Numerical Analysis for Data-Driven Reduced Order Model Closures

Birgul Koc

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

Doctor of Philosophy
in
Mathematics

Traian Iliescu, Chair
Jeffrey T Borggaard
Serkan Gugercin
Honghu Liu

April 16, 2021
Blacksburg, Virginia

Keywords: Reduced Order Model, Proper Orthogonal Decomposition, Spatial Filter, Variational Multiscale, Data-Driven Model, Numerical Analysis, Long-Time Behavior

Numerical Analysis for Data-Driven Reduced Order Model Closures

Birgul Koc

(ABSTRACT)

This dissertation contains work that addresses both theoretical and numerical aspects of reduced order models (ROMs). In an under-resolved regime, the classical Galerkin reduced order model (G-ROM) fails to yield accurate approximations. Thus, we propose a new ROM, the data-driven variational multiscale ROM (DD-VMS-ROM) built by adding a closure term to the G-ROM, aiming to increase the numerical accuracy of the ROM approximation without decreasing the computational efficiency.

The closure term is constructed based on the variational multiscale framework. To model the closure term, we use data-driven modeling. In other words, by using the available data, we find ROM operators that approximate the closure term. To present the closure term's effect on the ROMs, we numerically compare the DD-VMS-ROM with other standard ROMs. In numerical experiments, we show that the DD-VMS-ROM is significantly more accurate than the standard ROMs. Furthermore, to understand the closure term's physical role, we present a theoretical and numerical investigation of the closure term's role in long-time integration. We theoretically prove and numerically show that there is energy exchange from the most energetic modes to the least energetic modes in closure terms in a long time averaging.

One of the promising contributions of this dissertation is providing the numerical analysis of the data-driven closure model, which has not been studied before. At both the theoretical and the numerical levels, we investigate what conditions guarantee that the small difference between the data-driven closure model and the full order model (FOM) closure term implies that the approximated solution is close to the FOM solution. In other words, we perform theoretical and numerical investigations to show that the data-driven model is verifiable.

Apart from studying the ROM closure problem, we also investigate the setting in which the G-ROM converges optimality. We explore the ROM error bounds' optimality by considering the difference quotients (DQs). We theoretically prove and numerically illustrate that both the ROM projection error and the ROM error are suboptimal without the DQs, and optimal if the DQs are used.

Numerical Analysis for Data-Driven Reduced Order Model Closures

Birgul Koc

(GENERAL AUDIENCE ABSTRACT)

In many realistic applications, obtaining an accurate approximation to a given problem can require a tremendous number of degrees of freedom. Solving these large systems of equations can take days or even weeks on standard computational platforms. Thus, lower-dimensional models, i.e., reduced order models (ROMs), are often used instead. The ROMs are computationally efficient and accurate when the underlying system has dominant and recurrent spatial structures.

Our contribution to reduced order modeling is adding a data-driven correction term, which carries important information and yields better ROM approximations. This dissertation's theoretical and numerical results show that the new ROM equipped with a closure term yields more accurate approximations than the standard ROM.

To my parents: Emine and Onur Koc.
To my siblings: Fatma Koc Nefes and Selim Koc.
To my sweet nieces: Zumra Nefes and Ayse Vera Nefes.

Acknowledgments

First and foremost, I am incredibly grateful to my advisor Dr. Traian Iliescu for his invaluable advice, continuous support, and patience during my Ph.D. study. His immense knowledge and great experience have encouraged me in my academic research and daily life. I would also like to thank you, Dr. Jeffrey T Borggaard, Dr. Serkan Gugercin, and Dr. Honghu Liu, for serving on my committee and for their valuable feedback and suggestions. Finally, I would like to express my gratitude to my parents, siblings, and my friends in Turkey and in the USA for their moral support and love. Without their tremendous understanding and encouragement over the past few years, it would be impossible to complete my study.

Contents

List of Figures x

List of Tables xiii

1 Introduction 1

1.1 Motivation . . . 2

1.2 Reduced Order Modeling . . . 2

1.3 Closure Modeling . . . 3

1.4 Mathematical foundations for ROM Closures . . . 4

1.5 Overview . . . 5

Bibliography 6

2 On Optimal Pointwise in Time Error Bounds and Difference Quotients for the Proper Orthogonal Decomposition 10

2.1 Abstract . . . 11

2.2 Introduction . . . 11

2.3 Proper Orthogonal Decomposition (POD) . . . 17

2.3.1 POD Without Difference Quotients (noDQ Case) . . . 17

2.3.2 POD With Difference Quotients (DQ Case) . . . 20

2.4 Pointwise Projection Error Estimates . . . 21

2.4.1 Pointwise Error Estimates: noDQ Case . . . 22

2.4.2 POD Pointwise Error Estimates: DQ Case . . . 26

2.5 Pointwise Error Estimates: DQ Case . . . 28

2.5.1 Error estimates . . . 31

2.5.2 Optimality of Pointwise ROM Discretization Errors . . . 34

2.6 Numerical Results . . . 39

2.6.1 Counterexample 1 . . . 40
2.6.2 Counterexample 2 . . . 45
2.7 Conclusions . . . 49

Bibliography **51**

3 Data-Driven Variational Multiscale Reduced Order Models **56**
3.1 Abstract . . . 57
3.2 Introduction . . . 57
3.3 Data-Driven Variational Multiscale Reduced Order Models (DD-VMS-ROMs) 60
3.3.1 Classical VMS . . . 60
3.3.2 Galerkin ROM (G-ROM) . . . 61
3.3.3 Two-Scale Data-Driven Variational Multiscale ROMs (2S-DD-VMS-ROM) . . . 62
3.3.4 Three-Scale Data-Driven Variational Multiscale ROMs (3S-DD-VMS-ROM) . . . 64
3.4 Numerical Results . . . 66
3.4.1 Computational Setting . . . 68
3.4.2 Burgers Equation . . . 71
3.4.3 Flow Past A Cylinder . . . 77
3.4.4 Quasi-Geostrophic Equations (QGE) . . . 92
3.4.5 Backward Facing Step . . . 96
3.4.6 Qualitative Comparison of 2S-DD-VMS-ROM and 3S-DD-VMS-ROM 102
3.5 Conclusions and Outlook . . . 105

Bibliography **107**

4 Long-Time Reynolds Averaging of ROMs for Fluid Flows **114**
4.1 Reduced order modeling . . . 117
4.1.1 On the spectral decomposition . . . 122
4.2 Preliminaries on long-time averages . . . 124

4.3 Average transfer of energy at equilibrium . . . 126
4.3.1 The POD case . . . 127
4.3.2 The spectral case . . . 128
4.4 Numerical results . . . 133
4.4.1 Numerical results with step function initial condition . . . 133
4.5 Conclusions . . . 139

Bibliography **141**

5 Verifiability of the Data-Driven Variational Multiscale Reduced Order Model **144**
5.1 Abstract . . . 145
5.2 Introduction . . . 145
5.3 Galerkin ROM (G-ROM) . . . 147
5.4 Large Eddy Simulation ROM (LES-ROM) . . . 149
5.5 Data Driven Variational Multiscale ROM (DD-VMS-ROM) . . . 152
5.6 Verifiability of the DD-VMS-ROM . . . 153
5.6.1 Definition of Verifiability and Mean Dissipativity . . . 153
5.6.2 Proof of DD-VMS-ROM's Verifiability . . . 153
5.7 Numerical Results . . . 158
5.7.1 Numerical Implementation . . . 158
5.7.2 Assessment of Results . . . 160
5.7.3 Burgers Equation . . . 160
5.7.4 Flow Past A Cylinder . . . 162
5.8 Conclusions and Future Work . . . 165

Bibliography **167**

6 Conclusions and Future Work **170**
6.1 Conclusions . . . 171

6.2 Future Work 172

Bibliography **174**

List of Figures

1.1 Flow past a cylinder, $Re = 1000$. Three-scale data-driven variational multiscale reduced order model with four modes [25]. 2

2.1 Counterexample 2 (2.46), FOM plot: $h = 1/4096$ and $\Delta t = 0.02$. 47

3.1 Burgers equation, $\nu = 10^{-3}$, reconstructive regime. FOM projection, G-ROM, 2S-DD-VMS-DDC-ROM, and 3S-DD-VMS-DDC-ROM plots for $r = 7$. . . . 73

3.2 Geometry of the flow past a circular cylinder numerical experiment. 77

3.3 Flow past a cylinder, $Re = 100$, reconstructive regime. Time evolution of the kinetic energy for FOM projection, G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values. 80

3.4 Flow past a cylinder, $Re = 100$, cross-validation regime. Time evolution of the kinetic energy for FOM projection, G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values. 81

3.5 Flow past a cylinder, $Re = 100$, predictive regime. Time evolution of the kinetic energy for FOM projection, G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values. 82

3.6 Flow past a cylinder, $Re = 500$, reconstructive regime. Time evolution of the kinetic energy for FOM projection, G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values. 85

3.7 Flow past a cylinder, $Re = 500$, cross-validation regime. Time evolution of the kinetic energy for FOM projection, G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values. 86

3.8 Flow past a cylinder, $Re = 500$, predictive regime. Time evolution of the kinetic energy for FOM projection, G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values. 87

3.9 Flow past a cylinder, $Re = 1000$, reconstructive regime. Time evolution of the kinetic energy for FOM projection, G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values. 89

3.10 Flow past a cylinder, $Re = 1000$, cross-validation regime. Time evolution of the kinetic energy for FOM projection, G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values. 90

3.11 Flow past a cylinder, $Re = 1000$, predictive regime. Time evolution of the kinetic energy for FOM projection, G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values. 91

3.12 QGE, $Re = 450$, $Ro = 0.0036$, reconstructive regime. Time evolution of the kinetic energy for FOM, G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values. 94

3.13 QGE, $Re = 450$, $Ro = 0.0036$, reconstructive regime. Time-averaged stream-function ψ over the interval $[10, 80]$ for FOM, G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values. 95

3.14 Backward facing step, $Re = 1000$. Geometry and finite element mesh (top). Magnitude of FOM velocity field at $t = 125$ (bottom). 96

3.15 Backward facing step, $Re = 1000$. Time evolution of the FOM kinetic energy. 97

3.16 Backward facing step, $Re = 1000$, reconstructive regime. Time evolution of the y-component of the velocity, v, of FOM, G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM with $r = 15$ at the point with coordinates $(19, 1)$. 99

3.17 Backward facing step, $Re = 1000$, reconstructive regime. Time evolution of the kinetic energy for FOM projection, G-ROM, 2S-DD-VMS-ROM and 3S-DD-VMS-ROM for different r values. 100

3.18 Backward facing step, $Re = 1000$, reconstructive regime. The spectrum of the y-component of the velocity for FOM, G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM with $r = 15$ at the point with coordinates $(19, 1)$. 101

3.19 Flow past a cylinder, $Re = 1000$, reconstructive regime. Time evolution of the y-component of the velocity, v, of the FOM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM with $r = 5$ at the point with coordinates $(0.43, 0.2)$. . . 103

3.20 Flow past a cylinder, $Re = 1000$, reconstructive regime. Time evolution of the first component of the subscales for the FOM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM with $r = 5$. 104

4.1 DNS solution obtained by using a piecewise linear finite element spatial discretization and the backward Euler time discretization. 134

5.1 Burgers equation (5.55), reconstructive regime: linear regression for $\mathcal{E}(L^2)$ and $\eta(L^2)$ for fixed r values and different tolerance values in the truncated SVD. 162

5.2 Geometry of the flow past a circular cylinder numerical experiment. 162

5.3 Flow past a cylinder, $Re = 100$, reconstructive regime: linear regression for $\mathcal{E}(L^2)$ and $\eta(L^2)$ for fixed r values and different tolerance values in the truncated SVD. 164

5.4 Flow past a cylinder, $Re = 1000$, reconstructive regime: linear regression for $\mathcal{E}(L^2)$ and $\eta(L^2)$ for fixed r values and different tolerance values in the truncated SVD. 165

List of Tables

2.1 Counterexample 1 (2.44), $\Delta t = 1/16$, noDQ case: Pointwise projection error (2.89) at each time step. . . . 42

2.2 Counterexample 1 (2.44), noDQ case: Scaling factor (2.90) for different time step values. . . . 42

2.3 Counterexample 1 (2.44), $\Delta t = 1/16$, DQ case: Pointwise projection error (2.89) at each time step. . . . 43

2.4 Counterexample 1 (2.44), DQ case: Scaling factor (2.92) for different time step values. . . . 43

2.5 Counterexample 1 (2.44), noDQ case: Ratio (2.95) for different time step values. . . . 44

2.6 Counterexample 1 (2.44), $k = 8$, noDQ case: Ratio (2.95) for different time step values. . . . 44

2.7 Counterexample 1 (2.44), DQ case: Ratio (2.97) for different time step values. 44

2.8 Counterexample 1 (2.44), $k = 8$, and DQ case: Ratio (2.97) for different time step values. . . . 45

2.9 Counterexample 2 (2.46), $r = 4$, and noDQ case: Scaling factor (2.99) for different time step values. . . . 46

2.10 Counterexample 2 (2.46), $r = 4$, and DQ case: Scaling factor (2.101) for different time step values. . . . 47

2.11 Counterexample 2 (2.46) and noDQ case: Ratio (2.103) for fixed time step $\Delta t = 0.01$ and different r values. . . . 48

2.12 Counterexample 2 (2.46) and DQ case: Ratio (2.105) for the shorter time interval $[0, 0.05]$ and fixed time step $\Delta t = 0.01$ and different r values. . . . 49

3.1 Burgers equation, $\nu = 10^{-3}$, reconstructive regime, optimal tol, tol_S, and tol_L. Average L^2 error for G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values. . . . 72

3.2 Burgers equation, $\nu = 10^{-3}$, reconstructive regime, $tol = tol_L = 10^2$, and optimal tol_S. Average L^2 error for G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values. . . . 74

3.3 Burgers equation, $\nu = 10^{-3}$, reconstructive regime, $tol = tol_L = 10^1$, and optimal tol_S. Average L^2 error for G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values. 74

3.4 Burgers equation, $\nu = 10^{-3}$, reconstructive regime, $tol = tol_L = 10^0$, and optimal tol_S. Average L^2 error for G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values. 74

3.5 Burgers equation, $\nu = 10^{-3}$, reconstructive regime $tol = tol_L = 10^{-1}$, and optimal tol_S. Average L^2 error for G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values. 75

3.6 Burgers equation, $\nu = 10^{-3}$, reconstructive regime: $tol = tol_S = 10^0$ and optimal tol_L. Average L^2 error for G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values. 75

3.7 Burgers equation, $\nu = 10^{-3}$, reconstructive regime: $tol = tol_S = 10^{-1}$ and optimal tol_L. Average L^2 error for G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values. 75

3.8 Burgers equation, $\nu = 10^{-3}$, reconstructive regime: $tol = tol_S = 10^{-2}$ and optimal tol_L. Average L^2 error for G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values. 76

3.9 Burgers equation, $\nu = 10^{-3}$, cross-validation regime, optimal tol, tol_S, and tol_L. Average L^2 error for G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values. 76

3.10 Burgers equation, $\nu = 10^{-3}$, predictive regime, optimal tol, tol_S, and tol_L. Average L^2 error for G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values. 77

3.11 Flow past a cylinder, $Re = 100$, reconstructive regime. Average L^2 errors for G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values. . 79

3.12 Flow past a cylinder, $Re = 100$, cross-validation regime. Average L^2 errors for G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values. 79

3.13 Flow past a cylinder, $Re = 100$, predictive regime. Average L^2 errors for G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values. . 80

3.14 Flow past a cylinder, $Re = 500$, reconstructive regime. Average L^2 errors for G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values. . 83

3.15 Flow past a cylinder, $Re = 500$, cross-validation regime. Average L^2 errors for G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values. 83

3.16 Flow past a cylinder, $Re = 500$, predictive regime. Average L^2 errors for G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values. . 84

3.17 Flow past a cylinder, $Re = 1000$, reconstructive regime. Average L^2 errors for G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values. 88

3.18 Flow past a cylinder, $Re = 1000$, cross-validation regime. Average L^2 errors for G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values. 88

3.19 Flow past a cylinder, $Re = 1000$, predictive regime. Average L^2 errors for G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values. . 90

3.20 QGE, $Re = 450$, $Ro = 0.0036$, reconstructive regime. L^2 errors of the time-averaged streamfunction for G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values. 93

3.21 Backward facing step, $Re = 1000$, reconstructive regime. Average L^2 errors for G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values. 98

4.1 Case 1: Time-averages of $\mathrm{e}_m(u)$ and $\mathcal{E}_m(u)$ for $d = 37$ and different m values. 134

4.2 Case 2: Time-averages of $\mathrm{e}_m(u)$ and $\mathcal{E}_m(u)$ for $d = 38$, $\Delta t = 10^{-2}$, 1000 equally spaced quadrature points, and different m values. 135

4.3 Case 2: Time-averages of $\mathrm{e}_m(u)$ and $\mathcal{E}_m(u)$ for $d = 41$, $\Delta t = 10^{-3}$, 10000 equally spaced quadrature points, and different m values. 136

4.4 Case 2: Time-averages of $\mathrm{e}_m(u)$ and $\mathcal{E}_m(u)$ for $d = 43$, $\Delta t = 10^{-4}$, 10000 equally spaced quadrature points, and different m values. 136

4.5 Case 2: Time-averages of $\mathrm{e}_m(u)$ and $\mathcal{E}_m(u)$ for $d = 43$, $\Delta t = 2 * 10^{-5}$, 10000 equally spaced quadrature points, and different m values. 137

4.6 Case 3: Time-averages of $\mathrm{e}_m(u)$ for $d = 36$, $\Delta t = 10^{-2}$, different m values, and all subintervals used in the composite trapezoidal rule. 138

4.7 Case 3: Time-averages of $\mathcal{E}_m(u)$ for $d = 36$, $\Delta t = 10^{-2}$, different m values, and all subintervals used in the composite trapezoidal rule. 138

4.8 Case 4: Time-averages of $\mathrm{e}_m(u)$ and $\mathcal{E}_m(u)$ for $\Delta t = 2 * 10^{-5}$, different m values, $d = 18$, and 5000 equally spaced subintervals used in the composite trapezoidal rule. 139

4.9 Case 4: Time-averages of $\mathrm{e}_m(u)$ and $\mathcal{E}_m(u)$ for $\Delta t = 10^{-5}$, different m values, $d = 18$, and 5000 equally spaced subintervals used in the composite trapezoidal rule. 139

5.1 Burgers equation (5.55), reconstructive regime: $\mathcal{E}(L^2)$ and $\eta(L^2)$ values for fixed r values and different tolerance values in the truncated SVD. 161

5.2 Flow past a cylinder, $Re = 100$, reconstructive regime: $\mathcal{E}(L^2)$ and $\eta(L^2)$ values for fixed r values and different tolerance values in the truncated SVD. 163

5.3 Flow past a cylinder, $Re = 1000$, reconstructive regime: $\mathcal{E}(L^2)$ and $\eta(L^2)$ values for fixed r values and different tolerance values in the truncated SVD. 164

List of Abbreviations

ν Viscosity coefficient

Re Reynolds number

Ro Rossby number

2S-DD-VMS-ROM Two-Scale Data-Driven Variational Multiscale Reduced Order Model

3S-DD-VMS-ROM Three-Scale Data-Driven Variational Multiscale Reduced Order Model

CN Crank-Nicolson

CN POD-G-ROM Crank-Nicolson Proper Orthogonal Decomposition Galerkin Reduced Order Model

DD-VMS-ROM Data-Driven Variational Multiscale Reduced Order Model

DEIM Discrete Empirical Interpolation Method

DNS Direct Numerical Simulation

DQs Difference Quotients

EIM Empirical Interpolation Method

FE Finite Element

FEM Finite Element Method

FOM Full Order Model

G-ROM Galerkin ROM

LES Large Eddy Simulation

MZ Mori-Zwanzig

MZ-ROM Mori-Zwanzig Reduced Order Model

PDE Partial Differential Equation

POD Proper Orthogonal Decomposition

QGE Quasi-Geostrophic equations

RANS Reynolds-averaged Navier-Stokes

RBM Reduced Basis Method

ROM Reduced Order Model

SNS Solution-Based Nonlinear Subspace

SVD Singular Value Decomposition

VMS Variational Multiscale

VMS-ROM Variational Multiscale Reduced Order Model

Chapter 1

Introduction

1.1 Motivation

Obtaining high-fidelity numerical simulations plays an essential role in engineering and scientific applications. One needs to use millions and even billions of degrees of freedom to get a high-fidelity numerical approximation of complex fluid flows. In some design and control applications, theirs dominant and recurrent structures alleviate the computational burden. Reduced order models (ROMs) are low-dimensional models for the numerical simulation of linear and nonlinear systems. For structure dominated system, [11, 14, 15, 17, 27, 31, 32, 40, 41, 42], the ROMs can decrease the full order model (FOM) computational cost by orders of magnitude without losing the flows' key features.

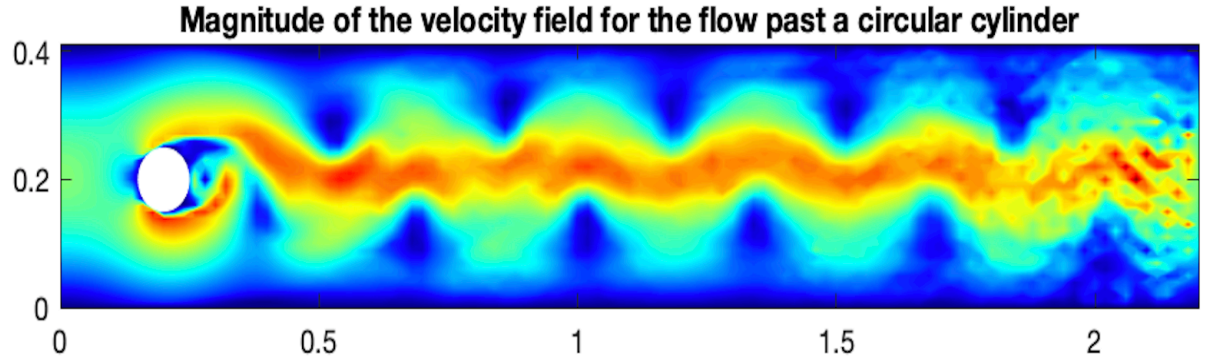

Figure 1.1: Flow past a cylinder, $Re = 1000$. Three-scale data-driven variational multiscale reduced order model with four modes [25].

1.2 Reduced Order Modeling

ROMs for fluid flows have been used to successfully minimize the computational cost of scientific and engineering applications dominated by a small number of recurring dominant spatial structures. Two of the most popular techniques to construct ROMs for fluid flows are the proper orthogonal decomposition (POD) [12, 17, 27, 43] and reduced basis methods (RBM) [8, 33, 38]. In this dissertation, we use the POD technique and we seek a low-dimensional ROM that produces an accurate approximate for a parabolic partial differential equation (PDE)

$$\dot{\boldsymbol{u}} = \boldsymbol{f}(\boldsymbol{u}), \tag{1.1}$$

where $\boldsymbol{f}$ could be a linear or nonlinear function. In an offline stage, the available FOM data (called snapshots), which is the solution of (1.1) at different time instances, are used to construct a low-dimensional space-dependent ROM basis $\{\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_d\}$, which is then utilized together with the Galerkin projection to build the ROM operators.

