. Reduced-order modeling of the upper tropical pacific ocean model using proper orthogonal decomposition. *Computers and Mathematics with Applications*, 52(8-9):1373–1386, 2006.

-
- [49] Y. Cao, J. Zhu, I. M. Navon, and Z. Luo. A reduced-order approach to four-dimensional variational data assimilation using proper orthogonal decomposition. *International Journal for Numerical Methods in Fluids*, 53(10):1571–1583, 2007.
- [50] M. A. Cardoso and L. J. Durlofsky. Linearized reduced-order models for subsurface flow simulation. *Journal of Computational Physics*, 229(3):681–700, 2010.
- [51] K. Carlberg and C. Farhat. A compact proper orthogonal decomposition basis for optimization-oriented reduced-order models. *AIAA Journal*, 5964:10–12, 2008.
- [52] W. Cazemier. *Proper orthogonal decomposition and low dimensional models for turbulent flows*. PhD thesis, University Library Groningen, Groningen, 1997.
- [53] W. Cazemier, R. W. C. P. Verstappen, and A. E. P. Veldman. Proper orthogonal decomposition and low-dimensional models for driven cavity flows. *Physics of Fluids*, 10:1685, 1998.
- [54] S. Chaturantabut and D. C. Sorensen. A state space error estimate for POD-DEIM nonlinear model reduction. *SIAM Journal on Numerical Analysis*, 50:46, 2012.
- [55] S. Chaturantabut, D. C. Sorensen, and J. C. Steven. Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing*, 32(5):2737–2764, 2010.
- [56] X. Chen, S. Akella, and I. M. Navon. A dual-weighted trust-region adaptive POD 4-D Var applied to a finite-volume shallow water equations model on the sphere. *International Journal for Numerical Methods in Fluids*, 68(3):377–402, 2012.
- [57] K. Cohen, S. Siegel, T. McLaughlin, and E. Gillies. Feedback control of a cylinder wake low-dimensional model. *AIAA Journal*, 41(7):1389–1391, 2003.
- [58] L. Cordier, E. Majd, B. Abou, and J. Favier. Calibration of POD reduced-order models using tikhonov regularization. *International Journal for Numerical Methods in Fluids*, 63(2):269–296, 2010.

-
- [59] M. Couplet, C. Basdevant, and P. Sagaut. Calibrated reduced-order POD-Galerkin system for fluid flow modelling. *Journal of Computational Physics*, 207:192–220, 2005.
- [60] M. Couplet, P. Sagaut, and C. Basdevant. Intermodal energy transfers in a proper orthogonal decomposition–Galerkin representation of a turbulent separated flow. *Journal of Fluid Mechanics*, 491:275–284, 2003.
- [61] D. T. Crommelin and A. J. Majda. Strategies for model reduction: comparing different optimal bases. *Journal of the Atmospheric Sciences*, 61:2206–2217, 2004.
- [62] D. N. Daescu and I. M. Navon. Efficiency of a POD-based reduced second-order adjoint model in 4D-Var data assimilation. *International Journal for Numerical Methods in Fluids*, 53(6):985–1004, 2007.
- [63] D. N. Daescu and I. M. Navon. A dual-weighted approach to order reduction in 4DVAR data assimilation. *Monthly Weather Review*, 136(3):1026–1041, 2008.
- [64] G. De Stefano and O. V. Vasilyev. Stochastic coherent adaptive large eddy simulation of forced isotropic turbulence. *Journal of Fluid Mechanics*, 646(1):453–470, 2010.
- [65] A. E. Deane, I. G. Kevrekidis, G. E. Karniadakis, and S. A. Orsag. Low-dimensional models for complex geometry flows: Application to grooved channels and circular cylinder. *Physics of Fluids A*, 3(10):2337–2354, 1991.
- [66] A. E. Deane and C. Mavriplis. Low-dimensional description of the dynamics in separated flow past thick airfoils. *AIAA Journal*, 6:1222–1234, 1994.
- [67] J. Degroote, J. Vierendeels, and K. Willcox. Interpolation among reduced-order matrices to obtain parameterized models for design, optimization and probabilistic analysis. *International Journal for Numerical Methods in Fluids*, 63(2):207–230, 2010.
- [68] J. W. Demmel. *Applied numerical linear algebra*. Society for Industrial and Applied Mathematics Philadelphia, 1997.
- [69] H. A. Dijkstra. *Dynamical Oceanography*. Springer-Verlag, Berlin, 2008.