The Galerkin ROM (G-ROM) is one of the most common types of ROMs for fluid flows [2, 18, 20, 24, 27]. The implementation of the G-ROM framework is straightforward and can be summarized in the following algorithm:

Algorithm 1 : Galerkin ROM (G-ROM)

1: choose dominant modes $\{\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_r\}$, $r << d$ (where d is the rank of the snapshot matrix) that contain the largest relative kinetic energy and represent the recurrent spatial structures of (1.1);

2: construct a ROM approximation $\boldsymbol{u}_r$ as a linear combination of ROM basis functions $\boldsymbol{\varphi}_i$ with time-dependent ROM coefficients $\boldsymbol{a}_i$, i.e., $\boldsymbol{u}_r = \sum_{i=1}^{r} \boldsymbol{a}_i(t)\boldsymbol{\varphi}_i(\boldsymbol{x})$;

3: replace $\boldsymbol{u}$ in (1.1) with the ROM solution $\boldsymbol{u}_r$ constructed in step 2;

4: use the Galerkin projection, which projects the resulting system in step 3 onto the ROM space X^r spanned by $\{\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_r\}$.

The steps summarized above lead us to solve the G-ROM for the time-dependent coefficient $\boldsymbol{a}$:

$$\dot{\boldsymbol{a}} = \boldsymbol{F}(\boldsymbol{a}), \tag{1.2}$$

where $\boldsymbol{F}$ contains the ROM operators. In the online stage, the low-dimensional G-ROM (1.2) can be repeatedly tested for a regime that is different from the training regime in which the ROM basis functions and ROM operators are constructed.

1.3 Closure Modeling

In POD, the ROM basis functions are constructed by solving an eigenvalue problem [43]. The largest eigenvalues correspond to the most important ROM basis functions, i.e., the spatial structures that dominate the dynamics of the underlying system. Thus, the rate of decay of the eigenvalues is a good indicator to understand how many ROM basis functions are required to approximate the given problem accurately. For a structure-dominated system, (i.e., a system dominated by relatively few recurrent spatial structures) since the eigenvalues of the ROM basis functions decay fast, using relatively few ROM basis functions is enough to capture the main dynamics of this system. However, for other systems (e.g., a convection-dominated system), the eigenvalues' decaying rate is not as fast as in the structure-dominated system. Thus, using a relatively small number of ROM basis functions often yields an inaccurate approximation. This numerical simulation in which the number of ROM modes in not large enough to capture the dynamics of the underlying system is called an under-resolved simulation.

In our work, we increase the numerical accuracy of an under-resolved simulation in two different ways:

(i) increasing the ROM dimension, i.e., r,

(ii) adding a low-dimensional closure term.

(See also [1, 4, 6, 7, 13, 21, 22, 23, 26, 28, 29, 30, 34, 35, 36, 37, 39, 45] for related, but different work.) Since numerical efficiency is one of the big advantages of the ROM, we want to increase the numerical accuracy while preserving the computational efficiency. Thus, we add a low-dimensional closure term $Closure(\boldsymbol{a})$ to the G-ROM and solve the following closed ROM,

$$\dot{\boldsymbol{a}} = \boldsymbol{F}(\boldsymbol{a}) + \text{Closure}(\boldsymbol{a}), \tag{1.3}$$

where the $Closure(\boldsymbol{a})$ term models the interaction between the unresolved ROM basis functions $\{\boldsymbol{\varphi}_{r+1}, \ldots, \boldsymbol{\varphi}_d\}$ and the resolved ROM basis functions $\{\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_r\}$.

Many researchers use the closure problem in the numerical simulation of complex systems for various purposes. For example, in the finite element method (FEM), researchers use the closure model to address the sub-grid scale effects. In large eddy simulation (LES), the closure term is built around physical insight stemming from Kolmogorov's statistical theory of turbulence. However, in this dissertation, we use *available data* to model the closure term in a ROM setting.

1.4 Mathematical foundations for ROM Closures

Over the last two decades, ROM closure modeling has witnessed a dynamic development. Different types of ROM closure models have been proposed. Functional ROM closures are constructed by using physical insight. Classical examples in this class include eddy viscosity models [44]. Structural ROM closures are a different class of models that are developed by using mathematical arguments. Examples in this class include the approximate deconvolution ROM [46], the Mori-Zwanzig formalism [16], and the parameterized manifolds [9, 10]. The most active research area in ROM closure modeling is in the development of data-driven ROM closures in which experimental or numerical data is utilized to build the ROM closure model. An example of data-driven ROM closure is the data-driven variational multiscale ROM that we present in this dissertation.

Despite the recent increased interest in ROM closure modeling, the mathematical foundations of these new ROM closures are relatively scarce. For example, fundamental questions in the numerical analysis of ROM closure modeling, e.g., stability and convergence, are still wide opened for many of these models. To our knowledge, the first numerical analysis of ROM

closures was performed in [5], where the Smagorinsky model was analyzed in a simplified setting. Next, the numerical analysis of eddy viscosity variational multiscale ROMs was carried out in [19]. Finally, the numerical analysis of the Samgorinsky model in a reduced basis method setting was performed in [3].

In this dissertation, we take a next step in the development of numerical analysis for ROM closures and prove verifiability for the data-driven variational mulsticale ROM. To our knowledge, this is the first time mathematical support for data-driven ROM closures is provided.

1.5 Overview

The dissertation is structured in four chapters. We investigate the parameter scaling for which the G-ROM converges optimally in Chapter 2. Chapter 3 aims to find the optimal model for the data-driven ROM closure term that yields the most accurate solutions. Chapter 4 presents a theoretical and numerical investigation of the ROM closure term's role in long-time integration. Finally, Chapter 5 investigates the numerical analysis for a data-driven ROM closure model which, to our knowledge, has not been addressed in the ROM community yet.

Chapter 2 is dedicated to addressing some critical issues related to optimal pointwise in time error bounds for the POD and ROM by using the heat equation. In particular, we study how difference quotients (DQs) affect the ROM error bounds' optimality when the time discretization and the ROM discretization error are considered. We theoretically and numerically prove that both the ROM projection error and the ROM error are suboptimal without the DQs, and optimal if the DQs are used.

Chapter 3 proposes a new ROM framework, the data-driven variational multiscale ROM (DD-VMS-ROM), in which the classical G-ROM is supplemented with one or more correction terms. Our numerical results show that the DD-VMS-ROM increases the numerical accuracy when the G-ROM yields inaccurate approximations in under-resolved simulations.

Chapter 4 addresses the time-average energy exchange between the resolved and unresolved POD modes in closure term, i.e., investigates whether the closure term is dissipative in the mean. We present a theoretical and numerical investigation of the role played by the closure term in long-time integration. Both experiments prove that the closure term has to dissipate energy on the average.

Chapter 5 proposes numerical analysis for a data-driven closure model. We investigate under which conditions a small difference between the ROM closure term and the FOM closure term implies that the ROM solution is close to the FOM solution. In other words, we theoretically and numerically demonstrate the verifiability of the data-driven ROM closure model.

Chapter 6 presents the conclusions and outlines several directions of future research.

Bibliography

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

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

[3] F. Ballarin, T. C. Rebollo, E. D. Ávila, M. G. Mármol, and G. Rozza. Certified reduced basis VMS-Smagorinsky model for natural convection flow in a cavity with variable height. *Computers & Mathematics with Applications*, 80(5):973–989, 2020.

[4] F. Bernard, A. Iollo, and S. Riffaud. Reduced-order model for the BGK equation based on POD and optimal transport. *J. Comput. Phys.*, 373:545–570, 2018.

[5] J. Borggaard, T. Iliescu, H. Lee, J. P. Roop, and H. Son. A two-level discretization method for the Smagorinsky model. *Multiscale Modeling & Simulation*, 7(2):599–621, 2008.

[6] N. Cagniart, Y. Maday, and B. Stamm. Model order reduction for problems with large convection effects. In *Contributions to Partial Differential Equations and Applications*, pages 131–150. Springer, 2019.

[7] K. Carlberg. Adaptive h-refinement for reduced-order models. *Int. J. Num. Meth. Eng.*, 102(5):1192–1210, 2015.

[8] T. Chacón Rebollo, E. Delgado Ávila, and M. M. Gómez Mármol. Reduced basis method for the Smagorinsky model. *Recent developments in numerical methods for model reduction (2016)*, 2016.

[9] M. D. Chekroun, H. Liu, and J. C. McWilliams. Variational approach to closure of nonlinear dynamical systems: Autonomous case. *Journal of Statistical Physics*, 179:1073–1160, 2020.

[10] M. D. Chekroun, H. Liu, and S. Wang. *Stochastic parameterizing manifolds and non-Markovian reduced equations: stochastic manifolds for nonlinear SPDEs II*. Springer, 2014.

[11] D. T. Crommelin and A. J. Majda. Strategies for model reduction: comparing different optimal bases. *J. Atmos. Sci.*, 61:2206–2217, 2004.

[12] H. Fareed and J. R. Singler. A note on incremental POD algorithms for continuous time data. *arXiv preprint arXiv:1807.00045*, 2018.

[13] J.-F. Gerbeau and D. Lombardi. Approximated Lax pairs for the reduced order integration of nonlinear evolution equations. *J. Comput. Phys.*, 265:246–269, 2014.

[14] M. Gunzburger, N. Jiang, and M. Schneier. An ensemble-proper orthogonal decomposition method for the nonstationary Navier-Stokes equations. *SIAM J. Numer. Anal.*, 55(1):286–304, 2017.

[15] J. S. Hesthaven, G. Rozza, and B. Stamm. *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*. Springer, 2015.

[16] C. Hijón, P. Español, E. Vanden-Eijnden, and R. Delgado-Buscalioni. Mori-Zwanzig formalism as a practical computational tool. *Faraday discussions*, 144:301–322, 2010.

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

[18] P. Holmes, J. L. Lumley, G. Berkooz, and C. W. Rowley. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry, second edition*. Cambridge university press, 2012.

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

[20] B. Koc, M. Mohebujjaman, C. Mou, and T. Iliescu. Commutation error in reduced order modeling of fluid flows. *Adv. Comput. Math.*, 45(5-6):2587–2621, 2019.

[21] H. Lu and D. M. Tartakovsky. Lagrangian dynamic mode decomposition for construction of reduced-order models of advection-dominated phenomena. *J. Comput. Phys.*, page 109229, 2020.

[22] D. J. Lucia. Reduced order modeling for high speed flows with moving shocks. Technical report, Air Force Inst. of Tech., Wright-Patterson Air Force Base, OH, 2001.

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

[24] C. Mou, B. Koc, O. San, L. G. Rebholz, and T. Iliescu. Data-driven variational multiscale reduced order models. *Computer Methods in Applied Mechanics and Engineering*, 373:113470, 2021.

[25] C. Mou, H. Liu, D. R. Wells, and T. Iliescu. Data-driven correction reduced order models for the quasi-geostrophic equations: A numerical investigation. *Int. J. Comput. Fluid Dyn.*, pages 1–13, 2020.

[26] N. J. Nair and M. Balajewicz. Transported snapshot model order reduction approach for parametric, steady-state fluid flows containing parameter-dependent shocks. *Int. J. Num. Meth. Engng.*, 117(12):1234–1262, 2019.

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

[28] M. Nonino, F. Ballarin, G. Rozza, and Y. Maday. Overcoming slowly decaying Kolmogorov n-width by transport maps: application to model order reduction of fluid dynamics and fluid–structure interaction problems. *arXiv preprint arXiv:1911.06598*, 2019.

[29] M. Ohlberger and S. Rave. Nonlinear reduced basis approximation of parameterized evolution equations via the method of freezing. *C. R. Math.*, 351(23-24):901–906, 2013.

[30] B. Peherstorfer. Model reduction for transport-dominated problems via online adaptive bases and adaptive sampling. *arXiv preprint arXiv:1812.02094*, 2018.

[31] S. Perotto, A. Reali, P. Rusconi, and A. Veneziani. HIGAMod: A Hierarchical IsoGeometric Approach for MODel reduction in curved pipes. *Comput. & Fluids*, 142:21–29, 2017.

[32] A. Quarteroni, A. Manzoni, and F. Negri. *Reduced Basis Methods for Partial Differential Equations: An Introduction*, volume 92. Springer, 2015.

[33] A. Quarteroni, G. Rozza, et al. *Reduced order methods for modeling and computational reduction*, volume 9. Springer, 2014.

[34] J. Reiss, P. Schulze, J. Sesterhenn, and V. Mehrmann. The shifted proper orthogonal decomposition: A mode decomposition for multiple transport phenomena. *SIAM J. Sci. Comput.*, 40(3):A1322–A1344, 2018.

[35] D. Rim, S. Moe, and R. J. LeVeque. Transport reversal for model reduction of hyperbolic partial differential equations. *SIAM-ASA J. Uncertain.*, 6(1):118–150, 2018.

[36] C. W. Rowley, I. G. Kevrekidis, J. E. Marsden, and K. Lust. Reduction and reconstruction for self-similar dynamical systems. *Nonlinearity*, 16(4):1257, 2003.

[37] C. W. Rowley and J. E. Marsden. Reconstruction equations and the Karhunen–Loève expansion for systems with symmetry. *Phys. D*, 142(1-2):1–19, 2000.

[38] G. Rozza and K. Veroy. On the stability of the reduced basis method for Stokes equations in parametrized domains. *Comput. Methods Appl. Mech. Engrg.*, 196(7):1244–1260, 2007.

[39] O. San and J. Borggaard. Principal interval decomposition framework for POD reduced-order modeling of convective Boussinesq flows. *Int. J. Num. Meth. Fluids*, 78(1):37–62, 2015.

[40] T. P. Sapsis and P. F. J. Lermusiaux. Dynamically orthogonal field equations for continuous stochastic dynamical systems. *Phys. D*, 238(23-24):2347–2360, 2009.

[41] R. Ştefănescu, A. Sandu, and I. M. Navon. POD/DEIM reduced-order strategies for efficient four dimensional variational data assimilation. *J. Comput. Phys.*, 295:569–595, 2015.

[42] K. Taira, M. S. Hemati, S. L. Brunton, Y. Sun, K. Duraisamy, S. Bagheri, S. T. M. Dawson, and C.-A. Yeh. Modal analysis of fluid flows: Applications and outlook. *AIAA J.*, pages 1–25, 2019.

[43] S. Volkwein. Proper orthogonal decomposition: Theory and reduced-order modelling. *Lecture Notes, University of Konstanz*, 2013. http://www.math.uni-konstanz.de/numerik/personen/volkwein/teaching/POD-Book.pdf.

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

[45] G. Welper. *h* and *hp*-adaptive interpolation by transformed snapshots for parametric and stochastic hyperbolic PDEs. *arXiv preprint arXiv:1710.11481*, 2017.

[46] X. Xie, D. Wells, Z. Wang, and T. Iliescu. Approximate deconvolution reduced order modeling. *Comput. Methods Appl. Mech. Engrg.*, 313:512–534, 2017.

Chapter 2

On Optimal Pointwise in Time Error Bounds and Difference Quotients for the Proper Orthogonal Decomposition

This chapter is submitted and is under revision in *SIAM Journal on Numerical Analysis (SINUM) journal.* [*]

In that paper, my contribution was being part of the theoretical development and performing numerical experiments for the heat equation in Section 2.6.

[*] **B. Koc**, S. Rubino, M. Schneier, J.R. Singler, and T. Iliescu. On optimal pointwise in time error bounds and difference quotients for the proper orthogonal decomposition.arXiv preprint arXiv:2010.03750, 2020.

2.1 Abstract

In this chapter, we resolve several long standing issues dealing with optimal pointwise in time error bounds for proper orthogonal decomposition (POD) reduced order modeling of the heat equation. In particular, we study the role played by difference quotients (DQs) in obtaining reduced order model (ROM) error bounds that are optimal with respect to both the time discretization error and the ROM discretization error. When the DQs are not used, we prove that both the ROM projection error and the ROM error are suboptimal. When the DQs are used, we prove that both the ROM projection error and the ROM error are optimal. The numerical results for the heat equation support the theoretical results.

2.2 Introduction

In this chapter, we consider the one-dimensional heat equation

$$u_t - \nu\, u_{xx} = f\,, \tag{2.1}$$

where the spatial domain is $[0,1]$, the time domain is $[0,T]$, and ν is the diffusion coefficient. For simplicity, we consider homogeneous Dirichlet boundary conditions $u(0,t) = u(1,t) = 0$ for $t > 0$ and given initial conditions $u(x,0) = u_0(x)$.

We also consider projection reduced order models (ROMs) for the heat equation. (See also [9, 47] where system theoretical methods are applied at the PDE level.) Specifically, we consider the proper orthogonal decomposition (POD) [25], which can be summarized as follows: (i) The full order model (FOM) for (2.1) is run for selected parameter values and/or time intervals to generate a set of snapshots $\{u^0, u^1, \ldots, u^N\}$; (ii) These snapshots and the singular value decomposition (SVD) are used to construct an orthonormal ROM basis $\{\varphi_1, \ldots, \varphi_s\}$ for a Hilbert space $\mathcal{H}$, where s is the rank of the snapshot matrix; (iii) The ROM approximation

$$u(x,t_n) \approx u_r^n(x) = \sum_{j=1}^{r} u_j^n\, \varphi_j(x)\,, \qquad n = 1, \ldots, N\,, \tag{2.2}$$

where $r < s$ is the ROM dimension, is used together with a Galerkin projection and a time discretization to yield a system of equations for u_j^n, which are the sought ROM coefficients.

Definition 2.1 (Generic Constant C)**.** For clarity, in what follows, we will denote by C a generic positive constant that may vary from a line to another, but which is always independent of the discretization parameters.

In the pioneering paper [36], Kunisch and Volkwein laid the foundations of numerical analysis for POD (see, e.g., [37, 42, 46] for relevant work).

In particular, for the ROM error

$$e^n(x) = u(x, t_n) - u_r^n(x)\,, \qquad n = 1, \ldots, N\,, \tag{2.3}$$

they proved the following error bound (see Theorem 7 in [36]):

$$\frac{1}{N+1}\sum_{n=1}^{N} \|e^n\|_{L^2}^2 \leq C\left(\text{time discretization error} + \text{ROM discretization error}\right). \tag{2.4}$$

This estimate was later extended to include the spatial discretization error and a pointwise in time estimate in [31], (see, e.g., [35, 46] for alternative pointwise in time estimates) i.e.,

$$\|e^n\|_{L^2} \leq C\Big(\text{space discretization error} + \text{time discretization error} + \text{ROM discretization error}\Big)\,.$$

Estimate (2.5) relied on an assumption about the POD projection error, which roughly says that the POD projection error at each time step is of the same order as the POD projection error at the remaining time steps. This assumption has since been generally used in proving pointwise in time error bounds for parabolic equations.

We emphasize that the error bound (2.5) includes all three ROM error sources: (i) the space discretization error, which results from the spatial discretization of the heat equation (2.1) with classical numerical methods, e.g., finite elements (FEs); (ii) the time discretization error, which results from the time discretization of the heat equation (2.1) with classical numerical methods, e.g., Euler or Crank-Nicolson methods; and (iii) the ROM discretization error, which results from the truncation in (2.2).

Pointwise error bounds like (2.5) without the ROM discretization error are standard in the numerical analysis of classical discretization methods for the heat equation (see, e.g., [54]). Pointwise errors bounds such as (2.5) are desirable since they eliminate the possibility of an error "spike" at a certain point in time, which in principle is allowed with a bound of the form (2.4). Our goal here is to better understand these pointwise error bounds in the context of POD reduced order modeling.

A fundamental issue in the POD numerical analysis is the *optimality* of the error bound (2.5). We emphasize that there are three types of optimality, corresponding to the three types of discretization levels: (i) space discretization optimality; (ii) time discretization optimality; and (iii) ROM discretization optimality. We discuss each optimality type below:

Space Discretization Optimality For simplicity, we consider a FE spatial discretization. We emphasize, however, that other standard numerical methods (e.g., finite difference, spectral,

or spectral element methods) could be considered. An error bound is optimal with respect to the spatial discretization if the error scalings with respect to the spatial discretization parameters only are of the following form:

$$\|e^n\|_{L^2} = \mathcal{O}(h^{m+1}), \tag{2.5}$$

$$\|\nabla e^n\|_{L^2} = \mathcal{O}(h^m), \tag{2.6}$$

where h is the size of the FE mesh and m is the FE order. Proving estimates that are optimal with respect to the spatial discretization is relatively straightforward (see, e.g., [16, 29, 31]), since it follows the standard FE numerical analysis [54]. Thus, the spatial discretization error component is generally ignored in POD numerical analysis papers (see, e.g., [36]). To simplify the presentation, we will not discuss the spatial discretization optimality in this paper. Thus, the spatial discretization error component is generally ignored in POD numerical analysis papers (see, e.g., [36, 57]). To simplify the presentation, we will not discuss the spatial discretization optimality in this paper. We note, however, that our results can be extended in a straightforward mannner to include the spatial discretization optimality.

Time Discretization Optimality An error bound is optimal with respect to the time discretization if the error scalings with respect to the time discretization parameters only are of the following form:

$$\|e^n\|_{L^2} = \mathcal{O}(\Delta t^k), \tag{2.7}$$

where Δt is the time step size used in the time discretization, and k is the time discretization order (e.g., $k = 1$ for Euler's method, and $k = 2$ for Crank-Nicolson).

The importance of the time discretization optimality was recognized early on. In Remark 1 of [36], Kunisch and Volkwein proposed the *difference quotients (DQs)* (i.e., scaled snapshots of the form $(u^n - u^{n-1})/\Delta t$, $n = 1, \ldots, N$) as a means to achieve time discretization optimality. (We note that including state-derivative information in the frequency domain is common in interpolatory model reduction methods; see, for example, [2].) Specifically, on page 121 of [36], the authors noted that, in the DQ case (i.e., if the DQs are used to build the POD basis), time discretization optimal error bounds of the type (2.7) follow. However, in the noDQ case (i.e., if the DQs are not used), the norm squared error bound has a suboptimal (Δt^{-2}) factor.

A major development in the study of POD optimality was made by Chapelle, Gariah, and Sainte-Marie in [6]. The authors showed that using the L^2 projection instead of the Ritz projection used in [36] (which is standard in the FE numerical analysis [54, 58]) avoids the difficulties posed by the POD approximation of the time derivative. Pointwise error bounds were not considered in [6]; however, it can be checked that using the L^2 projection eliminates the need to use DQs to achieve time discretization optimality if the pointwise POD projection error assumption mentioned earlier is made.