-
- [70] A. Doostan, R. G. Ghanem, and J. Red-Horse. Stochastic model reduction for chaos representations. *Computational Methods in Applied Mechanical Engineering*, 196(37-40):3951–3966, 2007.
- [71] Q. Du and M. Gunzburger. Finite-element approximations of a Ladyzhenskaya model for stationary incompressible viscous flow. *SIAM Journal on Numerical Analysis*, 27:1–19, 1990.
- [72] Q. Du and M. Gunzburger. Analysis of a Ladyzhenskaya model for incompressible viscous flow. *Journal of Mathematical Analysis and Applications*, 155:21–45, 1991.
- [73] E. Efficiency. Buildings energy data book. US Department of Energy, 2008.
- [74] S.J. Emmerich and K.B. McGrattan. Application of a large eddy simulation model to study room airflow. *Transactions Ransactions-American Society of Heating Refrigerating and Air conditioning Engineers*, 104:1128–1140, 1998.
- [75] M. Fahl. *Trust-region methods for flow control based on reduced order modelling*. PhD thesis, Universitätsbibliothek, 2001.
- [76] F. Fang, C. C. Pain, I. M. Navon, G. J. Gorman, M. D. Piggott, P. A. Allison, P. E. Farrell, and A. J. H. Goddard. A POD reduced order unstructured mesh ocean modelling method for moderate reynolds number flows. *Ocean Modelling*, 28(1-3):127–136, 2009.
- [77] F. Fang, C. C. Pain, I. M. Navon, M. D. Piggott, G. J. Gorman, P. E. Farrell, P. A. Allison, and A. J. H. Goddard. A POD reduced-order 4D-Var adaptive mesh ocean modelling approach. *International Journal for Numerical Methods in Fluids*, 60(7):709–732, 2009.
- [78] M. Farge and K. Schneider. Coherent vortex simulation (CVS), a semi-deterministic turbulence model using wavelets. *Flow, Turbulence and Combustion*, 66(4):393–426, 2001.

-
- [79] M. Farge, K. Schneider, and N. Kevlahan. Non-Gaussianity and coherent vortex simulation for two-dimensional turbulence using an adaptive orthogonal wavelet basis. *Physics of Fluids*, 11:2187, 1999.
- [80] B. F. Feeny. On the proper orthogonal modes and normal modes of continuous vibration systems. *Journal of Vibration and Acoustics*, 124:157, 2002.
- [81] B. F. Feeny. A complex orthogonal decomposition for wave motion analysis. *Journal of Sound and Vibration*, 310(1-2):77–90, 2008.
- [82] C. Foias, M. S. Jolly, I. G. Kevrekidis, and E. S. Titi. Dissipativity of the numerical schemes. *Nonlinearity*, 4:591–613, 1991.
- [83] E. Foster, T. Iliescu, and Z. Wang. A finite element discretization of the streamfunction formulation of the stationary quasi-geostrophic equations of the ocean. Submitted, 2011.
- [84] P. Galán del Sastre and R. Bermejo. Error estimates of proper orthogonal decomposition eigenvectors and Galerkin projection for a general dynamical system arising in fluid models. *Numerische Mathematik*, 110(1):49–81, 2008.
- [85] D. Galbally, K. Fidkowski, K. Willcox, and O. Ghattas. Non-linear model reduction for uncertainty quantification in large-scale inverse problems. *International Journal for Numerical Methods in Engineering*, 81(12):1581–1608, 2010.
- [86] B. Galletti, A. Bottaro, C. H. Bruneau, and A. Iollo. Accurate model reduction of transient and forced wakes. *European Journal of Mechanics-B/Fluids*, 26(3):354–366, 2007.
- [87] B. Galletti, C. H. Bruneau, L. Zannetti, and A. Iollo. Low-order modeling of laminar flow regimes past a confined square cylinder. *Journal of Fluid Mechanics*, 503:161–170, 2004.
- [88] M. Germano, U. Piomelli, P. Moin, and W. H. Cabot. A dynamic subgrid-scale eddy viscosity model. *Physics of Fluids A*, 3:1760–1765, 1991.

-
- [89] D. E. Goldstein and O. V. Vasilyev. Stochastic coherent adaptive large eddy simulation method. *Physics of Fluids*, 16:2497, 2004.
- [90] P. B. Gonçalves, F. M. A. Silva, and Z. Del Prado. Low-dimensional models for the nonlinear vibration analysis of cylindrical shells based on a perturbation procedure and proper orthogonal decomposition. *Journal of Sound and Vibration*, 315(3):641–663, 2008.
- [91] W. R. Graham, J. Peraire, and K. Y. Tang. Optimal control of vortex shedding using low-order models. part II – model-based control. *International Journal for Numerical Methods in Engineering*, 44:973–990, 1999.
- [92] V. Gravemeier. A consistent dynamic localization model for large eddy simulation of turbulent flows based on a variational formulation. *Journal of Computational Physics*, 218(2):677–701, 2006.
- [93] V. Gravemeier. Scale-separating operators for variational multiscale large eddy simulation of turbulent flows. *Journal of Computational Physics*, 212(2):400–435, 2006.
- [94] V. Gravemeier, W. A. Wall, and E. Ramm. A three-level finite element method for the instationary incompressible Navier-Stokes equations. *Computational Methods in Applied Mechanical Engineering*, 193(15-16):1323–1366, 2004.
- [95] V. Gravemeier, W. A. Wall, and E. Ramm. Large eddy simulation of turbulent incompressible flows by a three-level finite element method. *International Journal for Numerical Methods in Fluids*, 48(10):1067–1099, 2005.
- [96] M. A. Grepl, Y. Maday, N. C. Nguyen, and A. T. Patera. Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 41(3):575–605, 2007.
- [97] M. A. Grepl and A. T. Patera. *A Posteriori* error bounds for reduced-basis approximations of parametrized parabolic partial differential equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 39(1):157–181, 2005.

-
- [98] J.-L. Guermond. Stabilization of Galerkin approximations of transport equations by subgrid modeling. *M2AN. Mathematical Modelling and Numerical Analysis*, 33(6):1293–1316, 1999.
- [99] J.-L. Guermond. Stabilisation par viscosité de sous-maille pour l’approximation de Galerkin des opérateurs linéaires monotones. *Comptes Rendus Mathématique*, 328:617–622, 1999.
- [100] H. Gunes, S. Sirisup, and G. E. Karniadakis. Gappy data: To Krig or not to Krig? *Journal of Computational Physics*, 212(1):358–382, 2006.
- [101] B. Haasdonk and M. Ohlberger. Reduced basis method for finite volume approximations of parametrized linear evolution equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 42(02):277–302, 2008.
- [102] P. C. Hansen. *Rank-deficient and discrete ill-posed problems: numerical aspects of linear inversion*. Society for Industrial Mathematics, 1998.
- [103] K. Hasselmann. PIPs and POPs: The reduction of complex dynamical systems using principal interaction and oscillation patterns. *Journal of Geophysical Research*, 93(11):015–11, 1988.
- [104] A. Hay, J. Borggaard, I. Akhtar, and D. Pelletier. Reduced-order models for parameter dependent geometries based on shape sensitivity analysis. *Journal of Computational Physics*, 229(4):1327–1352, 2010.
- [105] A. Hay, J. Borggaard, and D. Pelletier. Local improvements to reduced-order models using sensitivity analysis of the proper orthogonal decomposition. *Journal of Fluid Mechanics*, 629:41–72, 2010.
- [106] N. Heitmann. Subgridscale stabilization of time-dependent convection dominated diffusive transport. *Journal of Mathematical Analysis and Applications*, 331(1):38–50, 2007.

-
- [107] J. Hoepffner, E. Akervik, U. Ehrenstein, and D. S. Henningson. Control of cavity-driven separated boundary layer. In *Proceedings of the Conference on Active Flow Control*, Berlin, September 2006.
- [108] P. Holmes, J. L. Lumley, and G. Berkooz. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*. Cambridge University Press, Cambridge, UK, 1996.
- [109] C. Homescu, L. R. Petzold, and R. Serban. Error estimation for reduced-order models of dynamical systems. *SIAM Journal on Numerical Analysis*, 43(4):1693–1714 (electronic), 2005.
- [110] C. Homescu, L. R. Petzold, and R. Serban. Error estimation for reduced-order models of dynamical systems. *SIAM Review*, 49(2):277–299, 2007.
- [111] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.
- [112] T. J. R. Hughes. Multiscale phenomena: Green’s functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods. *Computational Methods in Applied Mechanical Engineering*, 127(1-4):387–401, 1995.
- [113] T. J. R. Hughes, L. Mazzei, and K. E. Jansen. Large eddy simulation and the variational multiscale method. *Computing and Visualization in Science*, 3:47–59, 2000.
- [114] T. J. R. Hughes, L. Mazzei, A. Oberai, and A. Wray. The multiscale formulation of large eddy simulation: Decay of homogeneous isotropic turbulence. *Physics of Fluids*, 13(2):505–512, 2001.
- [115] T. J. R. Hughes, A. Oberai, and L. Mazzei. Large eddy simulation of turbulent channel flows by the variational multiscale method. *Physics of Fluids*, 13(6):1784–1799, 2001.
- [116] T. Iliescu and Z. Wang. Variational multiscale proper orthogonal decomposition: Convection-dominated convection-diffusion-reaction equations. *Mathematics of Computation*, 2011. In press.