ROM Discretization Optimality The first discussion of the ROM discretization optimality was presented in [30]. In that work, a pointwise in time error bound was said to be optimal with respect to the ROM discretization if the error scalings with respect to the ROM discretization parameters only take one of the following forms:

$$\|e^n\|_{L^2}^2 = \mathcal{O}\left(\frac{1}{N+1}\sum_{n=0}^{N}\|\eta^{proj}(t_n)\|_{L^2}^2\right) = \mathcal{O}\left(\sum_{i=r+1}^{s}\lambda_i\right), \tag{2.8}$$

$$\|\nabla e^n\|_{L^2}^2 = \mathcal{O}\left(\frac{1}{N+1}\sum_{n=0}^{N}\|\nabla\eta^{proj}(t_n)\|_{L^2}^2\right) = \mathcal{O}\left(\sum_{i=r+1}^{s}\lambda_i\|\nabla\varphi_i\|_{L^2}^2\right), \tag{2.9}$$

where η^{proj} is the *POD projection error*, which is defined as

$$\eta^{proj}(x,t) = u(x,t) - \sum_{i=1}^{r}\Big(u(\cdot,t),\varphi_i(\cdot)\Big)_{\mathcal{H}}\varphi_i(x), \tag{2.10}$$

and λ_i and φ_i are POD eigenvalues and modes. The first significant development in the study of POD optimality was made in [30], where it was shown (utilizing the technique from [6]) that not using the DQs yields pointwise error bounds that are suboptimal with respect to the ROM discretization. (We note that the optimality with respect to the time discretization was not considered in [30].) The first significant development in the study of POD optimality was made in [30], where it was shown that not using the DQs yields error bounds that may be optimal with respect to the time discretization (using the technique from [6]), but are suboptimal with respect to the ROM discretization. Specifically, in the noDQ case, it was shown in [30] that

$$\|e^n\|_{L^2}^2 = \mathcal{O}\left(\frac{1}{N+1}\sum_{n=0}^{N}\|\nabla\eta^{proj}(t_n)\|_{L^2}^2\right) = \mathcal{O}\left(\sum_{i=r+1}^{s}\lambda_i\|\nabla\varphi_i\|_{L^2}^2\right), \tag{2.11}$$

which is suboptimal with respect to the ROM discretization. Furthermore, in the DQ case, it was shown [30] that

$$\|e^n\|_{L^2}^2 = \mathcal{O}\left(\frac{1}{N+1}\sum_{n=0}^{N}\|\eta^{proj}(t_n)\|_{L^2}^2\right) = \mathcal{O}\left(\sum_{i=r+1}^{s}\lambda_i\right), \tag{2.12}$$

which is optimal with respect to the ROM discretization. However, two assumptions on the POD projection errors were made in order to establish these results.

To summarize, the current state-of-the-art in POD optimality *suggests* that

$$\boxed{\text{DQs are needed for optimal POD error bounds.}} \tag{2.13}$$

We emphasize that, to our knowledge, (2.13) *has never been proved*. Indeed, [36] focused on the time discretization optimality, but ignored the ROM discretization optimality. Specifically, the authors proved that using DQs yields error bounds that are optimal with respect

to the time discretization, but not necessarily with respect to the ROM discretization. In [6], the authors considered the noDQ case and developed a framework that yields error bounds that are optimal with respect to the time discretization, but not necessarily with respect to the ROM discretization. A completely different approach was taken in [30], where the focus was on ROM discretization optimality, without considering the time discretization optimality. Specifically, in [30] it was shown both theoretically and numerically that, in the noDQ case the error bounds are suboptimal with respect to the ROM discretization error, whereas in the DQ case the error bounds are optimal. The time discretization optimality was ignored in [30].

In this paper, we prove (2.13). Specifically, we make three main contributions:

First, in the noDQ case, we prove that the POD error bound is suboptimal not only with respect to the ROM discretization (as shown in [30]), but also with respect to the time discretization. Specifically, we show that the pointwise POD projection error assumption mentioned earlier can fail and the scaling of the error bound (2.11) with respect to the ROM discretization can degrade to

$$\|e^n\|_{L^2}^2 = \mathcal{O}\left(\Delta t^{-1} \sum_{i=r+1}^{s} \lambda_i\right) + \mathcal{O}\left(\sum_{i=r+1}^{s} \lambda_i \|\nabla \varphi_i\|_{L^2}^2\right). \tag{2.14}$$

In particular, we construct two analytical examples, and we prove that they satisfy (2.14) in the noDQ case. We note that the bound (2.14) is a significant improvement over the bound (2.11) proved in [30], since the latter did not display the time discretization suboptimality.

Our second main contribution is that we prove new pointwise in time error bounds in the DQ case, and we do not require any of the assumptions used in [30] to establish similar pointwise bounds. All of these error bounds are optimal with respect to the time discretization. One key component of our analysis is that we prove that an assumption from [30, 31] concerning pointwise in time behavior of POD projection errors is automatically satisfied in the DQ case.

Our third main contribution is that we revisit the definition of ROM discretization error optimality, introduce a new stronger notion of optimality, and show that all of the pointwise in time error bounds in the DQ case are optimal in at least one sense. Both pointwise in time error bounds using the H_0^1 norm are optimal in the new stronger sense; the pointwise in time bounds using the L^2 norm can be optimal in either sense. We note that to prove the stronger optimality of the L^2 error bounds, we do need a uniform boundedness assumption of the type made in [30].

We emphasize that we do not attempt here to prove error bounds for the POD ROM when the parameter ν or the initial data are different from those used to generate the POD basis. As in [36] and many other POD numerical analysis works, our main goal here is to attempt to understand and begin to explain the approximation errors of POD reduced order models

for PDEs. The much more challenging case of analyzing errors in the POD ROM with variations in initial data and parameters is left to be explored elsewhere.

We also note that the analysis we perform is focused on a priori error estimates. For a posteriori error estimates extensive work has been conducted within the reduced basis community for parameterized PDEs (see, e.g., [19, 23, 48, 55, 56]). We also mention that a popular approach is to combine the reduced basis method in parameter space with POD in time [14]. Whether or not the results in this paper could be used to improve this approach is not explored within this paper.

DQs in Applications The focus of this paper is on the role played by DQs in the POD numerical analysis. We emphasize, however, that DQs are also widely used in practical applications.

One of the most important uses of DQs in practice is in *hyperreduction* methods for ROMs of nonlinear systems of the form $y' = f(t, y)$. Hyperreduction methods [60] significantly decrease the computational cost of the nonlinear ROM operator evaluations, which can be prohibitive in realistic applications. Popular hyperreduction methods (e.g., the empirical interpolation method (EIM) [4] and its discrete counterpart, the discrete empirical interpolation method (DEIM) [7]) use the nonlinear snapshots $f(t, y)$ to construct accurate approximations of the nonlinear ROM operators. As noted on page 48 in [7], since $f(t, y) = y'$ and $(y^{n+1} - y^n)/\Delta t \approx y'$, using nonlinear snapshots is similar to including the DQs. The DQs' connection to nonlinear snapshots was also used in [8] to develop the solution-based nonlinear subspace (SNS) method as an efficient alternative to classical hyperreduction techniques. The SNS method was used in the reduced order modeling of the nonlinear diffusion equation and the parameterized quasi-1D Euler equation.

The DQs were explicitly used in various practical applications. For example, the DQs were utilized to develop data-driven ROMs for turbulent flows, in which the eddy viscosity field is a function of the time history of the velocity field (see Section 3.3 in [24]). The DQs were also used in the reduced order modeling of the FitzHugh–Nagumo equations, which are used to model the dynamics of a spiking neuron (see Section 4 in [35]). Furthermore, the DQs were employed to construct ROMs for the control of laser surface hardening [26], for feedback control of various PDEs [40], for partial integro-differential equations arising in financial applications [50], for subdiffusion equations [33], for convection-diffusion equations [62], for wave equations [22, 62], and for flow between offset cylinders and lid driven cavity flows [34].

In this paper, we use DQs with respect to time to obtain optimal pointwise in time error estimates. A different, yet related approach was utilized in, e.g., [5, 32, 63], where DQs with respect to system parameters and initial conditions were used to improve the predictive capabilities of reduced basis methods (RBMs) [23, 45] for parameterized problems. In this setting, the noDQ case is referred to as the Lagrange approach, whereas the DQ case is referred to as the Hermite approach [32]. The error sensitivity with respect to parameters

was investigated in, e.g., [27, 46].

The rest of the paper is organized as follows. In Section 2.3, we describe the POD construction in the noDQ and DQ cases.

In Section 2.4, we give more detail about the previously described POD pointwise projection error assumption, show using examples that it can fail in the noDQ case, and prove that it is always satisfied in the DQ case. These results allow us to complete the POD ROM error analysis in Section 2.5. For the first two main contributions, in Section 2.6 we illustrate numerically the theoretical results. Specifically, for the heat equation (2.1) and both analytical examples, we show the following: (i) in the noDQ case, the error scales as in (2.14) (i.e., is suboptimal), and (ii) in the DQ case, the error scales according to the new error bounds. Finally, in Section 2.7, we present our conclusions and future research directions.

2.3 Proper Orthogonal Decomposition (POD)

In this section we introduce two different approaches for constructing our reduced basis by using the *proper orthogonal decomposition (POD)* [25, 57]. Suppose we have a collection of snapshots $U = \{u^n\}_{n=0}^N$ contained in a real Hilbert space $\mathcal{H}$. In the POD numerical analysis, a typical assumption (see, e.g., [36, 57]) is that each snapshot u^n is exactly equal to $u(t_n)$, where $u \in C([0,T];\mathcal{H})$ and $t_n = n\Delta t$ for $n = 0, \ldots, N$ so that $t_0 = 0$, $t_N = T$, and $\Delta t = T/N$. For now, we only assume $T > 0$ is a fixed positive constant and we let $\Delta t = T/N$. We emphasize that T is fixed, but N is allowed to vary.

2.3.1 POD Without Difference Quotients (noDQ Case)

We begin by examining the POD problem without difference quotients. In what follows, we denote this case the *noDQ case*. Given a fixed $r > 0$, the problem is to find a set of orthonormal basis functions $\{\varphi_i\}_{i=1}^r \subset \mathcal{H}$, called POD modes or POD basis functions, that optimally approximate the snapshots in the sense that the following error measure is minimized:

$$E_r = \frac{1}{N+1} \sum_{n=0}^{N} \|u^n - P_r u^n\|_{\mathcal{H}}^2, \tag{2.15}$$

where $P_r : \mathcal{H} \to \mathcal{H}$ is the orthogonal projection onto $X^r = \text{span}\{\varphi_i\}_{i=1}^r$ given by

$$P_r u = \sum_{i=1}^{r} (u, \varphi_i)_{\mathcal{H}} \varphi_i, \quad u \in \mathcal{H}. \tag{2.16}$$

One way to find a solution of this problem is to solve the eigenvalue problem

$$K\mathbf{z}_i = \lambda_i \mathbf{z}_i, \quad \text{for } i = 1, \dots, r, \tag{2.17}$$

where K is the snapshot correlation matrix with entries

$$K_{mn} = \frac{1}{N+1} (u^m, u^n)_{\mathcal{H}}, \quad m, n = 0, \dots, N. \tag{2.18}$$

We order the eigenvalues $\{\lambda_i\}$ and corresponding orthonormal eigenvectors $\{\mathbf{z}_i\}$ so that $\lambda_1 \geq \lambda_2 \geq \lambda_{N+1} \geq 0$. The optimizing orthonormal set $\{\varphi_i\}_{i=1}^r \subset \mathcal{H}$ is given by

$$\varphi_i = \lambda_i^{-1/2} (N+1)^{-1/2} \sum_{m=0}^{N} (\mathbf{z}_i)^m u^m, \quad i = 1, \dots, r, \tag{2.19}$$

where $(\mathbf{z}_i)^m$ is the mth entry of $\mathbf{z}_i$. Using these POD modes gives the optimal value for the approximation error:

$$\frac{1}{N+1} \sum_{n=0}^{N} \|u^n - P_r u^n\|_{\mathcal{H}}^2 = \sum_{i>r} \lambda_i. \tag{2.20}$$

We note that the scaling factor $(N+1)^{-1}$ is important if one is interested in the solution of the optimization problem as more snapshots are collected, i.e., as Δt decreases or N increases. For certain choices of the scaling factor, the error measure E_r in (2.15) converges to a time integral or a constant multiple of a time integral, and the POD eigenvalues and POD modes also converge; see, e.g., [15, 20, 37, 52] for more information.

Different choices for the scaling factor in (2.15) have been used in the literature. We fix the scaling factor throughout this work to be $(N+1)^{-1}$ for simplicity. We note that since $\Delta t = T/N$, we have $(N+1)^{-1} = T_1^{-1} \Delta t$, where $T_1 = T + \Delta t$. Therefore, E_r in (2.15) is equal to the left Riemann sum approximation of the integral

$$\frac{1}{T_1} \int_0^{T_1} \|u(t) - P_r u(t)\|_{\mathcal{H}}^2 \, dt.$$

We note that the results in this work will hold for other scaling factors, as long as the scaling factor in question scales like a constant multiple of Δt.

Remark 2.2. One can also consider variable time steps and weights in the POD problem; we only consider a constant time step and single weight $(N+1)^{-1}$ for simplicity. Furthermore, one can use other quadrature rules, such as the midpoint rule or trapezoid rule, to obtain appropriate weights for the POD problem.

In the following result, we give POD approximation errors in different norms and using other projections onto X^r. Similar results have been proved in multiple works (see, e.g., [30, 31, 41, 51, 53]), and our proof relies on techniques from these works. We note that this result can be obtained directly from the general results in the recent reference [41]; however, we include a proof to be complete. In this work, a bounded linear operator $\Pi : Z \to Z$ for a normed space Z is a projection onto $Z^r \subset Z$ if $\Pi^2 = \Pi$ and the range of Π equals Z^r. In this case, $\Pi z = z$ for any $z \in Z^r$.

Lemma 2.3. *Let $X^r = \text{span}\{\varphi_i\}_{i=1}^r \subset \mathcal{H}$, let $P_r : \mathcal{H} \to \mathcal{H}$ be the orthogonal projection onto X^r as defined in (2.16), and let s be the number of positive POD eigenvalues for $U = \{u^n\}_{n=0}^N$. If W is a real Hilbert space with $U \subset W$ and $R_r : W \to W$ is a bounded linear projection onto X^r, then*

$$\frac{1}{N+1}\sum_{n=0}^{N} \|u^n - P_r u^n\|_W^2 = \sum_{i=r+1}^{s} \lambda_i \|\varphi_i\|_W^2, \tag{2.21}$$

$$\frac{1}{N+1}\sum_{n=0}^{N} \|u^n - R_r u^n\|_W^2 = \sum_{i=r+1}^{s} \lambda_i \|\varphi_i - R_r\varphi_i\|_W^2. \tag{2.22}$$

Proof. First, we note that (2.21) is a special case of (2.22) since $P_r\varphi_i = 0$ for $i > r$. Therefore, we only prove (2.22).

Next, by the POD approximation error formula (2.20), we have $u^n = P_s u^n$ for each n. If $r \geq s$, since R_r is a projection onto X^r we have $R_r u^n = R_r P_s u^n = P_s u^n = u^n$ and this proves the result. Therefore, assume $r < s$. Note by the definition of φ_i in (2.19), since $u^n \in W$ for each n we have $\varphi_i \in W$ for $i = 1, \ldots, r$. Therefore, $X^r \subset W$, and since the range of R_r equals X^r we know the W norm in (2.22) is well-defined.

Now, using the definition of P_r in (2.16) gives

$$\begin{aligned}\frac{1}{N+1}\sum_{n=0}^{N} \|u^n - R_r u^n\|_W^2 &= \frac{1}{N+1}\sum_{n=0}^{N} ((I - R_r)P_s u^n, (I - R_r)P_s u^n)_W \\ &= \frac{1}{N+1}\sum_{n=0}^{N}\sum_{i,j=1}^{s} (u^n, \varphi_j)_{\mathcal{H}} (u^n, \varphi_i)_{\mathcal{H}} ((I - R_r)\varphi_j, (I - R_r)\varphi_i)_W,\end{aligned}$$

where I is the identity operator. Next, take the $\mathcal{H}$ inner product of (2.19) with u^n and use the eigenvalue equations (2.17)-(2.18) to get

$$(u^n, \varphi_i)_{\mathcal{H}} = (N+1)^{1/2} \lambda_i^{1/2} (\mathbf{z}_i)^n.$$

Using this and also that $\{\mathbf{z}_i\}$ is orthonormal so that $\sum_{n=0}^{N} (\mathbf{z}_j)^n (\mathbf{z}_i)^n = \delta_{ij}$ gives

$$\frac{1}{N+1}\sum_{n=0}^{N}\|u^n - R_r u^n\|_W^2 = \sum_{i,j=1}^{s}(\lambda_i\lambda_j)^{1/2}\delta_{ij}((I-R_r)\varphi_j,(I-R_r)\varphi_i)_W$$
$$= \sum_{i=1}^{s}\lambda_i\|(I-R_r)\varphi_i\|_W^2.$$

Since $\varphi_i \in X^r$ for $i = 1,\ldots,r$ and R_r is a projection onto X^r, we have $R_r\varphi_i = \varphi_i$ for $i = 1,\ldots,r$ and this proves the result. □

2.3.2 POD With Difference Quotients (DQ Case)

In this section we consider a POD problem for the same snapshots as those in Section 2.3.1, this time utilizing the difference quotients [36]: find an orthonormal set of basis functions $\{\varphi_i\}_{i=1}^r \subset \mathcal{H}$ minimizing the approximation error

$$E_r^{\mathrm{DQ}} = \frac{1}{2N+1}\sum_{n=0}^{N}\|u^n - P_r u^n\|_{\mathcal{H}}^2 + \frac{1}{2N+1}\sum_{n=1}^{N}\|\partial u^n - P_r\partial u^n\|_{\mathcal{H}}^2, \tag{2.23}$$

where the *difference quotients (DQs)* $\{\partial u^n\}_{i=1}^N$ are defined by

$$\partial u^n = \frac{u^n - u^{n-1}}{\Delta t}. \tag{2.24}$$

In what follows, we denote this case the *DQ case*.

Remark 2.4. One can give different weights to the snapshot and DQ approximation errors by replacing the second scaling factor $1/(2N+1)$ in (2.23) by a weighted fraction $\theta/(2N+1)$, where θ is a positive constant that is independent of N. The main results in this work can be modified to handle this case; however, we consider the unweighted case in (2.23) to simplify the presentation.

The solution to the minimization of the approximation error in (2.23) can be found by setting $v^n = u^n$ for $n = 0,\ldots,N$ and $v^{N+n} = \partial u^n$ for $n = 1,\ldots,N$. This yields a new collection of snapshots $U^{\mathrm{DQ}} = \{v^n\}_{n=0}^M$, where $M = 2N$. Proceeding as outlined in Section 2.3.1 using the new collection $\{v^n\}_{n=0}^M$ in place of $\{u^n\}_{n=0}^N$ gives the solution of this different POD problem. We use $\{\lambda_i^{\mathrm{DQ}}\}$ to denote the POD eigenvalues for this POD problem; we use the same notation $\{\varphi_i\}_{i=1}^r$ for the POD basis functions. The optimal approximation error is given by

$$\frac{1}{2N+1}\sum_{n=0}^{N}\|u^n - P_r u^n\|_{\mathcal{H}}^2 + \frac{1}{2N+1}\sum_{n=1}^{N}\|\partial u^n - P_r \partial u^n\|_{\mathcal{H}}^2 = \sum_{i>r}\lambda_i^{\mathrm{DQ}}. \tag{2.25}$$

Again, the choice of the scaling factor in the approximation error (2.23) is important if we consider the case where the amount of data increases, i.e., a finer time discretization is used so that Δt decreases and N increases. The DQs are used to approximate the time derivative of the data; therefore, for an appropriate choice of the scaling factor the approximation error in (2.23) contains approximations of time integrals involving both the data $u(t)$ and also the time derivative of the data $\partial_t u(t)$. For the DQ case, we use $(2N+1)^{-1}$ for the scaling factor throughout for simplicity.

As before, we give POD approximation errors in different norms and using other projections onto X^r.

Lemma 2.5. *Let $X^r = \operatorname{span}\{\varphi_i\}_{i=1}^r \subset \mathcal{H}$, let $P_r : \mathcal{H} \to \mathcal{H}$ be the orthogonal projection onto X^r as defined in (2.16), and let s be the number of positive POD eigenvalues for the collection $U^{\mathrm{DQ}} = \{v^n\}_{n=0}^{2N}$ described above. If W is a real Hilbert space with $U^{\mathrm{DQ}} \subset W$ and $R_r : W \to W$ is a bounded linear projection onto X^r, then*

$$\frac{1}{2N+1}\left(\sum_{n=0}^{N}\|u^n - P_r u^n\|_W^2 + \sum_{n=1}^{N}\|\partial u^n - P_r \partial u^n\|_W^2\right) = \sum_{i=r+1}^{s}\lambda_i^{\mathrm{DQ}}\|\varphi_i\|_W^2, \tag{2.26}$$

$$\frac{1}{2N+1}\left(\sum_{n=0}^{N}\|u^n - R_r u^n\|_W^2 + \sum_{n=1}^{N}\|\partial u^n - R_r \partial u^n\|_W^2\right) = \sum_{i=r+1}^{s}\lambda_i^{\mathrm{DQ}}\|\varphi_i - R_r\varphi_i\|_W^2. \tag{2.27}$$

Proof. Apply Lemma 2.3 to the new collection of snapshots $\{v^n\}_{n=0}^M$ described above. □

Remark 2.6. In this section, we considered the DQs defined by (2.24). In practice the definition of the DQs will reflect the time discretization used to collect the snapshot data. For example, POD with central difference quotients is used for wave equations in [22, 62] and fractional difference quotients are used for a subdiffusion problem in [33].

It is possible that the results of this paper can be extended to these and other definitions of the DQs, such as those arising from the backward differentiation formulas (BDF2, BDF3, etc.). We leave this to be considered elsewhere.

2.4 Pointwise Projection Error Estimates

In the current literature on pointwise error bounds for the POD of parabolic problems several researchers make an assumption concerning the pointwise in time behavior of the

POD projection errors [10, 11, 12, 21, 30, 31, 34, 43, 59, 61]. Roughly, the assumption says that the POD projection error at any time is of the same order as the total POD projection errors considered in Section 2.3. Next, we formalize this assumption in Assumption 2.7, and then we discuss it for the noDQ case (Section 2.4.1) and the DQ case (Section 2.4.2).

We consider the POD of a collection of snapshots $U := \{u^n\}_{n=0}^N \subset \mathcal{H}$ and also $U \subset W$, as in Section 2.3. Recall, $P_r : \mathcal{H} \to \mathcal{H}$ is the orthogonal projection onto the first r POD modes. For either the noDQ case or the DQ case, the pointwise POD projection error assumption is given as follows:

Assumption 2.7. *There exists a constant C, depending on $T = N\Delta t$ only, such that the POD projection error satisfies*

$$\|u^n - P_r u^n\|_W^2 \leq C \sum_{i=r+1}^{s} \lambda_i \|\varphi_i\|_W^2 \quad \text{for all } r = 1, \ldots, s \text{ and } n = 0, \ldots, N. \tag{2.28}$$

In Section 2.4.1, we construct examples that show that this assumption can be violated in the noDQ case. In Section 2.4.2, we show in Theorem 2.14 that this assumption is always satisfied in the DQ case.