-
- [117] A. Iollo, A. Dervieux, J. A. Désidéri, and S. Lanteri. Two stable POD-based approximations to the Navier–Stokes equations. *Computing and Visualization in Science*, 3(1):61–66, 2000.
- [118] A. Iollo, S. Lanteri, and J. A. Désidéri. Stability Properties of POD-Galerkin Approximations for the Compressible Navier–Stokes Equations. *Theoretical and Computational Fluid Dynamics*, 13(6):377–396, 2000.
- [119] K. Ito and S. S. Ravindran. A reduced-order method for simulation and control of fluid flows. *Journal of Computational Physics*, 143(2):403–425, 1998.
- [120] K. Ito and J. D. Schroeter. Reduced order feedback synthesis for viscous incompressible flows. *Mathematical and computer modelling*, 33(1):173–192, 2001.
- [121] V. John and S. Kaya. A finite element variational multiscale method for the Navier–Stokes equations. *SIAM Journal on Scientific Computing*, 26:1485, 2005.
- [122] V. John and S. Kaya. Finite element error analysis of a variational multiscale method for the Navier-Stokes equations. *Advances in Computational Mathematics*, 28(1):43–61, 2008.
- [123] V. John, S. Kaya, and A. Kindl. Finite element error analysis for a projection-based variational multiscale method with nonlinear eddy viscosity. *Journal of Mathematical Analysis and Applications*, 344(2):627–641, 2008.
- [124] V. John, S. Kaya, and W. Layton. A two-level variational multiscale method for convection-dominated convection-diffusion equations. *Computational Methods in Applied Mechanical Engineering*, 195(33-36):4594–4603, 2006.
- [125] V. John and A. Kindl. Variants of projection-based finite element variational multiscale methods for the simulation of turbulent flows. *International Journal for Numerical Methods in Fluids*, 56(8):1321–1328, 2008.
- [126] V. John and A. Kindl. A variational multiscale method for turbulent flow simulation with adaptive large scale space. *Journal of Computational Physics*, 229:301–312, 2010.

-
- [127] V. John and A. Kindl. Numerical studies of finite element variational multiscale methods for turbulent flow simulations. *Computational Methods in Applied Mechanical Engineering*, 199(13-16):841–852, 2010.
- [128] B. H. Jørgensen, J. N. Sørensen, and M. Brøns. Low-dimensional modeling of a driven cavity flow with two free parameters. *Theoretical and computational fluid dynamics*, 16(4):299–317, 2003.
- [129] V. L. Kalb and A. E. Deane. An intrinsic stabilization scheme for proper orthogonal decomposition based low-dimensional models. *Physics of Fluids*, 19:054106, 2007.
- [130] K. Karhunen. Zur spektraltheorie stochastischer prozesse. *Annales Academiae Scientiarum Fennicae*, 37, 1946.
- [131] S. Kaya. *Numerical analysis of a variational multiscale method for turbulence*. PhD thesis, University of Pittsburgh, 2004.
- [132] G. Kerschen and J. C. Golinval. A model updating strategy of non-linear vibrating structures. *International Journal for Numerical Methods in Engineering*, 60(13):2147–2164, 2004.
- [133] G. Kerschen, V. Lenaerts, and J. C. Golinval. Identification of a continuous structure with a geometrical non-linearity. Part I: Conditioned reverse path method. *Journal of Sound and Vibration*, 262(4):889–906, 2003.
- [134] V. C. Klema and A. Laub. The singular value decomposition: Its computation and some applications. *Automatic Control, IEEE Transactions on*, 25(2):164–176, 1980.
- [135] E. Kreyszig. *Introductory functional analysis with applications*. Wiley Classics Library. John Wiley & Sons Inc., New York, 1989.
- [136] K. Kunisch and S. Volkwein. Control of the Burgers equation by a reduced-order approach using proper orthogonal decomposition. *Journal of Optimization Theory and Applications*, 102(2):345–371, 1999.

-
- [137] K. Kunisch and S. Volkwein. Galerkin proper orthogonal decomposition methods for parabolic problems. *Numerische Mathematik*, 90(1):117–148, 2001.
- [138] K. Kunisch and S. Volkwein. Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. *SIAM Journal on Numerical Analysis*, 40(2):492–515 (electronic), 2002.
- [139] K. Kunisch, S. Volkwein, and L. Xie. HJB-POD-based feedback design for the optimal control of evolution problems. *SIAM Journal on Applied Dynamical Systems*, 3:701, 2004.
- [140] F. Kwasniok. The reduction of complex dynamical systems using principal interaction patterns. *Physica D: Nonlinear Phenomena*, 92(1-2):28–60, 1996.
- [141] S. Lall, J. E. Marsden, and S. Glavaški. A subspace approach to balanced truncation for model reduction of nonlinear control systems. *International Journal of Robust and Nonlinear Control*, 12(6):519–535, 2002.
- [142] F. Lanata and A. D. Grosso. Damage detection and localization for continuous static monitoring of structures using a proper orthogonal decomposition of signals. *Smart materials and structures*, 15:1811, 2006.
- [143] J. Lawrie and J. W. Hearne. Using modified proper orthogonal decomposition (MPOD) for reducing ecosystem models. *Anziam Journal*, 48(4):461, 2007.
- [144] W. J. Layton. A connection between subgrid scale eddy viscosity and mixed methods. *Applied Mathematics and Computation*, 133:147–157, 2002.
- [145] W. J. Layton and L. Rebholz. *Approximate Deconvolution Models of Turbulence: Analysis, Phenomenology and Numerical Analysis*. Springer Verlag, 2012.
- [146] O. Lehmann, M. Luchtenburg, B. R. Noack, R. King, M. Morzyński, and G. Tadmor. Wake stabilization using POD Galerkin models with interpolated modes. In *Proc. 44th IEEE Conference on Decision and Control*, 2005.

-
- [147] Y. C. Liang, H. P. Lee, S. P. Lim, W. Z. Lin, K. H. Lee, and C. G. Wu. Proper orthogonal decomposition and its applications—Part I: Theory. *Journal of Sound and Vibration*, 252(3):527–544, 2002.
- [148] T. Lieu and C. Farhat. Adaptation of aeroelastic reduced-order models and application to an F-16 configuration. *AIAA Journal*, 45(6):1244–1257, 2007.
- [149] T. Lieu, C. Farhat, and M. Lesoinne. Reduced-order fluid/structure modeling of a complete aircraft configuration. *Computational Methods in Applied Mechanical Engineering*, 195(41):5730–5742, 2006.
- [150] J.-L. Lions. *Quelque méthodes de résolution des problèmes aux limites non linéaires*. Dunod, Gauthier–Villars, Paris, 1969.
- [151] M. M. Loève. *Probability Theory*. Van Nostrand, Princeton, NJ, 1955.
- [152] E. N. Lorenz and Air Force Cambridge Research Center (US) Geophysics Research Directorate. Empirical orthogonal functions and statistical weather prediction. 1956.
- [153] J. L. Lumley. The structure of inhomogeneous turbulent flows. In A.M. Yaglom, editor, *Atmospheric Turbulence and Radio Wave Propagation*, pages 166–178, 1967.
- [154] J. L. Lumley and B. Podvin. Dynamical systems theory and extra rates of strain in turbulent flows. *Experimental Thermal and Fluid Science*, 13(3):180–189, 1996.
- [155] Z. Luo, J. Chen, I. M. Navon, and X. Yang. Mixed finite element formulation and error estimates based on proper orthogonal decomposition for the nonstationary Navier-Stokes equations. *SIAM Journal on Numerical Analysis*, 47(1):1–19, 2008.
- [156] Z. Luo, J. Chen, J. Zhu, R. Wang, and I. M. Navon. An optimizing reduced order FDS for the tropical pacific ocean reduced gravity model. *International Journal for Numerical Methods in Fluids*, 55(2):143–161, 2007.
- [157] Z. Luo, J. Zhu, R. Wang, and I. M. Navon. Proper orthogonal decomposition approach and error estimation of mixed finite element methods for the tropical Pacific Ocean