Remark 2.8 (Avoiding Assumption 2.7)**.** We notice that Assumption 2.7 would follow directly from the POD approximation properties (2.21) (in the noDQ case) and (2.26) (in the DQ case) if we dropped the $1/(N+1)$ and $1/(2N+1)$ factors in the definitions (2.15) and (2.23) of the error measures E_r and E_r^{DQ} . In fact, when $\mathcal{H} = W = \mathbb{R}^m$, this approach is used in, e.g., [35]. We emphasize, however, that using this approach would increase by Δt^{-1} the magnitudes of the eigenvalues on the right-hand side of the POD approximation properties (2.21) and (2.26), which would yield suboptimal error estimates. Similar conclusions were reached in Remark 2.3 in [30] for the case $W = \mathcal{H}$.

Remark 2.9 (Similar Assumptions)**.** For $W = \mathcal{H}$, Assumption 2.7 is Assumption 2.1 in [30] (in which the L^2 inner product should be replaced with the correct $\mathcal{H}$ inner product). A similar assumption (but for the L^2 projection of a continuous solution on X^r when $\mathcal{H} = L^2$) is made in Assumption 3.2 in [31]. No such assumption is made in [29], since Theorem 3.5 proves an estimate for the average error, not for the pointwise in time error. Finally, we note that Figure 4 in [30] provided numerical validation for Assumption 2.7 for the particular setting in [30] when $W = \mathcal{H}$.

2.4.1 Pointwise Error Estimates: noDQ Case

First, we note that in general the scaling factor $N + 1$ is the worst case scenario for the failure of Assumption 2.7. To see this, note that for any fixed k we have

$$\|u^k - P_r u^k\|_W^2 = (N+1)\frac{1}{N+1}\|u^k - P_r u^k\|_W^2$$
$$\leq (N+1)\left(\frac{1}{N+1}\sum_{i=0}^{N}\|u^i - P_r u^i\|_W^2\right) \tag{2.29}$$
$$= (N+1)\sum_{i=r+1}^{s}\lambda_i^{noDQ}\|\varphi_i\|_W^2, \tag{2.30}$$

where we used Lemma 2.3 to obtain (2.30). Note that for many collections of snapshots $\{u^k\}_{k=0}^N$ the inequality in (2.29) will be very conservative. Nevertheless, we show below that the above $N+1$ scaling is attained for a family of examples.

Assumption 2.7 says that the error at any particular index is not much larger than the other pointwise errors, or equivalently the inequality (2.29) is overly conservative. Therefore, Assumption 2.7 will be false if there is an index n such that the projection error at index n is much larger than the remaining pointwise errors, i.e.,

$$\|u^n - P_r u^n\|_W^2 \gg \|u^i - P_r u^i\|_W^2, \quad \forall i \neq n, \quad 0 \leq i \leq N. \tag{2.31}$$

Next, we provide a *family of counterexamples* to Assumption 2.7, i.e., a family of exact solutions (data) that yield POD bases that satisfy condition (2.31).

Let $\{\varphi_k\}_{k\geq 1}$ be an orthonormal set in a Hilbert space $\mathcal{H}$, with $\dim(\mathcal{H}) \geq N+1$, and let $\lambda_1 \geq \lambda_2 \geq \cdots > 0$ be any sequence of positive numbers. Suppose the data $U = \{u^n\}_{n=0}^N \subset \mathcal{H}$ is given by

$$u^n = (N+1)^{1/2}\lambda_{n+1}^{1/2}\varphi_{n+1}, \quad n = 0, \ldots, N. \tag{2.32}$$

It can be checked that this data has POD eigenvalues $\{\lambda_k\}$ with corresponding POD modes $\{\varphi_k\}$.

Let W be a real Hilbert space with $U \subset W$. In Proposition 2.10, we show that Assumption 2.7 fails for the data above. Specifically, (2.33) shows that the assumption fails for the specific case of $r = N$ at index N. Furthermore, if the values $\{\lambda_k\}$ decay exponentially fast as in (2.34), then (2.35) shows that the assumption fails for any r at index r.

Proposition 2.10. *Let the data $U = \{u^n\}_{n=0}^N \subset \mathcal{H}$ be given in (2.32) as described above. Then the POD pointwise projection error for u^N is given by*

$$\left\|u^N - P_N u^N\right\|_W^2 = (N+1)\lambda_{N+1}\|\varphi_{N+1}\|_W^2. \tag{2.33}$$

Also, for any fixed r if

$$\lambda_k = \beta \|\varphi_k\|_W^{-2} e^{-\gamma k}, \quad k > r, \tag{2.34}$$

for some positive constants β and γ, then

$$\|u^r - P_r u^r\|_W^2 \geq \frac{\min\{1,\gamma\}}{2} (N+1) \sum_{k=r+1}^{N+1} \lambda_k \|\varphi_k\|_W^2. \tag{2.35}$$

Remark 2.11. Note that for the second part of the result we still assume the POD eigenvalues in (2.34) are ordered so that $\lambda_1 \geq \lambda_2 \geq \cdots > 0$. Depending on the values of $\|\varphi_k\|_W$ and γ, the POD eigenvalues in (2.34) may not be ordered in this way. In such a case, the POD eigenvalues may need to be reordered in order to obtain a similar result. If $W = \mathcal{H}$ or if $\|\varphi_k\|_W$ increases slowly relative to $e^{-\gamma k}$, then the ordering $\lambda_1 \geq \lambda_2 \geq \cdots > 0$ will automatically be satisfied.

Proof. Note that $P_r u^k = 0$ when $k \geq r$ and so

$$\|u^k - P_r u^k\|_W^2 = (N+1)\lambda_{k+1} \|\varphi_{k+1}\|_W^2, \quad k \geq r. \tag{2.36}$$

Thus, (2.33) follows immediately from (2.36) with $k = N$.

Next, to prove (2.35), fix r and assume (2.34) holds. Then (2.36) with $k = r$ gives

$$\|u^r - P_r u^r\|_W^2 = (N+1)\lambda_{r+1} \|\varphi_{r+1}\|_W^2. \tag{2.37}$$

We bound half of the right-hand side of (2.37) from below by a constant multiple of the remaining terms in the sum in (2.35). Note that the assumption (2.34) on the value of λ_{r+1} gives

$$\frac{1}{2}\lambda_{r+1} \|\varphi_{r+1}\|_W^2 = \frac{\beta}{2} e^{-\gamma(r+1)}. \tag{2.38}$$

Next, we note that the exponential term on the right-hand side of (2.38) satisfies the following estimate:

$$\frac{1}{\gamma} e^{-\gamma(r+1)} \geq \frac{1}{\gamma} \left(e^{-\gamma(r+1)} - e^{-\gamma(N+1)} \right) = \int_{r+1}^{N+1} e^{-\gamma x} dx \geq \sum_{k=r+2}^{N+1} e^{-\gamma k}. \tag{2.39}$$

Using (2.34), (2.38), and (2.39), we obtain

$$\frac{1}{2}(N+1)\lambda_{r+1} \|\varphi_{r+1}\|_W^2 \geq \frac{\gamma\beta}{2}(N+1) \sum_{k=r+2}^{N+1} e^{-\gamma k} = \frac{\gamma}{2}(N+1) \sum_{k=r+2}^{N+1} \lambda_k \|\varphi_k\|_W^2. \tag{2.40}$$

Using (2.37) and (2.40), we get

$$\|u^r - P_r u^r\|_W^2 \geq \frac{1}{2}(N+1)\lambda_{r+1}\|\varphi_{r+1}\|_W^2 + \frac{\gamma}{2}(N+1)\sum_{k=r+2}^{N+1}\lambda_k\|\varphi_k\|_W^2$$
$$\geq \frac{\min\{1,\gamma\}}{2}(N+1)\sum_{k=r+1}^{N+1}\lambda_k\|\varphi_k\|_W^2, \tag{2.41}$$

which proves (2.35). □

Proposition 2.10 yields a family of counterexamples to Assumption 2.7. Next, we consider two counterexamples that we investigate numerically in Section 2.6.

Counterexample 1

To construct the first counterexample to Assumption 2.7 (which we denote counterexample 1), we follow the theoretical setting in this section and construct a family of ROM basis functions that satisfy equation (2.32). Specifically, we consider an orthonormal set $\{\varphi_n\}_{n=0}^N$ in $\mathcal{H} = L^2(0,1)$ given by

$$\varphi_{n+1}(x) := 2^{1/2}\,\sin((k\,t_n+1)\pi\,x)\,, \tag{2.42}$$

where k is a positive integer, $x \in [0,1]$, and $t_n = n\,\Delta t$ is chosen such that $k\,t_n \in \mathbb{N}$, $\forall\, n \in \mathbb{N}$.

Next, we choose the eigenvalues

$$\lambda_1 = \lambda_2 = \cdots = \lambda_{N+1} = \frac{1}{2(N+1)}\,, \tag{2.43}$$

which satisfy $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{N+1} > 0$. Finally, choosing the analytical solution

$$u_{\text{counterexample 1}}(x,t) = \sin((k\,t+1)\pi\,x) \tag{2.44}$$

and the corresponding forcing term

$$f = (k\,\pi\,x)\cos((k\,t+1)\pi\,x) + \nu\pi^2(kt+1)^2\sin((k\,t+1)\pi\,x) \tag{2.45}$$

yield the data $U = \{u^n\}_{n=0}^N$ that satisfies equation (2.32). In Section 2.6, we investigate numerically counterexample 1 given by the analytical solution (2.44).

Remark 2.12. Equation (2.32) (see also the comment below Assumption A.1 in [43]) shows that the ROM basis functions are scaled versions of the snapshots. For counterexample 1, this scaling is illustrated in (2.42) and (2.44).

Counterexample 2

To construct the second counterexample to Assumption 2.7 (which we denote counterexample 2), we construct a family of ROM basis functions that satisfy both equation (2.32) and equation (2.34) in Proposition 2.10. Specifically, we consider the same orthonormal set $\{\varphi_n\}_{n=0}^{N}$ in $\mathcal{H} = L^2(0,1)$ given in (2.42) above, where again k is a positive integer, $x \in [0,1]$, and $t_n = n\,\Delta t$ is chosen such that $k\,t_n \in \mathbb{N}$, $\forall\, n \in \mathbb{N}$. Next, for positive constants α, δ, and ρ, with $\delta = \rho\Delta t$, we choose exponentially decaying eigenvalues as in (2.34):

$$\begin{aligned}\lambda_{n+1} &= \beta e^{-\gamma(n+1)},\\ \beta &= \frac{1}{4\delta(N+1)} e^{-\alpha+\alpha\delta^{-1}\Delta t} = \frac{1}{4\rho T_1} e^{-\alpha+\alpha\rho^{-1}},\\ \gamma &= \alpha\delta^{-1}\Delta t = \alpha\rho^{-1},\end{aligned}$$

where $T_1 = T + \Delta t$. Finally, it can be checked that choosing the analytical solution

$$u_{\text{counterexample 2}}(x,t) = \frac{1}{\sqrt{2\delta}} \left(e^{-\alpha(1+t/\delta)}\right)^{1/2} \sin((kt+1)\pi x) \tag{2.46}$$

and the corresponding forcing term

$$\begin{aligned} f = \frac{1}{\sqrt{2\delta}} \left(e^{-\alpha(1+t/\delta)}\right)^{1/2} \Big[&\frac{-\alpha}{2\delta} \sin((kt+1)\pi x) + (k\pi x)\cos((kt+1)\pi x)\\ &+ \nu\pi^2 (kt+1)^2 \sin((kt+1)\pi x)\Big] \end{aligned} \tag{2.47}$$

yield the data $U = \{u^n\}_{n=0}^{N}$ that satisfies equation (2.32), which shows that, in counterexample 2, the ROM basis functions are scaled versions of the snapshots. In Section 2.6, we investigate numerically counterexample 2 given by the analytical solution (2.46).

2.4.2 POD Pointwise Error Estimates: DQ Case

We now give one of the main results of this paper. In Theorem 2.14, we show that Assumption 2.7 is always satisfied in the DQ case. This will allow us to prove in Section 2.5 optimal pointwise in time ROM error bounds in the DQ case. In particular, Theorem 2.14 will show that the assumptions similar to Assumption 2.7 that have been made in, e.g., [30], are unnecessary for obtaining optimal error bounds in the DQ case.

In continuous time, it is well-known that the magnitude of a function $z \in H^1(0,T)$ at any point in time is bounded above by a constant multiple of the $H^1(0,T)$ norm of z. The constant in the bound only depends on T, and there is also a similar inequality that holds for functions taking values in a Banach space Z (see, e.g., [13, Section 5.9.2, page 302, Theorem 2 (iii)]). Below, we establish a discrete time analogue of this Sobolev embedding

$H^1(0,T;Z) \hookrightarrow C([0,T];Z)$, where the DQs replace the time derivative in the $H^1(0,T;Z)$ norm. This lemma will allow us to directly establish POD pointwise projection error bounds in Theorem 2.14, which shows that Assumption 2.7 is automatically satisfied in the DQ case.

Lemma 2.13 (Discrete time Sobolev inequality)**.** *Let $T > 0$, Z be a normed space, $\{z^n\}_{n=0}^N \subset Z$, and $\Delta t = T/N$. Then*

$$\max_{0\le k\le N} \|z^k\|_Z^2 \le C\left(\frac{1}{2N+1}\sum_{n=0}^{N}\|z^n\|_Z^2 + \frac{1}{2N+1}\sum_{n=1}^{N}\|\partial z^n\|_Z^2\right),$$

where $C = 6\max\{1,T^2\}$ and $\partial z^n = (z^n - z^{n-1})/\Delta t$ for $n = 1,\dots,N$.

Proof. For each k,ℓ with $N \ge k > \ell \ge 0$, we have $z^k - z^\ell = \Delta t\sum_{n=\ell+1}^{k}\partial z^n$. This gives

$$\|z^k\|_Z \le \|z^\ell\|_Z + \sum_{n=1}^{N}\Delta t^{1/2}(\Delta t^{1/2}\|\partial z^n\|_Z) \le \|z^\ell\|_Z + T^{1/2}\left(\sum_{n=1}^{N}\Delta t\|\partial z^n\|_Z^2\right)^{1/2}, \tag{2.48}$$

where we used $\sum_{n=1}^{N}\Delta t = N\Delta t = T$. This inequality is also clearly true for $k = \ell$, and a similar argument shows that this inequality also holds for $0 \le k < \ell \le N$.

Now we choose ℓ so that

$$\|z^\ell\|_Z = \min_{0\le n\le N}\|z^n\|_Z. \tag{2.49}$$

We know such an ℓ must exist since N is finite. Then

$$\begin{aligned}\|z^\ell\|_Z &= \frac{1}{N+1}(N+1)\|z^\ell\|_Z = \frac{1}{N+1}\sum_{n=0}^{N}\|z^\ell\|_Z\\ &\le \frac{1}{T}\sum_{n=0}^{N}\Delta t\|z^n\|_Z \le T^{-1/2}\left(\sum_{n=0}^{N}\Delta t\|z^n\|_Z^2\right)^{1/2},\end{aligned}$$

where we used (2.49), $1/(N+1) < 1/N = T^{-1}\Delta t$, $\sum_{n=1}^{N}\Delta t = N\Delta t = T$, and the Cauchy-Schwarz inequality. Using this inequality with (2.48) yields

$$\|z^k\|_Z \le T^{-1/2}\left(\sum_{n=0}^{N}\Delta t\|z^n\|_Z^2\right)^{1/2} + T^{1/2}\left(\sum_{n=1}^{N}\Delta t\|\partial z^n\|_Z^2\right)^{1/2}. \tag{2.50}$$

Squaring both sides, and using the inequalities $(a+b)^2 \le 2(a^2+b^2)$ and $\Delta t = (2T + \Delta t)/(2N+1) \le 3T/(2N+1)$, we obtain the result. □

Theorem 2.14. *Let $X^r = \operatorname{span}\{\varphi_i\}_{i=1}^r \subset \mathcal{H}$, let $P_r : \mathcal{H} \to \mathcal{H}$ be the orthogonal projection onto X^r as defined in (2.16), and let s be the number of positive POD eigenvalues for U^{DQ}. If W is a real Hilbert space with $U^{\mathrm{DQ}} \subset W$ and $R_r : W \to W$ is a bounded linear projection onto X^r, then*

$$\max_{0\le k\le N} \left\|u^k - P_r u^k\right\|_{\mathcal{H}}^2 \le C \sum_{i=r+1}^{s} \lambda_i^{\mathrm{DQ}}, \tag{2.51a}$$

$$\max_{0\le k\le N} \left\|u^k - P_r u^k\right\|_{W}^2 \le C \sum_{i=r+1}^{s} \lambda_i^{\mathrm{DQ}} \|\varphi_i\|_W^2, \tag{2.51b}$$

$$\max_{0\le k\le N} \left\|u^k - R_r u^k\right\|_{W}^2 \le C \sum_{i=r+1}^{s} \lambda_i^{\mathrm{DQ}} \|\varphi_i - R_r\varphi_i\|_W^2, \tag{2.51c}$$

where $C = 6\max\{1, T^2\}$.

Proof. First, note that (2.51a) follows from (2.51b) with $W = \mathcal{H}$ since $\|\varphi_i\|_{\mathcal{H}} = 1$ for all i. Also, (2.51b) follows from (2.51c) since $P_r\varphi_i = 0$ for $i > r$. Therefore, we only prove (2.51c).

Set $Z = W$ and $z^n = u^n - R_r u^n$ for each n. Using Lemma 2.13, $\partial z^n = \partial u^n - R_r \partial u^n$ for each n, and Lemma 2.5 gives the result. □

2.5 Pointwise Error Estimates: DQ Case

In this section, we prove pointwise in time error estimates for the heat equation and discuss the time and ROM discretization optimality of these estimates. In Section 2.5.1, we prove the pointwise in time error estimates using Crank-Nicolson time stepping in the DQ case (see Section 2.3.2). In Section 2.5.2, we consider three definitions of optimality for the ROM discretization error and classify the optimality types of each pointwise error estimate in Section 2.5.1. We show that all of the error estimates are optimal in some sense; although, in some cases we need to assume various POD projection uniform boundedness conditions are satisfied. We also briefly discuss error estimates and optimality for the noDQ case; see Remarks 2.16, 2.18, and 2.23. Below, we consider the DQ case unless explicitly mentioned otherwise.

We begin by establishing notation, definitions, and giving preliminary results that will be used in the ensuing analysis. We let $\Omega \in \mathbb{R}^d, d = 2, 3$ be a regular open domain with Lipschitz continuous boundary Ω and denote by $(\cdot,\cdot)_{L^2}$ and $\|\cdot\|_{L^2}$ the L^2 inner product and norm respectively. We define the function space $X = H_0^1(\Omega)$ as:

$$X := H_0^1(\Omega)^d = \{v \in H^1(\Omega)^d : v|_\Gamma = 0\}.$$

With the inner product $(u,v)_{H_0^1} = (\nabla u, \nabla v)_{L^2}$, the space $X = H_0^1(\Omega)$ is a Hilbert space.

For simplicity, we will only consider the heat equation (2.1). We take $u(\cdot,t) \in X$, $t \in [0,T]$ to be the weak solution of the weak formulation of the heat equation with homogeneous Dirichlet boundary conditions:

$$(\partial_t u, v)_{L^2} + \nu(\nabla u, \nabla v)_{L^2} = (f,v)_{L^2} \quad \forall v \in X. \tag{2.52}$$

Replacing the unknown u with u_r in the heat equation (2.52), using the Galerkin method, projecting the resulting equations onto a space $X^r \subset X$, and discretizing in time using Crank-Nicolson (CN), one obtains the standard CN POD-G-ROM for the heat equation:

$$(\partial u_r^{n+1}, v_r)_{L^2} + \nu(\nabla u_r^{n+1/2}, \nabla v_r)_{L^2} = (f^{n+1/2}, v_r)_{L^2} \quad \forall v_r \in X^r, \tag{2.53}$$

where $\partial u_r^{n+1} = (u_r^{n+1} - u_r^n)/\Delta t$. Also, here and below we use the notation $z^{n+1/2}$ for any discrete or continuous time function z to denote the average

$$z^{n+1/2} := \frac{1}{2}\left(z^{n+1} + z^n\right).$$

Note that, for continuous time functions, we do *not* use $z^{n+1/2}$ to denote $z(t_n + \Delta t/2)$.

Remark 2.15. An alternative CN approach to the time discretization is to replace $f^{n+1/2}$ in (2.53) with $f(t_n + \Delta t/2)$. The results in this section also hold for this case.

We now prove error estimates for the error $u^{n+1} - u_r^{n+1}$, where $u^{n+1} := u(t_{n+1})$ is the solution of the weak formulation of the heat equation (2.52), and u_r^{n+1} is the solution of the CN POD-G-ROM (2.53). For clarity of presentation, we only consider the error components corresponding to the POD truncation and time discretization, i.e., we ignore the spatial discretiztion (e.g., FE) error.

We start by noting that the weak solution of the heat equation evaluated at time $t = t_n + \Delta t/2$ satisfies:

$$\left(\partial u^{n+1}, v_r\right)_{L^2} + \nu(\nabla u^{n+1/2}, \nabla v_r)_{L^2} = (f^{n+1/2}, v_r)_{L^2} + \tilde{A}u_n(v_r) \quad \forall v_r \in X^r, \tag{2.54}$$

where $\partial u^{n+1} = (u^{n+1} - u^n)/\Delta t$ and, after integrating by parts, the consistency error is given by

$$\begin{aligned} \tilde{A}u_n(v) := &\left(\partial u^{n+1} - \partial_t u(t_n + \Delta t/2), v\right)_{L^2} + \nu\left(\Delta(u(t_n + \Delta t/2) - u^{n+1/2}), v\right)_{L^2} \\ &+ \left(f(t_n + \Delta t/2) - f^{n+1/2}, v\right)_{L^2}. \end{aligned} \tag{2.55}$$

We assume that the solution u and the forcing f are smooth enough so that $\tilde{A}u_n(v)$ is well defined for any $v \in X$. We provide a more precise regularity assumption below.

The error is split into two parts:

$$e^{n+1} = u^{n+1} - u_r^{n+1} = (u^{n+1} - w_r^{n+1}) - (u_r^{n+1} - w_r^{n+1}) = \eta^{n+1} - \phi_r^{n+1}, \tag{2.56}$$

where w_r^{n+1} is a proper projection of u^{n+1} on X^r, $\eta^{n+1} := u^{n+1} - w_r^{n+1}$, and $\phi_r^{n+1} = u_r^{n+1} - w_r^{n+1}$. Subtracting (2.53) from (2.54) then yields:

$$\begin{aligned}(\partial\phi_r^{n+1}, v_r)_{L^2} + \nu(\nabla\phi_r^{n+1/2}, \nabla v_r)_{L^2} = (\partial\eta^{n+1}, v_r)_{L^2} + \nu(\nabla\eta^{n+1/2}, \nabla v_r)_{L^2} \\ - \tilde{A}u_n\,(v_r) \quad \forall v_r \in X^r.\end{aligned} \tag{2.57}$$

The standard approach used to prove error estimates in this case is to use the Ritz projection [3, 29, 31, 36, 37, 49]. This is also the standard approach in the FE context [17, 39, 54, 58].