- reduced gravity model. *Computational Methods in Applied Mechanical Engineering*, 196(41-44):4184–4195, 2007.
- [158] Z. D. Luo, J. Chen, P. Sun, and X. Z. Yang. Finite element formulation based on proper orthogonal decomposition for parabolic equations. *Science in China Series A*, 52(3):585–596, 2009.
- [159] H. V. Ly, K. D. Mease, and E. S. Titi. Distributed and boundary control of the viscous Burgers’ equation. *Numerical Functional Analysis and Optimization*, 18(1-2):143–188, 1997.
- [160] X. Ma, G. S. Karamanos, and G. E. Karniadakis. Dynamics and low-dimensionality of a turbulent near wake. *Journal of Fluid Mechanics*, 410:29–65, 2000.
- [161] X. Ma and G. E. Karniadakis. A low-dimensional model for simulating three-dimensional cylinder flow. *Journal of Fluid Mechanics*, 458:181–190, 2002.
- [162] Y. Maday, A. T. Patera, and G. Turinici. Global a priori convergence theory for reduced-basis approximations of single-parameter symmetric coercive elliptic partial differential equations. *Comptes Rendus Mathematique*, 335(3):289–294, 2002.
- [163] C. Meneveau, T. Lund, and W. Cabot. A Lagrangian dynamic subgrid-scale model of turbulence. *Journal of Fluid Mechanics*, 319:353–385, 1996.
- [164] G. J. Minty. Monotone (nonlinear) operators in Hilbert space. *Duke Mathematical Journal*, 29:341–346, 1962.
- [165] K. Mohseni, H. Zhao, and J. E. Marsden. Shock regularization for Burgers equation. In *44th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, January 2006. AIAA paper 2006-1516.
- [166] Y. Morinishi and O. V. Vasilyev. Vector level identity for dynamic subgrid scale modeling in large eddy simulation. *Physics of Fluids*, 14:3616, 2002.
- [167] M. Morzyński, W. Stankiewicz, B. R. Noack, R. King, F. Thiele, and G. Tadmor. Continuous mode interpolation for control-oriented models of fluid flow. In *Active*

- flow control: papers contributed to the Conference" Active Flow Control 2006", Berlin, Germany, September 27 to 29, 2006*, volume 95, pages 260–278. Springer Verlag, 2007.
- [168] N. E. Murray and L. S. Ukeiley. An application of Gappy POD. *Experiments in Fluids*, 42(1):79–91, 2007.
- [169] N. C. Nguyen, A. T. Patera, and J. Peraire. A best points interpolation method for efficient approximation of parametrized functions. *International Journal for Numerical Methods in Engineering*, 73(4):521–543, 2008.
- [170] N. C. Nguyen and J. Peraire. An efficient reduced-order modeling approach for non-linear parametrized partial differential equations. *International Journal for Numerical Methods in Engineering*, 76(1):27–55, 2008.
- [171] N. C. Nguyen, G. Rozza, D. B. P. Huynh, and A. T. Patera. Reduced basis approximation and a posteriori error estimation for parametrized parabolic pdes: Application to real-time bayesian parameter estimation. *Large-Scale Inverse Problems and Quantification of Uncertainty*, pages 151–177, 2010.
- [172] B. R. Noack, K. Afanasiev, M. Morzynski, and F. Thiele. A hierarchy of low-dimensional models for the transient and post-transient cylinder wake. *Journal of Fluid Mechanics*, 497:335–363, 2003.
- [173] B. R. Noack, M. Morzynski, and G. Tadmor. *Reduced-Order Modelling for Flow Control*, volume 528. Springer Verlag, 2011.
- [174] B. R. Noack, P. Papas, and P. A. Monkewitz. Low-dimensional Galerkin model of a laminar shear-layer. Technical Report 2002-01, École Polytechnique Fédérale de Lausanne, 2002.
- [175] B. R. Noack, P. Papas, and P. A. Monkewitz. The need for a pressure-term representation in empirical Galerkin models of incompressible shear flows. *Journal of Fluid Mechanics*, 523:339–365, 2005.

-
- [176] B. R. Noack, M. Schlegel, B. Ahlborn, G. Mutschke, M. Morzynski, P. Comte, and G. Tadmor. A finite-time thermodynamics of unsteady fluid flows. *Journal of Non-equilibrium thermodynamics*, 33(2):103–148, 2008.
- [177] J. Nocedal and S. J. Wright. *Numerical optimization*. Springer verlag, 1999.
- [178] A. T. Patera and G. Rozza. *Reduced Basis Approximation and A Posteriori Error Estimation for Parameterized Partial Differential Equations*. Version 1.0, Copyright MIT 2006, to appear in (tentative rubric) MIT Pappalardo Graduate Monographs in Mechanical Engineering.
- [179] J. Pedlosky. *Geophysical fluid dynamics*. Springer-Verlag, second edition, 1992.
- [180] B. Podvin. On the adequacy of the ten-dimensional model for the wall layer. *Physics of Fluids*, 13(1):210–224, 2001.
- [181] B. Podvin. A proper-orthogonal-decomposition–based model for the wall layer of a turbulent channel flow. *Physics of Fluids*, 21:015111, 2009.
- [182] B. Podvin and J. L. Lumley. A low-dimensional approach for the minimal flow unit. *Journal of Fluid Mechanics*, 362:121–155, 1998.
- [183] F. Porté-Agel, C. Meneveau, and M. B. Parlange. A scale-dependent dynamic model for large-eddy simulation: application to a neutral atmospheric boundary layer. *Journal of Fluid Mechanics*, 415:261–284, 2000.
- [184] C. Prudhomme, D. V. Rovas, K. Veroy, L. Machiels, Y. Maday, A. T. Patera, and G. Turinici. Reliable real-time solution of parametrized partial differential equations: Reduced-basis output bound methods. *Journal of Fluids Engineering*, 124:70, 2002.
- [185] A. Quarteroni and A. Valli. *Numerical approximation of partial differential equations*. Springer Series in Computational Mathematics, 1994.
- [186] M. Rathinam and L. R. Petzold. A new look at proper orthogonal decomposition. *SIAM Journal on Numerical Analysis*, 41(5):1893–1925, 2004.