Thus, for the ensuing analysis we choose $w_r := R_r(u)$ in (2.56), where $R_r(u)$ is the Ritz projection of u on X^r:

$$(\nabla(u - R_r(u)), \nabla v_r)_{L^2} = 0 \quad \forall v_r \in X^r. \tag{2.58}$$

We will then denote $\eta_{Ritz} := u - R_r(u)$. Using the Ritz projection, (2.57) then becomes:

$$(\partial\phi_r^{n+1}, v_r)_{L^2} + \nu(\nabla\phi_r^{n+1/2}, \nabla v_r)_{L^2} = (\partial\eta_{Ritz}^{n+1}, v_r)_{L^2} - \tilde{A}u_n\,(v_r) \quad \forall v_r \in X^r, \tag{2.59}$$

where we have used the fact that $(\nabla\eta_{Ritz}^{n+1/2}, \nabla v_r)_{L^2} = 0$ by (2.58).

Remark 2.16. In the noDQ case (see Section 2.3.1), a different approach is typically used to prove error estimates; see, e.g., [6, 30, 31, 53]. Instead of the Ritz projection, in the noDQ case we use the L^2 projection $\Pi_r^{L^2}$ and take $w_r^{n+1} = \Pi_r^{L^2} u^{n+1}$. The term $\nu(\nabla\eta^{n+1/2}, \nabla v_r)_{L^2}$ in (2.57) no longer vanishes; instead, the DQ projection error term is eliminated, i.e., $(\partial\eta^{n+1}, v_r)_{L^2} = 0$ in (2.57). However, as explained in Remark 2.18, the resulting pointwise error estimates are suboptimal.

For the POD basis construction, we must specify a Hilbert space $\mathcal{H}$. For this problem, two natural Hilbert spaces that are often used are $\mathcal{H} = L^2(\Omega)$ or $\mathcal{H} = X = H_0^1(\Omega)$. Let X^r be the span of the first r POD modes for the data set containing the snapshots $\{u^n\}_{n=0}^N$ and the snapshot DQs $\{\partial u^n\}_{n=1}^N$. We can use Lemma 2.5 and Theorem 2.14 to obtain POD approximation error results with either $W = L^2(\Omega)$ or $W = H_0^1(\Omega)$. We note that in the case $\mathcal{H} = H_0^1(\Omega)$, the standard orthogonal POD projection P_r is exactly equal to the Ritz projection R_r.

We emphasize that, in this section, we use the exact solution of the heat equation for the POD basis construction in order to focus on the POD and time discretization errors. Exact solution data was used in this way by Kunisch and Volkwein in their original POD numerical analysis work [36], and also by many other researchers in subsequent works.

2.5.1 Error estimates

We give multiple error bounds for the solution when both the L^2 and H_0^1 POD bases are used. Specifically, we first provide a pointwise in time error bound for the L^2 norm of the solution, and an error bound for the solution norm (a discrete time analogue of the $L^2(0,T;H_0^1(\Omega))$ norm) that includes the L^2 norm of the solution at the final time step. Then, we prove a pointwise in time error bound for the H_0^1 norm of the solution.

We assume the solution u of the heat equation (2.1) and the forcing f satisfy the regularity condition

$$u_{ttt},\ \Delta u_{tt},\ f_{tt} \in L^2(0,T;L^2(\Omega)). \tag{2.60}$$

We also define the regularity constants

$$\begin{aligned} I_{n,1}(u,f) &:= \|u_{ttt}\|_{L^2(t_n,t_{n+1};L^2)} + \|\Delta u_{tt}\|_{L^2(t_n,t_{n+1};L^2)} + \|f_{tt}\|_{L^2(t_n,t_{n+1};L^2)}, \\ I_n(u,f) &:= \|u_{ttt}\|^2_{L^2(t_n,t_{n+1};L^2)} + \|\Delta u_{tt}\|^2_{L^2(t_n,t_{n+1};L^2)} + \|f_{tt}\|^2_{L^2(t_n,t_{n+1};L^2)}, \\ I(u,f) &:= \|u_{ttt}\|^2_{L^2(0,T;L^2)} + \|\Delta u_{tt}\|^2_{L^2(0,T;L^2)} + \|f_{tt}\|^2_{L^2(0,T;L^2)}. \end{aligned} \tag{2.61}$$

As mentioned above, for all of the results below we assume X^r is the span of the first r POD modes for the data set containing the snapshots $\{u^n\}_{n=0}^N$ of the exact solution and the snapshot DQs $\{\partial u^n\}_{n=1}^N$. Furthermore, as pointed out in the introduction, we prove POD error bounds when the parameter ν and the initial data are the same as those used to generate the POD basis.

Lemma 2.17. *Consider the CN POD-G-ROM scheme (2.53). If (2.60) is satisfied, then the following error bounds hold when the L^2 POD basis is used:*

$$\max_{1\le k\le N}\|e^k\|_{L^2}^2 \le C\left(\sum_{i=r+1}^{s}\lambda_i^{DQ}\|\varphi_i - R_r(\varphi_i)\|_{L^2}^2 + \|\phi_r^0\|_{L^2}^2 + \Delta t^4 I(u,f)\right), \tag{2.62}$$

$$\begin{aligned} \|e^N\|_{L^2}^2 + \Delta t\sum_{n=0}^{N-1}\|\nabla e^{n+1/2}\|_{L^2}^2 \le C\Bigg(&\sum_{i=r+1}^{s}\lambda_i^{DQ}(\|\varphi_i - R_r(\varphi_i)\|_{L^2}^2 \\ &+ \|\nabla(\varphi_i - R_r(\varphi_i))\|_{L^2}^2) + \|\phi_r^0\|_{L^2}^2 + \Delta t^4 I(u,f)\Bigg), \end{aligned} \tag{2.63}$$

and the following error bounds hold when the H_0^1 POD basis is used

$$\max_{1\le k\le N}\|e^k\|_{L^2}^2 \le C\left(\sum_{i=r+1}^{s}\lambda_i^{DQ}\|\varphi_i\|_{L^2}^2 + \|\phi_r^0\|_{L^2}^2 + \Delta t^4 I(u,f)\right), \tag{2.64}$$

$$\|e^N\|_{L^2}^2 + \Delta t \sum_{n=0}^{N-1} \|\nabla e^{n+1/2}\|_{L^2}^2 \leq C \left(\sum_{i=r+1}^{s} (1 + \|\varphi_i\|_{L^2}^2)\lambda_i^{DQ} + \|\phi_r^0\|_{L^2}^2 + \Delta t^4 I(u,f) \right). \quad (2.65)$$

Proof. We use $2\Delta t\,(\partial\phi_r^{n+1}, \phi_r^{n+1/2})_{L^2} = \|\phi_r^{n+1}\|_{L^2}^2 - \|\phi_r^n\|_{L^2}^2$ and let $v_r := \phi_r^{n+1/2}$ in equation (2.59) and to obtain

$$\begin{aligned} \|\phi_r^{n+1}\|_{L^2}^2 - \|\phi_r^n\|_{L^2}^2 + 2\nu\Delta t\|\nabla\phi_r^{n+1/2}\|_{L^2}^2 = 2\Delta t[(\partial\eta_{Ritz}^{n+1}, \phi_r^{n+1/2})_{L^2} \\ - \tau_n(\phi_r^{n+1/2})]. \end{aligned} \quad (2.66)$$

Next, use the Cauchy-Schwarz inequality, the Poincaré inequality $\|\phi_r^{n+1/2}\|_{L^2} \leq C\|\nabla\phi_r^{n+1/2}\|_{L^2}$, and Taylor's theorem[†] to bound the right-hand side and obtain

$$\begin{aligned} \|\phi_r^{n+1}\|_{L^2}^2 - \|\phi_r^n\|_{L^2}^2 + 2\nu\Delta t\|\nabla\phi_r^{n+1/2}\|_{L^2}^2 \\ \leq C\Delta t\big(\|\partial\eta_{Ritz}^{n+1}\|_{L^2} + \Delta t^{3/2} I_{n,1}(u,f)\big)\|\nabla\phi_r^{n+1/2}\|_{L^2}. \end{aligned} \quad (2.67)$$

Applying Young's inequality and using $(a+b+c)^2 \leq 3(a^2+b^2+c^2)$ yields

$$\begin{aligned} \|\phi_r^{n+1}\|_{L^2}^2 - \|\phi_r^n\|_{L^2}^2 + 2\nu\Delta t\,\|\nabla\phi_r^{n+1/2}\|_{L^2}^2 \leq \Big(C\Delta t\,\big\|\partial\eta_{Ritz}^{n+1}\big\|_{L^2}^2 + C\Delta t^4 I_n(u,f) \\ + \nu\Delta t\,\|\nabla\phi_r^{n+1/2}\|_{L^2}^2 \Big). \end{aligned} \quad (2.68)$$

Now, summing from $n = 0$ to $k-1$ gives

$$\|\phi_r^k\|_{L^2}^2 + \nu \sum_{n=0}^{k-1} \Delta t\,\|\nabla\phi_r^{n+1/2}\|_{L^2}^2 \leq C\Big(\sum_{n=0}^{k-1} \Delta t\,\big\|\partial\eta_{Ritz}^{n+1}\big\|_{L^2}^2 + \Delta t^4 I(u,f) + \|\phi_r^0\|_{L^2}^2 \Big). \quad (2.69)$$

By the triangle inequality we have $\|e^k\|_{L^2}^2 \leq 2(\|\eta_{Ritz}^k\|_{L^2}^2 + \|\phi_r^k\|_{L^2}^2)$. Applying this inequality, rearranging terms, dropping an unnecessary term, and taking a maximum among constants it then follows from (2.69) that

$$\|e^k\|_{L^2}^2 \leq C\Big(\Delta t \sum_{n=1}^{N} \|\partial\eta_{Ritz}^n\|_{L^2}^2 + \|\eta_{Ritz}^k\|_{L^2}^2 + \|\phi_r^0\|_{L^2}^2 + \Delta t^4 I(u,f) \Big). \quad (2.70)$$

The pointwise in time estimates (2.62) and (2.64) then follow from applying Lemma 2.5 and Theorem 2.14 with $W = L^2(\Omega)$ and $\eta_{Ritz}^k = u^k - R_r u^k$, where R_r is the Ritz projection (which also equals P_r for the H_0^1 POD basis), and using $\Delta t(2N+1) = (2+1/N)T \leq 3T$

The error bounds (2.63) and (2.65) in the solution norm follow by taking $k = N$ in (2.69) and proceeding similarly. □

[†]for more details, see, e.g., [39, Lemma 26, page 166] or [54, pages 16-17]

Remark 2.18. We briefly provide one pointwise in time error estimate for the noDQ case with the L^2 POD basis; other pointwise estimates can be obtained using similar ideas. In the noDQ case, to obtain a pointwise in time L^2 error estimate one can proceed in a similar fashion to the above proof using the L^2 projection instead of the Ritz projection, as discussed in Remark 2.16. The error estimate (2.71) can be obtained using Lemma 2.3 with $\mathcal{H} = L^2(\Omega)$ and $W = H_0^1(\Omega)$, and the worst case pointwise projection error bound (2.30):

$$\max_{1\leq k\leq N}\|e^k\|_{L^2}^2 \leq C\Bigg((N+1)\sum_{i=r+1}^{N+1}\lambda_i^{noDQ} + \sum_{i=r+1}^{N+1}\lambda_i^{noDQ}\|\nabla\varphi_i\|_{L^2}^2 + \|\phi_r^0\|_{L^2}^2 + \Delta t^4 I(u,f)\Bigg). \tag{2.71}$$

If Assumption 2.7 is satisfied, then the $(N+1)$ scaling factor can be removed.

We emphasize that the error estimate (2.71) is suboptimal; see Remark 2.23 below for precise optimality definitions. First, the estimate is suboptimal with respect to the time discretization error because of the extra factor $(N+1) = (T\Delta t^{-1}+1)$. Second, the estimate is suboptimal with respect to the ROM projection error because of the second term on the right-hand side, which contains $\|\nabla\varphi_i\|_{L^2}^2$ instead of $\|\varphi_i\|_{L^2}^2$. This is a consequence of using the L^2 projection instead of the classical Ritz projection (see Remark 2.16). As explained in [30], using the L^2 projection eliminates the need to use the DQs, but yields suboptimal estimates with respect to the ROM projection error. Thus, even if Assumption 2.7 is satisfied and the $(N+1)$ scaling factor can be removed, the error estimate (2.71) is still suboptimal. If the H_0^1 POD basis is used instead, the resulting error estimate is also suboptimal, even if Assumption 2.7 is satisfied; the details are similar.

Next, we prove a pointwise in time error bound in the H_0^1 norm.

Lemma 2.19. *Consider the CN POD-G-ROM scheme (2.53). If (2.60) is satisfied, then the following error bound holds when the L^2 POD basis is used*

$$\max_{1\leq k\leq N}\|\nabla e^k\|_{L^2}^2 \leq C\Bigg(\sum_{i=r+1}^{s}\lambda_i^{DQ}\left(\|\varphi_i - R_r(\varphi_i)\|_{L^2}^2 + \|\nabla(\varphi_i - R_r(\varphi_i))\|_{L^2}^2\right) + \|\nabla\phi_r^0\|_{L^2}^2 + \Delta t^4 I(u,f)\Bigg), \tag{2.72}$$

and the following error bound holds when the H_0^1 POD basis is used

$$\max_{1\leq k\leq N}\|\nabla e^k\|_{L^2}^2 \leq C\left(\sum_{i=r+1}^{s}\lambda_i^{DQ}(1+\|\varphi_i\|_{L^2}^2) + \|\nabla\phi_r^0\|_{L^2}^2 + \Delta t^4 I(u,f)\right). \tag{2.73}$$

Proof. We let $v_r := \partial\phi_r^{n+1}$ in (2.59):

$$\|\partial\phi_r^{n+1}\|_{L^2}^2 + \frac{\nu}{2\Delta t}(\|\nabla\phi_r^{n+1}\|_{L^2}^2 - \|\nabla\phi_r^n\|_{L^2}^2) = (\partial\eta_{Ritz}^{n+1}, \partial\phi_r^{n+1})_{L^2} - \tilde{A}u_n(\partial\phi_r^{n+1}). \tag{2.74}$$

Applying Cauchy-Schwarz and Young's inequalities along with Taylor's theorem on the RHS of (2.74), we get:

$$\begin{aligned}\nu(\|\nabla\phi_r^{n+1}\|_{L^2}^2 - \|\nabla\phi_r^n\|_{L^2}^2) + 2\Delta t\,\|\partial\phi_r^{n+1}\|_{L^2}^2 \leq &\Delta t\,\|\partial\eta_{Ritz}^{n+1}\|_{L^2}^2 + \frac{3}{2}\Delta t\,\|\partial\phi_r^{n+1}\|_{L^2}^2 \\ &+ C\Delta t^4 I_n(u,f).\end{aligned} \tag{2.75}$$

Next, sum from $n=0$ to $n=k-1$ and drop an unnecessary term:

$$\|\nabla\phi_r^k\|_{L^2}^2 \leq \frac{1}{\nu}\sum_{n=0}^{N-1}\Delta t\,\|\partial\eta_{Ritz}^{n+1}\|_{L^2}^2 + C\Delta t^4 I(u,f) + \|\nabla\phi_r^0\|_{L^2}^2.$$

Now use $\|\nabla e^k\|_{L^2}^2 \leq 2(\|\nabla\eta_{Ritz}^k\|_{L^2}^2 + \|\nabla\phi_r^k\|_{L^2}^2)$ to obtain

$$\|\nabla e^k\|_{L^2}^2 \leq C\left(\sum_{n=0}^{N-1}\Delta t\,\|\partial\eta_{Ritz}^{n+1}\|_{L^2}^2 + \|\nabla\eta_{Ritz}^k\|_{L^2}^2 + \Delta t^4 I(u,f) + \|\nabla\phi_r^0\|_{L^2}^2\right).$$

We use Lemma 2.5, Theorem 2.14, and $\Delta t(2N+1) = (2+1/N)T \leq 3T$ to complete the proof. □

2.5.2 Optimality of Pointwise ROM Discretization Errors

Next, we discuss three different definitions of optimality for pointwise in time ROM discretization errors. Again, we assume we are in the DQ case throughout; although we do briefly discuss the noDQ case in Remark 2.23 below. We classify the optimality type of each pointwise in time error bound for the DQ case from Section 2.5.1.

The optimality type of a pointwise error bound depends on both the space $\mathcal{H}$ for the POD basis and the space W for the pointwise error norm. In Section 2.5.1 we considered four possibilities: we used $\mathcal{H} = L^2$ or $\mathcal{H} = H_0^1$ for the POD basis, and we used $W = L^2$ or $W = H_0^1$ for the error norm. Below, we let $\mathcal{H}$ and W be any real Hilbert spaces, we consider the DQ case, and we let $e^k = u^k - u_r^k$ be the ROM error for $k = 0, \ldots, N$. For the discretization, we assume that, if certain conditions are satisfied, then there exists a constant C so that the following pointwise error bound holds:

$$\max_{1\leq k\leq N}\|e^k\|_W^2 \leq C\left(\Lambda_r + \Lambda_r^0 + \zeta(\Delta t) + \xi(h)\right), \tag{2.76}$$

where

- Λ_r is the ROM discretization error, and depends only on r, the POD eigenvalues, and the POD modes;

- Λ_r^0 is the ROM discretization error for the initial condition only, and depends only on r, the POD eigenvalues, and the POD modes;
- $\zeta(\Delta t)$ is an *optimal* time discretization error; and
- $\xi(h)$ is an *optimal* spatial discretization error.

We automatically consider the discretization error suboptimal if either the time or space discretization errors are suboptimal; therefore, we assume those errors are optimal here and focus on the ROM discretization error.

Let $X^r \subset \mathcal{H}$ be the span of the first r POD modes, and assume X^r is also contained in W. Let $P_r : \mathcal{H} \to \mathcal{H}$ be the orthogonal POD projection onto X^r, and let $\Pi_r^W : W \to W$ be the W-orthogonal projection onto X^r. Also, let s be the number of positive POD eigenvalues.

Definition 2.20. We say the ROM discretization error Λ_r is

- **truly optimal** if there exists a constant C such that

$$\Lambda_r \leq C\Lambda_r^\star, \quad \Lambda_r^\star := \max_{1\leq k\leq N} \|u^k - \Pi_r^W u^k\|_W^2, \tag{2.77}$$

- **optimal-I** if there exists a constant C such that

$$\Lambda_r \leq C\Lambda_r^I, \quad \Lambda_r^I := \sum_{i=r+1}^{s} \lambda_i \|\varphi_i\|_W^2, \tag{2.78}$$

- **optimal-II** if there exists a constant C such that

$$\Lambda_r \leq C\Lambda_r^{II}, \quad \Lambda_r^{II} := \sum_{i=r+1}^{s} \lambda_i \|\varphi_i - \Pi_r^W \varphi_i\|_W^2. \tag{2.79}$$

The constant C above should be independent of all discretization parameters, but may depend on the solution data and the problem data.

We note that the first two notions of optimality above are generalizations of definitions discussed in [30], while we believe the optimal-II definition is new. We discuss each type of optimality below.

Remark 2.21. Note that we do not consider the ROM discretization error for the initial condition, Λ_r^0, in these optimality definitions. These definitions can be modified to include the ROM initial condition error, if desired.

Truly optimal: Since Π_r^W is the W-orthogonal projection, the quantity $\Lambda_r^\star$ defined in (2.77) is the best possible pointwise POD data approximation error. As discussed in [30], this is the most natural definition of optimality; however, it may not be straightforward to evaluate the quantity $\Lambda_r^\star$ and compare it to the ROM discretization error bound Λ_r.

Optimal-I (Optimal type I): Since it may not be easy to deal with the notion of truly optimal, Iliescu and Wang proposed the notion of Optimal-I in [30]. Optimal-I has the advantage of being simple to compute since Λ_r^I involves only the POD eigenvalues and modes. Optimal-I is also simple to interpret since from Lemma 2.5 we have

$$\Lambda_r^I = \frac{1}{2N+1}\sum_{n=0}^{N} \|u^n - P_r u^n\|_W^2 + \frac{1}{2N+1}\sum_{n=1}^{N} \|\partial u^n - P_r \partial u^n\|_W^2 . \tag{2.80}$$

Therefore, Λ_r^I is the *total* POD projection error for all of the data using the POD projection P_r and the error norm W.

Optimal-II (Optimal type II): The value of Λ_r^{II} is also relatively straightforward to compute, since it involves only POD eigenvalues, modes, and the projection Π_r^W. Also, by Lemma 2.5 we have

$$\Lambda_r^{II} = \frac{1}{2N+1}\sum_{n=0}^{N} \left\|u^n - \Pi_r^W u^n\right\|_W^2 + \frac{1}{2N+1}\sum_{n=1}^{N} \left\|\partial u^n - \Pi_r^W \partial u^n\right\|_W^2 . \tag{2.81}$$

Since Π_r^W is the W-orthogonal projection, the quantity Λ_r^{II} is the best possible *total* POD data approximation error, and (2.80)–(2.81) imply

$$\Lambda_r^{II} \leq \Lambda_r^I.$$

Optimal-II has the advantage of using a best possible POD approximation error, while also being relatively simple to compute and understand. Finally, we note that if $W = \mathcal{H}$ then $P_r = \Pi_r^W$ and therefore Optimal-I and Optimal-II are identical; however, Optimal-I and Optimal-II may be different if $\mathcal{H} \neq W$.

Comparing the optimality types: Since we are in the DQ case, the pointwise POD projection error result Theorem 2.14 implies that there exists a constant C such that

$$\Lambda_r^\star \leq C\Lambda_r^{II}.$$

The above definitions, observations, and inequalities give the following result comparing the optimality types.

Proposition 2.22. *The following hold:*

(i) If the ROM discretization error is truly optimal, then it is Optimal-II.

(ii) If the ROM discretization error is Optimal-II, then it is Optimal-I.

(iii) If $\mathcal{H} = W$, then Optimal-I and Optimal-II are identical conditions.

(iv) If there exists a constant C such that

$$\|\varphi_i\|_W \leq C\|\varphi_i - \Pi_r^W \varphi_i\|_W, \quad r+1 \leq i \leq s, \tag{2.82}$$

and if the ROM discretization error is Optimal-I, then it is Optimal-II.

In general, we do not know if Optimal-II implies truly optimal; however, again, Λ_r^{II} is easier to deal with compared to $\Lambda_r^\star$. We also do not know in general if Optimal-I implies Optimal-II when $\mathcal{H} \neq W$. We discuss condition (2.82) below.

Remark 2.23 (The noDQ case)**.** In the noDQ case, the same definitions of optimality can be used and Lemma 2.3 also gives interpretations of Λ_r^I and Λ_r^{II} as total POD projections errors in the W norm. As in the DQ case, Optimal-II implies Optimal-I, the two conditions are equivalent if $\mathcal{H} = W$, and Optimal-I with (2.82) implies Optimal-II.

However, as shown in Proposition 2.10, in general we cannot bound the pointwise POD projection error by a constant multiple of the total POD projection error, i.e., Assumption 2.7 is not always satisfied. Thus, we do not know if truly optimal implies Optimal-II. Furthermore, even if Assumption 2.7 is satisfied, the L^2 pointwise error estimate (2.71) in Remark 2.18 is not optimal in any sense, since the second term on its right-hand side contains $\|\nabla\varphi_i\|_{L^2}^2$ instead of $\|\varphi_i\|_{L^2}^2$.

Optimality of Bounds in Section 2.5.1: Next, we consider the optimality type of each pointwise in time error bound for the DQ case from Section 2.5.1. Comparing the pointwise bounds in Lemmas 2.17 and 2.19 to the above optimality definitions gives the following result.

Theorem 2.24. *For the pointwise error bounds in Lemma 2.17 with error norm $W = L^2$:*

(i) If the L^2 POD basis is used (i.e., $\mathcal{H} = L^2$) and there exists a constant C such that

$$\|\varphi_i - R_r(\varphi_i)\|_{L^2} \leq C, \quad r+1 \leq i \leq s, \tag{2.83}$$

then the ROM discretization error in (2.62) is Optimal-I (which is identical to Optimal-II).

(ii) If the H_0^1 POD basis is used (i.e., $\mathcal{H} = H_0^1$), then the ROM discretization error in (2.64) is Optimal-I.

(iii) If the H_0^1 POD basis is used (i.e., $\mathcal{H} = H_0^1$) and condition (2.82) is satisfied (with $W = L^2$), then the ROM discretization error in (2.64) is Optimal-II.

For the pointwise error bounds in Lemma 2.19 with error norm $W = H_0^1$:

(iv) If the L^2 POD basis is used (i.e., $\mathcal{H} = L^2$), then the ROM discretization error in (2.72) is Optimal-II.

(v) If the H_0^1 POD basis is used (i.e., $\mathcal{H} = H_0^1$), then the ROM discretization error in (2.73) is Optimal-I (which is identical to Optimal-II).

Proof. Beginning with (i), the ROM discretization error from (2.62) is given by

$$\Lambda_r = \sum_{i=r+1}^{s} \lambda_i^{DQ} \|\varphi_i - R_r(\varphi_i)\|_{L^2}^2. \tag{2.84}$$

By (2.83), the L^2 orthonormality of the POD basis, and the definition of Optimal-I it follows that

$$\Lambda_r \leq C \sum_{i=r+1}^{s} \lambda_i^{DQ} = C \sum_{i=r+1}^{s} \lambda_i^{DQ} \|\varphi_i\|_{L^2}^2 = C\Lambda_r^I. \tag{2.85}$$

From Proposition 2.22 since $\mathcal{H} = W$ this is identical to Optimal-II.

For (ii) the ROM discretization error from (2.64) is given by

$$\Lambda_r = \sum_{i=r+1}^{s} \lambda_i^{DQ} \|\varphi_i\|_{L^2}^2, \tag{2.86}$$

which is Optimal-I by definition.

Next, (iii) follows from (ii) and Proposition 2.22.

For (iv), the ROM discretization error in (2.72) is given by

$$\Lambda_r = \sum_{i=r+1}^{s} \lambda_i^{DQ} \left(\|\varphi_i - R_r(\varphi_i)\|_{L^2}^2 + \|\nabla(\varphi_i - R_r(\varphi_i))\|_{L^2}^2 \right). \tag{2.87}$$

Applying Poincaré's inequality to $\|\varphi_i - R_r(\varphi_i)\|_{L^2}^2$ shows that Λ_r is Optimal-II.

Finally, to prove (v) we use the fact that $P_r = R_r$ for $\mathcal{H} = H_0^1$, Poincaré's inequality, and the fact that $P_r\varphi_i = 0$ for $i > r$ to obtain

$$\Lambda_r = C \sum_{i=r+1}^{s} \lambda_i^{DQ} \left(\|\varphi_i - P_r(\varphi_i)\|_{L^2}^2 + \|\nabla(\varphi_i - P_r(\varphi_i))\|_{L^2}^2 \right) \leq C \sum_{i=r+1}^{s} \lambda_i^{DQ} \|\nabla\varphi_i\|_{L^2}^2,$$

which is Optimal-I by definition. Since $W = \mathcal{H} = H_0^1$, this is identical to Optimal-II by Proposition 2.22. □

The $W = L^2$ and $\mathcal{H} = H_0^1$ case suggests it may be possible for the ROM discretization error to be Optimal-I but not Optimal-II, since an additional assumption is required for Optimal-II. However, no other case shows a substantial difference between Optimal-I and Optimal-II. It is possible that further differences arise for other partial differential equations; we leave this to be investigated elsewhere.

We note that equations (2.82) and (2.83) are uniform boundedness type conditions for non-orthogonal POD projections. Indeed, for the case $W = \mathcal{H} = L^2$, the Ritz projection $R_r : L^2 \to L^2$ is not orthogonal (even though it is orthogonal when viewed as a mapping $R_r : H_0^1 \to H_0^1$). Thus, (2.83) is a uniform boundedness condition for a non-orthogonal POD projection. Furthermore, for the case $W = L^2$ and $\mathcal{H} = H_0^1$, we have $R_r\varphi_i = 0$ for $i > r$, and so (2.82) can be viewed as

$$\|\varphi_i - R_r\varphi_i\|_{L^2} \leq C\|\varphi_i - \Pi_r^{L^2}\varphi_i\|_{L^2}, \quad r+1 \leq i \leq s. \tag{2.88}$$

Thus, (2.82) is a uniformly bounded comparison of a non-orthogonal POD projection with an orthogonal POD projection. These type of uniform boundedness conditions have been considered in [6, 30, 34, 41, 53, 59], but they are not well understood. We do not consider them further here; we leave them to be more fully explored elsewhere.

2.6 Numerical Results

In this section, we investigate numerically Assumption 2.7. Specifically, we consider the following questions: (i) Is Assumption 2.7 satisfied? (ii) Is the pointwise in time projection error optimal? (iii) Is the pointwise in time ROM error optimal? To investigate these questions numerically, we use the two counterexamples proposed in Sections 2.4.1-2.4.1: counterexample 1, which was defined in (2.44), and counterexample 2, which was defined in (2.46). For each counterexample, we consider both the noDQ case (i.e., when the DQs are not used to construct the ROM basis; see Section 2.3.1) and the DQ case (i.e., when the DQs are used to construct the ROM basis; see Section 2.3.2).

Based on the theoretical results in Sections 2.4.1 and 2.5.1, we expect the noDQ case to (i) violate Assumption 2.7 (see (2.31)); (ii) yield suboptimal pointwise projection errors (see (2.33) in Proposition 2.10); and (iii) yield suboptimal pointwise ROM errors (see (2.71)). In contrast, based on the theoretical results in Sections 2.4.2 and 2.5.1, we expect the DQ case to (i) fulfill Assumption 2.7 (see Theorem 2.14); (ii) yield optimal pointwise projection errors (see Theorem 2.14); and (iii) yield optimal pointwise ROM errors (see 2.62).

In our numerical investigation, we use the one-dimensional heat equation (2.1), which was used in the theoretical development in Section 2.5. For all the numerical experiments, we consider $\nu = 1$. We note that the time step, Δt, plays an important role in our theoretical and numerical investigation. Indeed, an $(N+1) = (T\Delta t^{-1} + 1)$ factor determines the sub-

optimality of the pointwise projection and ROM error bounds for the noDQ case (see (2.33) and (2.71), respectively). Thus, in our numerical investigation it is desirable to consider as many Δt values as possible in order to study the asymptotic behavior of the error as Δt goes to zero. We note, however, that the two counterexamples that we investigate restrict the Δt values that we can consider. The reason is that, while the two counterexamples yield ROM basis functions that are scaled versions of the snapshots (which is advantageous for the theoretical development), the treatment of their boundary conditions is somewhat delicate. Indeed, both counterexamples vanish at $x = 0$, but not at $x = 1$. To simplify the numerical treatment of the right boundary condition, we consider snapshots at Δt values for which $k\,\Delta t$ is an integer. This choice yields snapshots that vanish both at $x = 0$ and at $x = 1$, which allows for a straightforward ROM construction. To summarize, in our numerical investigation we strive to consider optimal k values that are large enough to ensure a large number of Δt values (while satisfying the restriction $k\,\Delta t \in \mathbb{N}$), and also low enough so that the numerical approximation is accurate.

Snapshot Generation Counterexamples 1 and 2 display a highly oscillatory behavior for the relatively large k values chosen (i.e., $k = 128$ and $k = 100$, respectively). Thus, to minimize the numerical error in generating the snapshots, we do not use a standard (e.g., FE) discretization. Instead, to construct the snapshots, we use the analytical forms of counterexamples 1 and 2 given in (2.44) and (2.46), respectively.

ROM Construction To construct the ROM basis, we collect equally spaced snapshots on the time interval $[0, 1]$ and $[0, 0.2]$ for counterexamples 1 and 2, respectively. Thus, the snapshot matrix K is $(N + 1)$-dimensional in the noDQ case, and $(2N + 1)$-dimensional in the DQ case, as explained in Section 2.3.1 and Section 2.3.2, respectively. To construct K, in (2.18) we use the standard Lagrange interpolant operator with respect to the FE nodes to interpolate the analytical solution of counterexamples 1 and 2. Next, we use K to build the ROM basis for the noDQ and DQ cases. We emphasize that, although K has different dimensions in the noDQ and DQ cases, to ensure a fair comparison, we use the same r value in all the numerical experiments. We construct the ROM operators by using the FE mass and stiffness matrices, which are obtained by using a linear FE spatial discretization with mesh size $\Delta h = 1/4096$. As ROM initial condition, we use the L^2 projection of the initial condition in the noDQ case, and the Ritz projection of the initial condition in the DQ case. We use these ROM operators to build the ROM, and run it over the time interval $[0, T]$ with the Crank-Nicolson time discretization and the timestep $\Delta t = T/N$.

2.6.1 Counterexample 1

In this section, we consider counterexample 1, which was proposed in (2.44) of Section 2.4.1. In all the numerical experiments in this section, we consider $k = 128$ in (2.44). The numerical

results are organized as follows: In Section 2.6.1, for both the noDQ and the DQ cases, we investigate numerically whether (i) Assumption 2.7 holds; and (ii) the pointwise projection error is optimal. In Section 2.6.1, for both the noDQ and the DQ cases, we investigate numerically whether the pointwise ROM errors are optimal.

As explained in Section 2.4.1, counterexample 1 was constructed to display the suboptimality of the pointwise projection and ROM bounds when $r = N$ and $t = t_N$. Thus, in our numerical investigation we also consider $r = N$ and $t = t_N$.

Pointwise Projection Error

In this section, we investigate numerically whether Assumption 2.7 holds. To this end, we monitor the magnitude of the projection error (2.10)

$$\left\|\eta^{proj}(.,t_n)\right\|_{L^2} = \left\|u(.,t_n) - \sum_{i=1}^{N} \Big(u(.,t_n),\varphi_i\Big)_{L^2}\varphi_i\right\|_{L^2}, \qquad n = 0,\dots,N, \tag{2.89}$$

at all the time instances, and check whether there are large variations in its magnitude.

Furthermore, for various Δt values, we investigate numerically whether the projection error (2.89) at the last time step is suboptimal (i.e., it has a suboptimal Δt^{-1} factor). Specifically, as shown in (2.30) for counterexample 1 in the noDQ case, the projection error at the last time step satisfies

$$\left\|\eta^{proj}(.,t_N)\right\|_{L^2}^2 = C_{proj}^{noDQ} \sum_{i=N+1}^{N+1} \lambda_i^{noDQ} \left\|\varphi_i\right\|_{L^2}^2, \tag{2.90}$$

where

$$C_{proj}^{noDQ} = T\,\Delta t^{-1} + 1 = (N+1)\,. \tag{2.91}$$

Moreover, as shown in (2.51b) for counterexample 1 in the DQ case, the projection error at the last time step satisfies

$$\left\|\eta^{proj}(.,t_N)\right\|_{L^2}^2 \le C_{proj}^{DQ} \sum_{i=N+1}^{N+1} \lambda_i^{DQ} \left\|\varphi_i\right\|_{L^2}^2, \tag{2.92}$$

where

$$C_{proj}^{DQ} = \mathcal{O}(1). \tag{2.93}$$

In this section, we investigate numerically the scalings (2.90) and (2.92).

n	$\|\|\eta^{proj}(.,t_n)\|\|_{L^2}$	n	$\|\|\eta^{proj}(.,t_n)\|\|_{L^2}$	n	$\|\|\eta^{proj}(.,t_n)\|\|_{L^2}$
0	$2.79e-08$	6	$2.11e-08$	12	$0.00e+00$
1	$2.24e-08$	7	$0.00e+00$	13	$1.49e-08$
2	$2.69e-08$	8	$1.67e-08$	14	$7.45e-09$
3	$7.45e-09$	9	$1.05e-08$	15	$1.67e-08$
4	$1.49e-08$	10	$2.11e-08$	16	$7.07e-01$
5	$1.83e-08$	11	$1.05e-08$		

Table 2.1: Counterexample 1 (2.44), $\Delta t = 1/16$, noDQ case: Pointwise projection error (2.89) at each time step.

noDQ Case In Table 2.1, for the noDQ case, we list the pointwise projection errors (2.89) at each time step. These results show that the pointwise projection error at the last time step is orders of magnitude higher than the pointwise projection error at the other time steps. Thus, we conclude that, in the noDQ case, counterexample 1 violates Assumption 2.7.

In Table 2.2, we list the scaling factor (2.90) for different Δt values. As expected from (2.91), these results show that the scaling factor is equal to $(N+1)$. Thus, we conclude that, in the noDQ case, counterexample 1 yields suboptimal pointwise projection errors.

Δt	1/4	1/8	1/16	1/32	1/64	1/128
C_{proj}^{noDQ}	$5.0e+00$	$9.0e+00$	$1.7e+01$	$3.3e+01$	$6.5e+01$	$1.3e+02$

Table 2.2: Counterexample 1 (2.44), noDQ case: Scaling factor (2.90) for different time step values.

DQ Case In Table 2.3, for the DQ case, we list the pointwise projection errors (2.89) at each time step. These results show that, in contrast with the noDQ case, the pointwise projection error at the last time step is of the same order of magnitude as the pointwise projection error at the other time steps. Thus, we conclude that, in the DQ case, counterexample 1 satisfies Assumption 2.7.

In Table 2.4, we list the scaling factor (2.92) for different time step values. As expected from (2.93), these results show that the scaling factor is bounded. Thus, we conclude that, in the DQ case, counterexample 1 yields optimal pointwise projection errors.

The numerical results in this section support the theoretical results in Section 2.4. Specifically, counterexample 1 satisfies Assumption 2.7 in the DQ case, but not in the noDQ case. Furthermore, the pointwise projection error at the last time step is optimal in the DQ case, and suboptimal in the noDQ case.

n	$\|\|\eta^{proj}(.,t_n)\|\|_{L^2}$	n	$\|\|\eta^{proj}(.,t_n)\|\|_{L^2}$	n	$\|\|\eta^{proj}(.,t_n)\|\|_{L^2}$
0	$1.7144e-01$	6	$1.7144e-01$	12	$1.7146e-01$
1	$1.7144e-01$	7	$1.7145e-01$	13	$1.7146e-01$
2	$1.7144e-01$	8	$1.7145e-01$	14	$1.7146e-01$
3	$1.7144e-01$	9	$1.7145e-01$	15	$1.7146e-01$
4	$1.7144e-01$	10	$1.7145e-01$	16	$1.7147e-01$
5	$1.7144e-01$	11	$1.7146e-01$		

Table 2.3: Counterexample 1 (2.44), $\Delta t = 1/16$, DQ case: Pointwise projection error (2.89) at each time step.

Δt	$1/4$	$1/8$	$1/16$	$1/32$	$1/64$	$1/128$
$\mathcal{C}_{proj}^{DQ}$	$1.8e+00$	$1.9e+00$	$1.9e+00$	$2.0e+00$	$2.0e+00$	$2.0e+00$

Table 2.4: Counterexample 1 (2.44), DQ case: Scaling factor (2.92) for different time step values.

Pointwise ROM Error

In this section, we investigate whether the pointwise ROM error is suboptimal.

noDQ Case In the noDQ case, we investigate numerically the error estimate proved in (2.71):

$$\max_{1\le k\le N}\|e^k\|_{L^2}^2 = \mathcal{O}\left((N+1)\sum_{i=N+1}^{N+1}\lambda_i^{noDQ}\|\varphi_i\|_{L^2}^2 + \Delta t^4 + \sum_{i=N+1}^{N+1}\lambda_i^{noDQ}\|\nabla\varphi_i\|_{L^2}^2\right). \tag{2.94}$$

We note that, since the ROM initial condition is the L^2 projection of the initial condition, the term $\|\phi_r^0\|_{L^2}^2$ in (2.71) vanishes in (2.94). As explained in Remark 2.18, the error bound (2.94) is suboptimal with respect to the time step due to the factor $(N+1) = (\Delta t^{-1}+1)$ in the first term on the right-hand side. To investigate numerically the suboptimality of the error bound (2.94), in Table 2.5 we list the ratio

$$C_{rom}^{noDQ} = \left(\max_{1\le k\le N}\|e^k\|_{L^2}^2\right) \Big/ \left((N+1)\sum_{i=N+1}^{N+1}\lambda_i^{noDQ}\|\varphi_i\|_{L^2}^2 + \Delta t^4 + \sum_{i=N+1}^{N+1}\lambda_i^{noDQ}\|\nabla\varphi_i\|_{L^2}^2\right). \tag{2.95}$$

The results in Table 2.5 show that the ratio (2.95) is bounded from below. Thus, we conclude that the pointwise ROM error in the noDQ case is suboptimal.

Δt	1/4	1/8	1/16	1/32	1/64	1/128
$\mathcal{C}_{rom}^{noDQ}$	$3.0e-04$	$1.8e-04$	$1.0e-04$	$2.0e-04$	$7.6e-04$	$7.9e-04$

Table 2.5: Counterexample 1 (2.44), noDQ case: Ratio (2.95) for different time step values.

To investigate the sensitivity of our numerical results with respect to k (i.e., the level of oscillations in counterexample 1), in Table 2.6 we list the ratio (2.95) for $k = 8$. The results in Table 2.6 confirm the results in Table 2.5, i.e., the pointwise ROM error in the noDQ case is suboptimal.

Δt	1/2	1/4	1/8
$\mathcal{C}_{rom}^{noDQ}$	$3.75e-03$	$6.371e-03$	$1.13-02$

Table 2.6: Counterexample 1 (2.44), $k = 8$, noDQ case: Ratio (2.95) for different time step values.

DQ Case In the DQ case, we investigate numerically the error estimate proved in (2.62):

$$\max_{1\leq k\leq N} \|e^k\|_{L^2}^2 = \mathcal{O}\left(\sum_{i=N+1}^{N+1} \lambda_i^{DQ} \|\varphi_i - R_r(\varphi_i)\|_{L^2}^2 + \Delta t^4\right), \tag{2.96}$$

We note that the error bound (2.96) is optimal. In Table 2.7, we list the ratio

$$C_{rom}^{DQ} = \left(\max_{1\leq k\leq N} \|e^k\|_{L^2}^2\right) / \left(\sum_{i=N+1}^{N+1} \lambda_i^{DQ} \|\varphi_i - R_r(\varphi_i)\|_{L^2}^2 + \Delta t^4\right). \tag{2.97}$$

The results in Table 2.7 show that the ratio (2.97), while increasing, seems to be bounded, as predicted by (2.96).

Δt	1/4	1/8	1/16	1/32	1/64	1/128
C_{rom}^{DQ}	$7.8e-02$	$1.3e-01$	$2.0e-01$	$3.5e-01$	$5.3e-01$	$8.7e-01$

Table 2.7: Counterexample 1 (2.44), DQ case: Ratio (2.97) for different time step values.

The increase of C_{rom}^{DQ} in Table 2.7 is due to the highly oscillatory character of counterexample 1 in (2.44), which makes the ROM simulation in the DQ case challenging. To alleviate the

highly oscillatory behavior of counterexample 1, we keep all the parameters unchanged and choose a lower k value (i.e., $k = 8$) in (2.44), which yields a solution with fewer oscillations. In Table 2.8, we list the ratio (2.97) for $k = 8$. The results in Table 2.8 show that the ratio (2.97) is bounded, as predicted by (2.96).

Δt	1/2	1/4	1/8
$\mathcal{C}_{rom}^{DQ}$	$4.73e-01$	$5.92e-01$	$2.55e-01$

Table 2.8: Counterexample 1 (2.44), $k = 8$, and DQ case: Ratio (2.97) for different time step values.

The numerical results in this section support the theoretical results in Section 2.5. Specifically, for counterexample 1, the pointwise ROM error is optimal in the DQ case, and suboptimal in the noDQ case.

2.6.2 Counterexample 2

In this section, we consider counterexample 2, which was proposed in equation (2.46) of Section 2.4.1. In all the numerical experiments in this section, we consider $k = 100$, $\delta = 0.01$, and $\alpha = 1$ in (2.46). The numerical results are organized as follows: In Section 2.6.2, for both the noDQ and the DQ cases, we investigate numerically whether (i) Assumption 2.7 holds; and (ii) the pointwise projection error is optimal. In Section 2.6.2, for both the noDQ and the DQ cases, we investigate numerically whether the pointwise ROM errors are optimal.

As explained in Section 2.4.1, counterexample 2 was constructed to display the suboptimality of the pointwise projection and ROM error bounds for any r values. In our numerical investigation, we consider general r and $t = t_k$ values for both the pointwise projection error and the pointwise ROM error.

Pointwise Projection Error

In this section, we investigate numerically whether Assumption 2.7 holds. To this end, for various Δt values, we investigate numerically whether the projection error (2.98) at various time instances is suboptimal.

$$\left\|\eta^{proj}(.,t_r)\right\|_{L^2} = \left\|u(.,t_r) - \sum_{i=1}^{r} \Big(u(.,t_r),\varphi_i\Big)_{L^2}\varphi_i\right\|_{L^2}, \qquad r = 1,\ldots,N, \tag{2.98}$$

Specifically, as shown in (2.35) in Proposition 2.10 for counterexample 2 in the noDQ case, for fixed r values, the projection error at $t = t_r$ satisfies

$$\left\|\eta^{proj}(.,t_r)\right\|_{L^2}^2 = C_{proj}^{noDQ}\,(N+1)\sum_{i=r+1}^{N+1}\lambda_i^{noDQ}\left\|\varphi_i\right\|_{L^2}^2, \tag{2.99}$$

where

$$C_{proj}^{noDQ} \geq \frac{\min\{1,\gamma\}}{2}. \tag{2.100}$$

Moreover, as shown in (2.51b) for counterexample 2 in the DQ case, the projection error at various time instances satisfies

$$\max_{0\leq k\leq N}\left\|u(.,t_k)-\sum_{i=1}^{r}\Big(u(.,t_k),\varphi_i\Big)_{L^2}\varphi_i\right\|_{L^2}^2 \leq C_{proj}^{DQ}\sum_{i=r+1}^{d}\lambda_i^{\mathrm{DQ}}\|\varphi_i\|_{L^2}, \tag{2.101}$$

where

C_{proj}^{DQ} is bounded from above. In this section, we investigate numerically the scalings (2.99)–(2.100) and (2.101).

noDQ Case In Table 2.9, for $r = 4$, we list the scaling factor $\mathcal{C}_{proj}^{DQ}$ in (2.99) for different time step values. As expected from (2.100), these results show that the scaling factor is bounded from below. Thus, we conclude that, in the noDQ case, counterexample 2 yields suboptimal pointwise projection errors.