-
- [187] S. S. Ravindran. Reduced-order adaptive controllers for fluid flows using POD. *Journal of Scientific Computing*, 15(4):457–478, 2000.
- [188] S. S. Ravindran. A reduced-order approach for optimal control of fluids using proper orthogonal decomposition. *International Journal for Numerical Methods in Fluids*, 34(5):425–448, 2000.
- [189] S. S. Ravindran. Adaptive reduced-order controllers for a thermal flow system using proper orthogonal decomposition. *SIAM Journal on Scientific Computing*, 23:1924, 2002.
- [190] S. S. Ravindran. Error estimates for reduced order POD models of Navier-Stokes equations. In *2008 International Mechanical Engineering Congress and Exposition*, volume 12, pages 651–657. ASME, 2008.
- [191] S. S. Ravindran. Error analysis for Galerkin POD approximation of the nonstationary boussinesq equations. *Numerical Methods for Partial Differential Equations*, 27(6):1639–1665, 2011.
- [192] D. Rempfer. Investigations of boundary layer transition via Galerkin projections on empirical eigenfunctions. *Physics of Fluids*, 8:175, 1996.
- [193] D. Rempfer. On low-dimensional Galerkin models for fluid flow. *Theoretical and Computational Fluid Dynamics*, 14(2):75–88, 2000.
- [194] D. Rempfer and H. F. Fasel. Dynamics of three-dimensional coherent structures in a flat-plate boundary layer. *Journal of Fluid Mechanics*, 275:257–283, 1994.
- [195] M. Rewieński and J. White. A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 22(2):155–170, 2003.
- [196] M. Rewieński and J. White. Model order reduction for nonlinear dynamical systems based on trajectory piecewise-linear approximations. *Linear Algebra and its Applications*, 415(2):426–454, 2006.

-
- [197] J. D. Rodriguez and L. Sirovich. Low-dimensional dynamics for the complex Ginzburg-Landau equation. *Physica D: Nonlinear Phenomena*, 43(1):77–86, 1990.
- [198] G. Rozza, D. B. P. Huynh, and A. T. Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Archives of Computational Methods in Engineering*, 15(3):1–47, 2007.
- [199] P. Sagaut. *Large eddy simulation for incompressible flows*. Scientific Computation. Springer-Verlag, Berlin, third edition, 2006.
- [200] O. San, A. E. Staples, Z. Wang, and T. Iliescu. Approximate deconvolution large eddy simulation of a barotropic ocean circulation model. *Ocean Modelling*, 40:120–132, 2011.
- [201] A. Sawant and A. Acharya. Model reduction via parametrized locally invariant manifolds: Some examples. *Computational Methods in Applied Mechanical Engineering*, 195(44):6287–6311, 2006.
- [202] W. H. A. Schilders, H. A. Van Der Vorst, and J. Rommes. *Model order reduction: theory, research aspects and applications*, volume 13. Springer Verlag, 2008.
- [203] M. Schu and E. W. Sachs. Reduced order models in PIDE constrained optimization. *Control and Cybernetics*, 39(3):661–675, 2010.
- [204] R. Serban, C. Homescu, and L. R. Petzold. The effect of problem perturbations on nonlinear dynamical systems and their reduced-order models. *SIAM Journal on Scientific Computing*, 29(6):2621–2643, 2007.
- [205] S. Sirisup and G. E. Karniadakis. A spectral viscosity method for correcting the long-term behavior of POD models. *Journal of Computational Physics*, 194(1):92–116, 2004.
- [206] L. Sirovich. Turbulence and the dynamics of coherent structures. I. Coherent structures. *Quarterly of Applied Mathematics*, 45(3):561–571, 1987.
- [207] L. Sirovich. Turbulence and the dynamics of coherent structures. II. Symmetries and transformations. *Quarterly of Applied Mathematics*, 45(3):573–582, 1987.

-
- [208] L. Sirovich. Turbulence and the dynamics of coherent structures. III. Dynamics and scaling. *Quarterly of Applied Mathematics*, 45(3):583–590, 1987.
- [209] L. Sirovich and M. Kirby. Low-dimensional procedure for the characterization of human faces. *Journal of Optical Society of America A*, 4(3):529–524, 1987.
- [210] L. Sirovich and J. D. Rodriguez. Coherent structures and chaos: a model problem. *Physics Letters A*, 120(5):211–214, 1987.
- [211] J. S. Smagorinsky. General circulation experiments with the primitive equations. *Monthly Weather Review*, 91:99–164, 1963.
- [212] E. Tadmor. Convergence of spectral methods for nonlinear conservation laws. *SIAM Journal on Numerical Analysis*, 26(1):30, 1989.
- [213] G. Tadmor and B. R. Noack. Dynamic estimation for reduced Galerkin models of fluid flows. In *Proceedings of the American Control Conference*, volume 1, pages 746–751. IEEE, 2004.
- [214] V. Thomée. *Galerkin finite element methods for parabolic problems*. Springer Verlag, 2006.
- [215] 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.
- [216] G. K. Vallis. *Atmospheric and oceanic fluid dynamics: fundamentals and large-scale circulation*. Cambridge University Press, 2006.
- [217] O. V. Vasilyev and D. E. Goldstein. Local spectrum of commutation error in large eddy simulation. *Physics of Fluids*, 16(2):470–473, 2004.
- [218] O. V. Vasilyev, T. S. Lund, and P. Moin. A general class of commutative filters for LES in complex geometries. *Journal of Computational Physics*, 146(1):82–104, 1998.

-
- [219] D. Venturi and G. E. Karniadakis. Gappy data and reconstruction procedures for flow past a cylinder. *Journal of Fluid Mechanics*, 519(1):315–336, 2004.
- [220] D. Venturi, X. Wan, and G. E. Karniadakis. Stochastic low-dimensional modelling of a random laminar wake past a circular cylinder. *Journal of Fluid Mechanics*, 606:339–368, 2008.
- [221] K. Veroy, C. Prudhomme, D. V. Rovas, and A. T. Patera. A posteriori error bounds for reduced-basis approximation of parametrized noncoercive and nonlinear elliptic partial differential equations. In *Proceedings of the 16th AIAA Computational Fluid Dynamics Conference*, volume 3847, 2003.
- [222] S. Volkwein and A. Hepberger. Impedance identification by POD model reduction techniques (impedanz-identifikation mittels POD modellreduktion). *at-Automatisierungstechnik*, 56(8/2008):437–446, 2008.
- [223] Z. Wang, I. Akhtar, J. Borggaard, and T. Iliescu. Two-level discretizations of nonlinear closure models for proper orthogonal decomposition. *Journal of Computational Physics*, 230:126–146, 2011.
- [224] Z. Wang, I. Akhtar, J. Borggaard, and T. Iliescu. Proper orthogonal decomposition closure models for turbulent flows: A numerical comparison. *Computational Methods in Applied Mechanical Engineering*, 2012. In press.
- [225] Z. Wang and T. Iliescu. A mixed finite element method for the time-dependent quasi-geostrophic equations of the ocean. Submitted, 2011.
- [226] J. Weller, S. Camarri, and A. Iollo. Feedback control by low-order modelling of the laminar flow past a bluff body. *Journal of Fluid Mechanics*, 634(1):405–418, 2009.
- [227] J. Weller, E. Lombardi, and A. Iollo. Robust model identification of actuated vortex wakes. *Physica D: nonlinear phenomena*, 238(4):416–427, 2009.
- [228] K. Willcox. Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition. *Computers & fluids*, 35(2):208–226, 2006.

-
- [229] C. H. K. Williamson. Vortex dynamics in the cylinder wake. *Annual Review of Fluid Mechanics*, 28:477–539, 1996.
- [230] A. Yakhot, T. Anor, and G. E. Karniadakis. A reconstruction method for gappy and noisy arterial flow data. *IEEE Transactions on Medical Imaging*, 26(12):1681–1697, 2007.
- [231] B. Yildirim, C. Chryssostomidis, and G. E. Karniadakis. Efficient sensor placement for ocean measurements using low-dimensional concepts. *Ocean Modelling*, 27(3-4):160–173, 2009.
- [232] Y. Zhang, M. A. Henson, and Y. G. Kevrekidis. Nonlinear order reduction of discretized cell population models. In *Proceedings of the American Control Conference*, volume 3, pages 2383–2388. IEEE, 2003.