Δt	0.05	0.04	0.02	0.01
$\mathcal{C}_{proj}^{noDQ}$	$1.00e+00$	$9.82e-01$	$8.65e-01$	$6.32e-01$

Table 2.9: Counterexample 2 (2.46), $r = 4$, and noDQ case: Scaling factor (2.99) for different time step values.

DQ Case In Table 2.10, for $r = 4$, we list the scaling factor (2.101) for different time step values. As expected, these results show that the scaling factor is bounded from above. Thus, we conclude that, in the DQ case, counterexample 2 yields optimal pointwise projection errors.

The numerical results in this section support the theoretical results in Section 2.4. Specifically, for a generic r value, counterexample 2 satisfies Assumption 2.7 in the DQ case, but not in the noDQ case. Furthermore, the pointwise projection error is optimal in the DQ case, and suboptimal in the noDQ case.

Δt	0.05	0.04	0.02	0.01
$\mathcal{C}_{proj}^{DQ}$	$1.83e+00$	$1.76e-02$	$8.32e-03$	$3.84e-03$

Table 2.10: Counterexample 2 (2.46), $r = 4$, and DQ case: Scaling factor (2.101) for different time step values.

Pointwise ROM Error

In this section, we investigate whether the pointwise ROM error is suboptimal. We note that the time evolution of the analytical solution in counterexample 2 (which is displayed in Figure 2.1) prompted us to make the following parameter choices in the numerical investigation of the pointwise ROM error. Since the magnitude of the analytical solution is significant on the time interval $[0, 0.04]$ and almost negligible on the time interval $[0.04, 0.2]$, we decided to compute the pointwise ROM errors for both the noDQ and the DQ cases on the time interval $[0, 0.05]$. Furthermore, since the DQ ROM basis functions with large indices are very oscillatory, we decided to use low r values in order to avoid numerical instabilities.

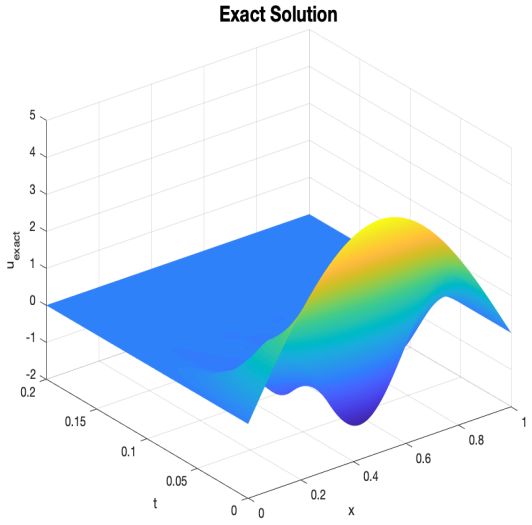

Figure 2.1: Counterexample 2 (2.46), FOM plot: $h = 1/4096$ and $\Delta t = 0.02$.

noDQ Case In the noDQ case, we investigate numerically the error estimate proved in (2.71):

$$\max_{1\leq k\leq N} \|e^k\|_{L^2}^2 = \mathcal{O}\left((N+1)\sum_{i=r+1}^{N+1} \lambda_i^{noDQ}\|\varphi_i\|_{L^2}^2 + \Delta t^4 + \sum_{i=r+1}^{N+1} \lambda_i^{noDQ}\|\nabla\varphi_i\|_{L^2}^2\right). \quad (2.102)$$

We note that, since the ROM initial condition is the L^2 projection of the initial condition, the term $\|\phi_r^0\|_{L^2}^2$ in (2.71) vanishes in (2.102). As explained in Remark 2.18, the error bound (2.102) is suboptimal with respect to the time step due to the factor $(N+1) = (T\Delta t^{-1}+1)$ in the first term on the right-hand side. To investigate numerically the suboptimality of the error bound (2.102), in Table 2.11 we list the ratio (2.103) for fixed Δt values and various r values. The ratios in Table 2.11 are bounded from below. Thus, we conclude that the pointwise ROM error in the noDQ case is suboptimal.

$$C_{rom}^{noDQ} = \left(\max_{1\le k\le N}\|e^k\|_{L^2}^2\right) / \left((N+1)\sum_{i=r+1}^{N+1}\lambda_i^{noDQ}\|\varphi_i\|_{L^2}^2 + \Delta t^4 + \sum_{i=r+1}^{N+1}\lambda_i^{noDQ}\|\nabla\varphi_i\|_{L^2}^2\right). \tag{2.103}$$

r	1	2	3	4	5	6
$\mathcal{C}_{rom}^{noDQ}$	$1.7e-01$	$9.8e-02$	$1.1e-01$	$2.2e-01$	$4.4e-01$	$9.2e-01$

Table 2.11: Counterexample 2 (2.46) and noDQ case: Ratio (2.103) for fixed time step $\Delta t = 0.01$ and different r values.

DQ Case In the DQ case, we investigate numerically the error estimate proved in (2.62):

$$\max_{1\le k\le N}\|e^k\|_{L^2}^2 = \mathcal{O}\left(\sum_{i=r+1}^{N+1}\lambda_i^{DQ}\|\varphi_i - R_r(\varphi_i)\|_{L^2}^2 + \Delta t^4\right). \tag{2.104}$$

To investigate numerically the suboptimality of the error bound (2.104), in Table 2.12 we list the ratio (2.103) for fixed Δt values and various r values. The ratios in Table 2.12 are bounded. Thus, we conclude that the pointwise ROM error in the DQ case is optimal.

$$C_{rom}^{DQ} = \left(\max_{1\le k\le N}\|e^k\|_{L^2}^2\right) / \left(\sum_{i=r+1}^{N+1}\lambda_i^{DQ}\|\varphi_i - R_r(\varphi_i)\|_{L^2}^2 + \Delta t^4\right). \tag{2.105}$$

The numerical results in this section support the theoretical results in Section 2.5. Specifically, for counterexample 2, the pointwise ROM error is optimal in the DQ case, and suboptimal in the noDQ case.

r	1	2	3	4	5	6
$\mathcal{C}_{rom}^{DQ}$	$2.9e-03$	$4.0e-03$	$4.9e-03$	$5.7e-03$	$1.0e-02$	$2.9e-02$

Table 2.12: Counterexample 2 (2.46) and DQ case: Ratio (2.105) for the shorter time interval $[0, 0.05]$ and fixed time step $\Delta t = 0.01$ and different r values.

2.7 Conclusions

In this paper, we resolved several theoretical issues dealing with the optimality of pointwise in time error bounds for POD model order reduction of the heat equation. In particular, we studied the role played by the DQs in the optimality of pointwise POD error bounds with respect to (i) the time discretization error, and (ii) the ROM discretization error.

First, in the noDQ case (i.e., when the DQs are not used to construct the POD basis), we proved that the error bound is suboptimal not only with respect to the ROM discretization (as shown in [30]), but also with respect to the time discretization. Specifically, in Proposition 2.10 we constructed two classes of analytical examples, and we proved that these examples violate Assumption 2.7, and yield suboptimal (with respect to the time discretization) pointwise projection error bounds. Furthermore, we noted that these suboptimal pointwise projection error bounds yield suboptimal ROM error bounds (see Remark 2.18). Finally, we illustrated the suboptimality of the pointwise projection and ROM error bounds in the numerical simulation of the heat equation.

Our second main contribution is Theorem 2.14, where we proved that, in the DQ case (i.e., when the DQs are used to construct the POD basis), Assumption 2.7 is always satisfied. To prove Theorem 2.14, in Lemma 2.13 we first proved a discrete time Sobolev inequality for the DQ case. Next, in Section 2.5, we used Theorem 2.14 to prove pointwise ROM error bounds that are optimal with respect to both the ROM discretization error and the time discretization error in the DQ case. In Section 2.6, we illustrated the optimality of the pointwise projection and error bounds in the numerical simulation of the heat equation.

Our third main contribution is that, in Definition 2.20, we proposed a new definition for the optimality of pointwise in time ROM discretization errors. In Section 2.5.2, we carefully discussed the relationship between this new optimality definition and the other two optimality definitions in current use. In Theorem 2.24, for two of the three optimality definitions, we showed that the DQ case yields optimal bounds, whereas the noDQ case yields suboptimal error bounds.

Our theoretical and numerical investigations (see also [30, 36, 53]) show that the DQs are needed to prove optimal pointwise in time error bounds. There are, however, several research directions that need to be investigated.

At a theoretical level, the uniform boundedness type conditions for non-orthogonal POD projections considered in Proposition 2.22 and Theorem 2.24 are important in proving some

of the optimal pointwise ROM error bounds. These type of uniform boundedness conditions have been studied both theoretically and numerically in [6, 30, 34, 41, 53, 59], but they are not well understood. Further investigation of these conditions is needed.

Additionally, at a theoretical level we considered optimal uniform estimates only for the heat equation. How these estimates will extend to more complicated nonlinear PDEs (e.g., the Navier-Stokes equations) is an open problem.

In this paper, we considered equally spaced snapshots to construct the POD basis. The POD adaptivity in time (see, e.g., [1, 28, 38, 44] and the survey in [18]) aims at choosing snapshot time instances that are optimal in some sense (e.g., such that the error between the ROM and FOM trajectories is minimized [38]). The effect of POD adaptivity in time on the optimality of error bounds in the noDQ and DQ cases should also be investigated.

At a numerical level, further investigation of the role of DQs in practical computations is needed. The theoretical and numerical results in this paper focus exclusively on the optimality of the rates of convergence of ROM error bounds, but do not address the absolute size of the ROM error. In our numerical investigation, the size of the ROM error was of the same order in the noDQ and DQ cases (results not included). In the current literature, the results do not yield a clear conclusion: In some references [26, 35], the ROM error is lower in the DQ case than in the noDQ case; in other references [30, 34, 36], the situation is reversed. Further investigation of the relative size of the ROM error in the noDQ and DQ cases is needed.

At a numerical level, further investigation of the role of DQs in practical computations is needed. In this paper, we focus exclusively on the optimality of the rates of convergence of ROM error bounds. We emphasize, however, that we do not address the size of the ROM error. In our numerical investigation, the size of the ROM error was sometimes lower in the noDQ case and other times lower in the DQ case. Overall, the size of the ROM error was of the same order in the noDQ and DQ cases (results not included). We note that the current literature yields similar qualitative conclusions: In some references [26, 35], the ROM error is lower in the DQ case than in the noDQ case; in other references [30, 34, 36], the situation is reversed. Further investigation of the size of the ROM error in the noDQ and DQ cases is needed.

Bibliography

[1] A. Alla, C. Grässle, and M. Hinze. A posteriori snapshot location for POD in optimal control of linear parabolic equations. *ESAIM: Math. Model. Numer. Anal.*, 52(5):1847–1873, 2018.

[2] A. C. Antoulas, C. A. Beattie, and S. Güğercin. *Interpolatory methods for model reduction*. SIAM, 2020.

[3] M. Azaïez, T. Chacón Rebollo, and S. Rubino. A streamline derivative projection-based POD-ROM for convection-dominated flows. Part I : Numerical Analysis. ArXiv e-prints, https://arxiv.org/abs/1711.09780v1, 2017.

[4] 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. *C. R. Acad. Sci. Paris, Ser. I*, 339:667–672, 2004.

[5] K. Carlberg and C. Farhat. A low-cost, goal-oriented compact proper orthogonal decomposition basis for model reduction of static systems. *Int. J. Num. Meth. Eng.*, 86(3):381–402, 2011.

[6] D. Chapelle, A. Gariah, and J. Sainte-Marie. Galerkin approximation with proper orthogonal decomposition: new error estimates and illustrative examples. *ESAIM: Math. Model. Numer. Anal.*, 46:731–757, 2012.

[7] S. Chaturantabut and D. C. Sorensen. A state space error estimate for POD-DEIM nonlinear model reduction. *SIAM J. Numer. Anal.*, 50:46–63, 2012.

[8] Y. Choi, D. Coombs, and R. Anderson. SNS: a solution-based nonlinear subspace method for time-dependent model order reduction. *SIAM J. Sci. Comput.*, 42(2):A1116–A1146, 2020.

[9] R. F. Curtain and K. Glover. Balanced realisations for infinite dimensional systems. 1985.

[10] V. DeCaria, T. Iliescu, W. Layton, M. McLaughlin, and M. Schneier. An artificial compression reduced order model. *SIAM J. Numer. Anal.*, 58(1):565–589, 2020.

[11] F. G. Eroglu, S. Kaya, and L. G. Rebholz. A modular regularized variational multiscale proper orthogonal decomposition for incompressible flows. *Comput. Meth. Appl. Mech. Eng.*, 325:350–368, 2017.

[12] F. G. Eroglu, S. Kaya, and L. G. Rebholz. POD-ROM for the Darcy-Brinkman equations with double-diffusive convection. *J. Numer. Math.*, 27(3):123–139, 2019.

[13] L. C. Evans. *Partial differential equations*, volume 19 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, second edition, 2010.

[14] L. Fick, Y. Maday, A. T. Patera, and T. Taddei. A stabilized POD model for turbulent flows over a range of Reynolds numbers: Optimal parameter sampling and constrained projection. *J. Comp. Phys.*, 371:214–243, 2018.

[15] 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. *Numer. Math.*, 110(1):49–81, 2008.

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

[17] V. Girault and P.-A. Raviart. *Finite element methods for Navier-Stokes equations*, volume 5 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 1986. Theory and algorithms.

[18] C. Gräßle. *Adaptivity in model order reduction with proper orthogonal decomposition*. PhD thesis, University of Hamburg, 2019.

[19] M. A. Grepl and A. T. Patera. A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations. *ESAIM: Math. Model. Numer. Anal.*, 39(1):157–181, 2005.

[20] M. Gubisch and S. Volkwein. Proper orthogonal decomposition for linear-quadratic optimal control. In *Model reduction and approximation*, volume 15 of *Comput. Sci. Eng.*, pages 3–63. SIAM, Philadelphia, PA, 2017.

[21] M. Gunzburger, N. Jiang, and M. Schneier. An ensemble-proper orthogonal decomposition method for the nonstationary Navier-Stokes equations. *SIAM J. Numer. Anal.*, 55(1):286–304, 2017.

[22] S. Herkt, M. Hinze, and R. Pinnau. Convergence analysis of Galerkin POD for linear second order evolution equations. *Electron. Trans. Numer. Anal.*, 40:321–337, 2013.

[23] J. S. Hesthaven, G. Rozza, and B. Stamm. *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*. Springer, 2015.

[24] S. Hijazi, G. Stabile, A. Mola, and G. Rozza. Data-driven POD-Galerkin reduced order model for turbulent flows. *J. Comput. Phys.*, page 109513, 2020.

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

[26] D. Hömberg and S. Volkwein. Control of laser surface hardening by a reduced-order approach using proper orthogonal decomposition. *Math. Comput. Modelling*, 38(10):1003–1028, 2003.

[27] C. Homescu, L. R. Petzold, and R. Serban. Error estimation for reduced-order models of dynamical systems. *SIAM J. Numer. Anal.*, 43(4):1693–1714 (electronic), 2005.

[28] R. H. W. Hoppe and Z. Liu. Snapshot location by error equilibration in proper orthogonal decomposition for linear and semilinear parabolic partial differential equations. *J. Numer. Math.*, 22(1):1–32, 2014.

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

[30] T. Iliescu and Z. Wang. Are the snapshot difference quotients needed in the proper orthogonal decomposition? *SIAM J. Sci. Comput.*, 36(3):A1221–A1250, 2014.

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

[32] K. Ito and S. S. Ravindran. A reduced-order method for simulation and control of fluid flows. *J. Comput. Phys.*, 143(2):403–425, 1998.

[33] B. Jin and Z. Zhou. An analysis of Galerkin proper orthogonal decomposition for subdiffusion. *ESAIM Math. Model. Numer. Anal.*, 51(1):89–113, 2017.

[34] K. Kean and M. Schneier. Error Analysis of Supremizer Pressure Recovery for POD based Reduced-Order Models of the Time-Dependent Navier–Stokes Equations. *SIAM J. Numer. Anal.*, 58(4):2235–2264, 2020.

[35] T. Kostova-Vassilevska and G. M. Oxberry. Model reduction of dynamical systems by proper orthogonal decomposition: Error bounds and comparison of methods using snapshots from the solution and the time derivatives. *Journal of Computational and Applied Mathematics*, 330:553–573, 2018.

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

[37] K. Kunisch and S. Volkwein. Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. *SIAM J. Numer. Anal.*, 40(2):492–515 (electronic), 2002.

[38] K. Kunisch and S. Volkwein. Optimal snapshot location for computing POD basis functions. *ESAIM: Math. Model. Numer. Anal.*, 44(3):509–529, 2010.

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

[40] F. Leibfritz and S. Volkwein. Numerical feedback controller design for PDE systems using model reduction: techniques and case studies. In *Real-time PDE-constrained optimization*, volume 3 of *Comput. Sci. Eng.*, pages 53–72. SIAM, Philadelphia, PA, 2007.

[41] S. Locke and J. Singler. New proper orthogonal decomposition approximation theory for PDE solution data. *SIAM J. Numer. Anal.*, 58(6):3251–3285, 2020.

[42] Z. Luo, J. Chen, I. M. Navon, and X. Yang. Mixed finite element formulation and error estimates based on proper orthogonal decomposition for the nonstationary Navier-Stokes equations. *SIAM J. Numer. Anal.*, 47(1):1–19, 2008.

[43] M. Mohebujjaman, L. G. Rebholz, X. Xie, and T. Iliescu. Energy balance and mass conservation in reduced order models of fluid flows. *J. Comput. Phys.*, 346:262–277, 2017.

[44] G. M. Oxberry, T. Kostova-Vassilevska, W. Arrighi, and K. Chand. Limited-memory adaptive snapshot selection for proper orthogonal decomposition. *Int. J. Numer. Meth. Engng.*, 109(2):198–217, 2017.

[45] A. Quarteroni, A. Manzoni, and F. Negri. *Reduced Basis Methods for Partial Differential Equations: An Introduction*, volume 92. Springer, 2015.

[46] M. Rathinam and L. R. Petzold. A new look at proper orthogonal decomposition. *SIAM J. Numer. Anal.*, 41(5):1893–1925, 2003.

[47] T. Reis and T. Selig. Balancing transformations for infinite-dimensional systems with nuclear hankel operator. *Integral Equations and Operator Theory*, 79(1):67–105, 2014.

[48] 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. *Arch. Comput. Method. E.*, 15(3):229–275, 2008.

[49] S. Rubino. A streamline derivative POD-ROM for advection-diffusion-reaction equations. *ESAIM: ProcS*, 64:121–136, 2018.

[50] E. Sachs and M. Schu. A priori error estimates for reduced order models in finance. *ESAIM: Math. Model. Numer. Anal.*, 47(2):449–469, 2013.

[51] J. Shen, J. R. Singler, and Y. Zhang. HDG-POD reduced order model of the heat equation. *J. Comput. Appl. Math.*, 362:663–679, 2019.

[52] J. R. Singler. Convergent snapshot algorithms for infinite-dimensional Lyapunov equations. *IMA J. Numer. Anal.*, 31(4):1468–1496, 2011.

[53] J. R. Singler. New POD error expressions, error bounds, and asymptotic results for reduced order models of parabolic PDEs. *SIAM J. Numer. Anal.*, 52(2):852–876, 2014.

[54] V. Thomée. *Galerkin finite element methods for parabolic problems*. Springer Verlag, 2006.

[55] K. Urban and A. T. Patera. A new error bound for reduced basis approximation of parabolic partial differential equations. *C. R. Acad. Sci. Paris Sér. I Math.*, 350(3):203–207, 2012.

[56] K. Veroy and A. T. Patera. Certified real-time solution of the parametrized steady incompressible Navier–Stokes equations: rigorous reduced-basis a posteriori error bounds. *Int. J. Numer. Meth. Fluids*, 47(8-9):773–788, 2005.

[57] S. Volkwein. Proper orthogonal decomposition: Theory and reduced-order modelling. *Lecture Notes, University of Konstanz*, 2013. http://www.math.uni-konstanz.de/numerik/personen/volkwein/teaching/POD-Book.pdf.

[58] M. F. Wheeler. A priori L_2 error estimates for Galerkin approximations to parabolic partial differential equations. *SIAM J. Numer. Anal.*, 10(4):723–759, 1973.

[59] X. Xie, D. Wells, Z. Wang, and T. Iliescu. Numerical analysis of the Leray reduced order model. *J. Comput. Appl. Math.*, 328:12–29, 2018.

[60] M. Yano. Discontinuous Galerkin reduced basis empirical quadrature procedure for model reduction of parametrized nonlinear conservation laws. *Adv. Comput. Math.*, 45:2287–2320, 2019.

[61] C. Zerfas, L. G. Rebholz, M. Schneier, and T. Iliescu. Continuous data assimilation reduced order models of fluid flow. *Comput. Meth. Appl. Mech. Eng.*, 357:112596, 2019.

[62] S. Zhu, L. Dedè, and A. Quarteroni. Isogeometric analysis and proper orthogonal decomposition for parabolic problems. *Numer. Math.*, 135(2):333–370, 2017.

[63] R. Zimmermann. Gradient-enhanced surrogate modeling based on proper orthogonal decomposition. *J. Comput. Appl. Math.*, 237(1):403–418, 2013.

Chapter 3

Data-Driven Variational Multiscale Reduced Order Models

The content of this chapter has been published in *Computer Methods in Applied Mechanics and Engineering (CMAME)* [*]

In that paper, my contribution was being part of the model's conceptual development and presenting numerical experiments for the Burgers equation in Section 3.4.2.

[*]C. Mou, **B. Koc**, O. San, L. G. Rebholz, and T. Iliescu. Data-driven variational multiscale reduced order models. Computer Methods in Applied Mechanics and Engineering, 373:113470.

3.1 Abstract

We propose a new data-driven reduced order model (ROM) framework that centers around the hierarchical structure of the variational multiscale (VMS) methodology and utilizes data to increase the ROM accuracy at a modest computational cost. The VMS methodology is a natural fit for the hierarchical structure of the ROM basis: In the first step, we use the ROM projection to separate the scales into three categories: (i) resolved large scales, (ii) resolved small scales, and (iii) unresolved scales. In the second step, we explicitly identify the VMS-ROM closure terms, i.e., the terms representing the interactions among the three types of scales. In the third step, we use available data to model the VMS-ROM closure terms. Thus, instead of phenomenological models used in VMS for standard numerical discretizations (e.g., eddy viscosity models), we utilize available data to construct new structural VMS-ROM closure models. Specifically, we build ROM operators (vectors, matrices, and tensors) that are closest to the true ROM closure terms evaluated with the available data. We test the new data-driven VMS-ROM in the numerical simulation of four test cases: (i) the 1D Burgers equation with viscosity coefficient $\nu = 10^{-3}$; (ii) a 2D flow past a circular cylinder at Reynolds numbers $Re = 100$, $Re = 500$, and $Re = 1000$; (iii) the quasi-geostrophic equations at Reynolds number $Re = 450$ and Rossby number $Ro = 0.0036$; and (iv) a 2D flow over a backward facing step at Reynolds number $Re = 1000$. The numerical results show that the data-driven VMS-ROM is significantly more accurate than standard ROMs.

3.2 Introduction

For structure dominated systems, *reduced order models (ROMs)* [16, 25, 28, 30, 53, 60, 61, 76, 79, 80] can decrease the full order model (FOM) computational cost by orders of magnitude. ROMs are low-dimensional models that are constructed from available data: In an offline stage, the FOM is run for a small set of parameters to construct a low-dimensional ROM basis $\{\varphi_1, \ldots, \varphi_r\}$, which is used to build the ROM:

$$\dot{\boldsymbol{a}} = \boldsymbol{F}(\boldsymbol{a}), \tag{3.1}$$

where $\boldsymbol{a}$ is the vector of coefficients in the ROM approximation $\sum_{i=1}^{r} a_i(t)\varphi_i(\boldsymbol{x})$ of the variable of interest and $\boldsymbol{F}$ comprises the ROM operators (e.g., vectors, matrices, and tensors) that are preassembled from the ROM basis in the offline stage. In the online stage, the low-dimensional ROM (3.1) is then used in a regime that is different from the training regime. Since the ROM (3.1) is low-dimensional, its computational cost is orders of magnitude lower than the FOM cost.

Unfortunately, current ROMs cannot be used in complex, realistic settings, since they require too many modes (degrees of freedom). For example, to capture all the relevant scales in practical engineering flows, hundreds [56, 77] and even thousands of ROM modes can be

necessary [56, 86]. Thus, although ROMs decrease the FOM computational cost by orders of magnitude, they cannot be used in many important practical settings (e.g., digital twin applications, where a real-time control of physical assets may be required [27]).

One of the main roadblocks in the development of ROMs for complex practical settings is their notorious inaccuracy. The drastic ROM truncation is one of the most important reasons for the ROMs' numerical inaccuracy: Instead of using a sufficient number of ROM modes $\{\varphi_1, \ldots, \varphi_R\}$ to capture the dynamics of the underlying system, current ROMs use only a handful of ROM modes $\{\varphi_1, \ldots, \varphi_r\}$ to ensure a low computational cost. This drastic truncation yields acceptable results in simple, academic test problems, but yields inaccurate results in many practical settings [56], where the *ROM closure problem* [4, 6, 7, 12, 19, 26, 30, 45, 46, 47, 57, 66, 78, 83, 84] needs to be solved: One needs to model the effect of the discarded ROM modes $\{\varphi_{r+1}, \ldots, \varphi_R\}$ on the ROM dynamics, i.e., on the time evolution of the resolved ROM modes $\{\varphi_1, \ldots, \varphi_r\}$:

$$\dot{\boldsymbol{a}} = \boldsymbol{F}(\boldsymbol{a}) + \text{Closure}(\boldsymbol{a}), \tag{3.2}$$

where $\text{Closure}(\boldsymbol{a})$ is a low-dimensional term that models the effect of the discarded ROM modes $\{\varphi_{r+1}, \ldots, \varphi_R\}$ on $\{\varphi_1, \ldots, \varphi_r\}$.

The closure problem is ubiquitous in the numerical simulation of complex systems. For example, classical numerical discretization of turbulent flows (e.g., finite element or finite volume methods), inevitably takes place in the *under-resolved regime* (e.g., on coarse meshes) and requires closure modeling (i.e., modeling the sub-grid scale effects). In classical CFD, e.g., large eddy simulation (LES), there are hundreds (if not thousands) of closure models [70]. This is in stark contrast with ROM, where only relatively few ROM closure models have been investigated. The reason for the discrepancy between ROM closure and LES closure is that the latter has been entirely built around physical insight stemming from Kolmogorov's statistical theory of turbulence (e.g., the concept of eddy viscosity), which is generally posed in the Fourier setting [70]. This physical insight is generally not available in a ROM setting. Thus, current ROM closure models have generally been deprived of many tools of this powerful methodology that represents the core of most LES closure models. Since physical insight cannot generally be used in the ROM setting, alternative ROM closure modeling strategies need to be developed. Our vision is that *data* represents a natural solution for ROM closure modeling.

In this paper, we put forth a new ROM framework that centers around the hierarchical structure of *variational multiscale (VMS)* methodology [31, 32, 33, 34], which naturally separates the scales into (i) resolved large, (ii) resolved small, and (iii) unresolved. We also construct new structural ROM closure models for the three scales by using available data. We believe that the VMS methodology is a natural fit for the hierarchical structure of the ROM basis: In the first step of the new VMS-ROM framework, we use the ROM projection to unambiguously separate the scales into three categories: (i) *resolved large* scales, (ii) *resolved small* scales, and (iii) *unresolved* scales. In the second step, we explicitly identify

the *ROM closure* terms representing the interactions among the three types of scales by projecting the equations onto the corresponding resolved large, resolved small, and unresolved spaces. In the third step, instead of phenomenological modeling techniques used in VMS for standard discretizations (e.g., finite element methods), we utilize *data-driven modeling* [9, 44, 59] to construct novel, robust, *structural* ROM closure models. Thus, instead of ad hoc, phenomenological models used in VMS for standard numerical discretizations (e.g., eddy viscosity models), we utilize available data to construct new structural models for the interaction among the three types of scales. Specifically, we use FOM data to develop VMS-ROM closure terms that account for the *under-resolved* numerical regime. We emphasize that, in the new *data-driven VMS-ROM (DD-VMS-ROM)* framework, we use data only to *complement* classical physical modeling (i.e., only for closure modeling) [49, 85], not to completely replace it [9, 62]. Thus, the resulting ROM framework combines the strengths of both physical and data-driven modeling.

Previous Relevant Work The VMS methodology has been used in ROM settings [8, 18, 24, 35, 37, 69, 78, 84]. We emphasize, however, that the DD-VMS-ROM framework that we propose is different from the other VMS-ROMs.

The VMS-ROMs in [8, 35, 37, 78, 84] are *phenomenological* models in which the role of the VMS closure models is to *dissipate energy* from the ROM. In contrast, the new DD-VMS-ROM utilizes data to construct general structural VMS-ROM closure terms, which are *not required to be dissipative*. (Of course, if deemed appropriate, we may impose additional constraints to mimic the physical properties of the underlying system [50].)

The new DD-VMS-ROM is also different from the reduced-order subscales ROM proposed in [5] (see also [67, 68, 81]): The reduced-order subscales model in [5] minimizes the difference between the *solutions* of the FOM and ROM (see equations (18)–(19) in [6]), whereas the new DD-VMS-ROM minimizes the difference between the VMS-ROM *closure terms* and the "true" (i.e., high-resolution) closure terms. Furthermore, the reduced-order subscales model in [5] builds *linear* closure models (see also [55]), whereas the new DD-VMS-ROM constructs *nonlinear* closure models.

Another ROM closure strategy that is related to the VMS-ROM framework is the adjoint Petrov-Galerkin method [58] (see [10, 11, 23] for related work), which is based on the Mori-Zwanzig (MZ) formalism [21, 43]. In the MZ-ROM approach, the ROM closure model is represented by a memory term that depends on the temporal history of the resolved scales. The memory term is approximated to construct effective ROM closure models and, therefore, practical ROMs. The main difference between the adjoint Petrov-Galerkin method proposed in [58] and the new DD-VMS-ROM is the tool used to define the ROM closure term: The former uses a statistical tool (i.e., the MZ formalism), whereas the latter utilizes a spectral-like projection (i.e., the ROM projection).

Finally, we note that the VMS-ROM framework proposed herein belongs to the wider class of

hybrid physical/data-driven ROMs, in which data-driven modeling is used to model only the missing information (i.e., the ROM closure term) in ROMs constructed from first principles (i.e., from a Galerkin projection of the underlying equations); see, e.g., [4, 12, 15, 20, 29, 45, 54, 58, 85].

The rest of the paper is organized as follows: In Section 3.3, we introduce the new DD-VMS-ROM. In Section 3.4, we test the DD-VMS-ROM in the numerical simulation of four test cases: (i) the 1D Burgers equation with viscosity coefficient $\nu = 10^{-3}$; (ii) a 2D flow past a circular cylinder at Reynolds numbers $Re = 100$, $Re = 500$, and $Re = 1000$; (iii) the quasi-geostrophic equations at Reynolds number $Re = 450$ and Rossby number $Ro = 0.0036$; and (iv) a 2D flow over a backward facing step at Reynolds number $Re = 1000$. Finally, in Section 3.5, we draw conclusions and outline future research directions.

3.3 Data-Driven Variational Multiscale Reduced Order Models (DD-VMS-ROMs)

In this section, we construct the new *data-driven VMS-ROM (DD-VMS-ROM)* framework, which can significantly increase the accuracy of under-resolved ROMs, i.e., ROMs whose dimension is too low to capture the complex dynamics of realistic applications. In Section 3.3.1 we briefly sketch the VMS methodology for general numerical discretizations (see, e.g., [3, 39] for more details), and in Section 3.3.2 we outline the standard Galerkin ROMs.

We construct the new DD-VMS-ROM in two stages: In Section 3.3.3, we construct the two-scale DD-VMS-ROM, which is the simplest DD-VMS-ROM. We note that the two-scale data-driven VMS-ROM was investigated in [85] under the name "data-driven filtered ROM" and in [50, 52] under the name "data-driven correction ROM." However, we decided to outline the construction of the two-scale data-driven VMS-ROM since it is the most straightforward illustration of the DD-VMS-ROM framework.

In Section 3.3.4, we construct the novel three-scale DD-VMS-ROM. This new model separates the scales into three categories (instead of two, as in the two-scale DD-VMS-ROM), which allows more flexibility in constructing the ROM closure models and could lead to more accurate ROMs.

3.3.1 Classical VMS

The VMS methods are general numerical discretizations that increase the *accuracy* of classical Galerkin approximations in *under-resolved* simulations, e.g., on coarse meshes or when not enough basis functions are available. The VMS framework, which was proposed by Hughes and coworkers [31, 32, 33, 34], has made a profound impact in several areas of com-

putational mathematics (see, e.g., [3, 14, 39, 64] for surveys). To illustrate the standard VMS methodology, we consider a general nonlinear system/PDE

$$\dot{\boldsymbol{u}} = \boldsymbol{f}(\boldsymbol{u}) \,, \tag{3.3}$$

whose weak (variational) form is

$$(\dot{\boldsymbol{u}}, \boldsymbol{v}) = (\boldsymbol{f}(\boldsymbol{u}), \boldsymbol{v}), \qquad \forall\, \boldsymbol{v} \in \boldsymbol{X}, \tag{3.4}$$

where $\boldsymbol{f}$ is a general nonlinear function and $\boldsymbol{X}$ is an appropriate infinite dimensional space. To build the VMS framework, we start with a sequence of *hierarchical spaces* of increasing resolutions: $\boldsymbol{X}_1, \boldsymbol{X}_1 \oplus \boldsymbol{X}_2, \boldsymbol{X}_1 \oplus \boldsymbol{X}_2 \oplus \boldsymbol{X}_3, \ldots$. Next, we project system (3.3) onto *each* of the spaces $\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{X}_3, \ldots$, which yields a separate equation for each space. The goal is, of course, to solve for the $\boldsymbol{u}$ component that lives in the coarsest space (i.e., $\boldsymbol{X}_1$), since this yields the lowest-dimensional system:

$$(\dot{\boldsymbol{u}}, \boldsymbol{v}_1) = \big(\boldsymbol{f}(\boldsymbol{u}), \boldsymbol{v}_1\big) \qquad \forall\, \boldsymbol{v}_1 \in \boldsymbol{X}_1 \,. \tag{3.5}$$

System (3.5), however, is *not* closed, since its right-hand side

$$\big(\boldsymbol{f}(\boldsymbol{u}), \boldsymbol{v}_1\big) = \big(\boldsymbol{f}(\boldsymbol{u}_1 + \boldsymbol{u}_2 + \boldsymbol{u}_3 + \ldots), \boldsymbol{v}_1\big) \qquad \forall\, \boldsymbol{v}_1 \in \boldsymbol{X}_1 \,, \tag{3.6}$$

involves $\boldsymbol{u}$ components that do *not* live in $\boldsymbol{X}_1$ (i.e., $\boldsymbol{u}_2 \in \boldsymbol{X}_2, \boldsymbol{u}_3 \in \boldsymbol{X}_3, \ldots$). This coupling is mainly due to the *nonlinearity* of $\boldsymbol{f}$. Thus, the *VMS closure problem* needs to be solved, i.e., (3.6) needs to be approximated in $\boldsymbol{X}_1$. The VMS (3.5) equipped with an appropriate closure model yields an accurate approximation of the large scale $\boldsymbol{X}_1$ component of $\boldsymbol{u}$.

The main reasons for the VMS framework's impressive success are its *utter simplicity* and its *generality* (it can be applied to *any* Galerkin based numerical discretization). The classical VMS methodology, however, is facing several major challenges: (i) The *hierarchical spaces* can be difficult to construct in classical Galerkin methods (e.g., finite elements); and (ii) Developing *VMS closure* models for the coupling terms (i.e., the terms that model the interactions among scales) can be challenging.

In this paper, we propose a new data-driven VMS-ROM framework that overcomes these major challenges of standard VMS methodology: (i) The ROM setting allows a natural, straightforward construction of ROM hierarchical spaces. (ii) We use available data to construct data-driven VMS-ROM closure models. Thus, we avoid the ad hoc assumptions and phenomenological arguments that are often used in traditional VMS closures.

3.3.2 Galerkin ROM (G-ROM)

Before building the new VMS-ROM framework, we sketch the standard Galerkin ROM derivation: (i) Use available data (snapshots) for few parameter values to construct orthonormal modes $\{\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_R\}$, $R = \mathcal{O}(10^3)$, which represent the recurrent spatial structures; (ii) Choose the dominant modes $\{\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_r\}$, $r = \mathcal{O}(10)$, as basis functions for the

ROM; (iii) Use a Galerkin truncation $\boldsymbol{u}_r(\boldsymbol{x},t) = \sum_{j=1}^{r} a_j(t)\,\boldsymbol{\varphi}_j(\boldsymbol{x})$; (iv) Replace $\boldsymbol{u}$ with $\boldsymbol{u}_r$ in (3.3); (v) Use a Galerkin projection of the PDE obtained in step (iv) onto the ROM space $\boldsymbol{X}^r := \text{span}\{\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_r\}$ to obtain an r-dimensional system, which is the *Galerkin ROM (G-ROM)*:

$$\left(\dot{\boldsymbol{u}}_r\,, \boldsymbol{\varphi}_i\right) = \left(\boldsymbol{f}(\boldsymbol{u}_r)\,, \boldsymbol{\varphi}_i\right), \qquad i = 1, \ldots, r; \tag{3.7}$$

(vi) In an offline stage, compute the ROM operators; (vii) In an online stage, repeatedly use the G-ROM (3.7) (for parameters different from the training parameters and/or longer time intervals).

We illustrate the G-ROM for the Navier-Stokes equations (NSE):

$$\frac{\partial \boldsymbol{u}}{\partial t} - Re^{-1}\Delta \boldsymbol{u} + \boldsymbol{u}\cdot\nabla\boldsymbol{u} + \nabla p = \boldsymbol{0}\,, \tag{3.8}$$

$$\nabla\cdot\boldsymbol{u} = 0\,, \tag{3.9}$$

where $\boldsymbol{u}$ is the velocity, p the pressure, and Re the Reynolds number. For clarity of presentation, we use homogeneous Dirichlet boundary conditions. The NSE (3.8)–(3.9) can be cast in the general form (3.3) by choosing $\boldsymbol{f} = Re^{-1}\Delta\boldsymbol{u} - \boldsymbol{u}\cdot\nabla\boldsymbol{u}$ and $\boldsymbol{X}$ the space of weakly divergence-free functions in $\boldsymbol{H}_0^1$. For the NSE, the G-ROM reads

$$\dot{\boldsymbol{a}} = A\,\boldsymbol{a} + \boldsymbol{a}^{\top} B\,\boldsymbol{a}, \tag{3.10}$$

where $\boldsymbol{a}(t)$ is the vector of unknown coefficients $a_j(t), 1 \le j \le r$, A is an $r \times r$ matrix with entries $A_{im} = -Re^{-1}\left(\nabla\boldsymbol{\varphi}_m, \nabla\boldsymbol{\varphi}_i\right)$, and B is an $r \times r \times r$ tensor with entries $B_{imn} = -\left(\boldsymbol{\varphi}_m\cdot\nabla\boldsymbol{\varphi}_n, \boldsymbol{\varphi}_i\right)$, $1 \le i, m, n \le r$. The G-ROM (3.10) does not include a pressure approximation, since we assumed that the ROM modes are discretely divergence-free (which is the case if, e.g., the snapshots are discretely divergence-free). ROMs that provide a pressure approximation are discussed in, e.g., [17, 28, 61]. Once the matrix A and tensor B are assembled in the offline stage, the G-ROM (3.10) is a low-dimensional, efficient dynamical system that can be used in the online stage for numerous parameter values. We emphasize, however, that the G-ROM generally yields inaccurate results when used in *under-resolved*, realistic, complex flows [30, 53, 56, 84].

3.3.3 Two-Scale Data-Driven Variational Multiscale ROMs (2S-DD-VMS-ROM)

The first DD-VMS-ROM that we outline is the *two-scale data-driven VMS-ROM (2S-DD-VMS-ROM)*, which utilizes two orthogonal spaces, $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$. Since the ROM basis is orthonormal by construction, we can build the two orthogonal spaces in a natural way: $\boldsymbol{X}_1 := \text{span}\{\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_r\}$, which represents the resolved ROM scales, and $\boldsymbol{X}_2 := \text{span}\{\boldsymbol{\varphi}_{r+1}, \ldots, \boldsymbol{\varphi}_R\}$, which represents the unresolved ROM scales. We note that, in practical settings, we are forced to use *under-resolved* ROMs, i.e., ROMs whose dimension r is much

lower than the dimension of the snapshot data set (i.e., R). Next, we use the best ROM approximation of $\boldsymbol{u}$ in the space $\boldsymbol{X}_1 \oplus \boldsymbol{X}_2$, i.e., $\boldsymbol{u}_R \in \boldsymbol{X}_1 \oplus \boldsymbol{X}_2$ defined as

$$\boldsymbol{u}_R = \sum_{j=1}^{R} a_j \, \boldsymbol{\varphi}_j = \sum_{j=1}^{r} a_j \, \boldsymbol{\varphi}_j + \sum_{j=r+1}^{R} a_j \, \boldsymbol{\varphi}_j = \boldsymbol{u}_r + \boldsymbol{u}' \,, \tag{3.11}$$

where $\boldsymbol{u}_r \in \boldsymbol{X}_1$ represents the resolved ROM component of $\boldsymbol{u}$, and $\boldsymbol{u}' \in \boldsymbol{X}_2$ represents the unresolved ROM component of $\boldsymbol{u}$. Plugging $\boldsymbol{u}_R$ in (3.3), projecting the resulting equation onto $\boldsymbol{X}_1$, and using the ROM basis orthogonality to show that $\left(\dot{\boldsymbol{u}}_R \,, \boldsymbol{\varphi}_i\right) = \left(\dot{\boldsymbol{u}}_r \,, \boldsymbol{\varphi}_i\right)$, $\forall \, i = 1, \ldots, r$, we obtain

$$\left(\dot{\boldsymbol{u}}_r \,, \boldsymbol{\varphi}_i\right) = \left(\boldsymbol{f}(\boldsymbol{u}_r) \,, \boldsymbol{\varphi}_i\right) + \underbrace{\boxed{\left[\left(\boldsymbol{f}(\boldsymbol{u}_R) \,, \boldsymbol{\varphi}_i\right) - \left(\boldsymbol{f}(\boldsymbol{u}_r) \,, \boldsymbol{\varphi}_i\right)\right]}}_{\text{VMS-ROM closure term}} \,, \quad \forall \, i = 1, \ldots, r. \tag{3.12}$$

The boxed term in (3.12) is the *VMS-ROM closure term*, which models the interaction between the ROM modes $\{\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_r\}$ and the discarded ROM modes $\{\boldsymbol{\varphi}_{r+1}, \ldots, \boldsymbol{\varphi}_R\}$. The VMS-ROM closure term is essential for the accuracy of (3.12): If we drop the VMS-ROM closure term, we are left with the G-ROM (3.7), which yields inaccurate results in the under-resolved regime. The VMS-ROM closure term is a *correction term* that ensures an accurate approximation of $\boldsymbol{u}_r \in \boldsymbol{X}_1$ in the higher-dimensional space $\boldsymbol{X}_1 \oplus \boldsymbol{X}_2$.

Next, we approximate the VMS-ROM closure term with $\boldsymbol{g}(\boldsymbol{u}_r)$, where $\boldsymbol{g}$ is a *generic* function whose coefficients/parameters still need to be determined:

$$\text{VMS-ROM closure term} = \left[\left(\boldsymbol{f}(\boldsymbol{u}_R) \,, \boldsymbol{\varphi}_i\right) - \left(\boldsymbol{f}(\boldsymbol{u}_r) \,, \boldsymbol{\varphi}_i\right)\right] \approx \left(\boldsymbol{g}(\boldsymbol{u}_r) \,, \boldsymbol{\varphi}_i\right). \tag{3.13}$$

To determine the coefficients/parameters in $\boldsymbol{g}$ used in (3.13), in the offline stage, we solve the following low-dimensional *least squares problem*:

$$\boxed{\min_{\boldsymbol{g} \text{ parameters}} \sum_{j=1}^{M} \left\| \left[\left(\boldsymbol{f}(\boldsymbol{u}_R^{FOM}(t_j)) \,, \boldsymbol{\varphi}_i\right) - \left(\boldsymbol{f}(\boldsymbol{u}_r^{FOM}(t_j)) \,, \boldsymbol{\varphi}_i\right)\right] - \left(\boldsymbol{g}(\boldsymbol{u}_r^{FOM}(t_j)) \,, \boldsymbol{\varphi}_i\right) \right\|^2 \,,} \tag{3.14}$$

where $\boldsymbol{u}_R^{FOM}$ and $\boldsymbol{u}_r^{FOM}$ are obtained from the FOM data and M is the number of snapshots. Once $\boldsymbol{g}$ is determined, the model (3.12) with the VMS-ROM closure term replaced by $\boldsymbol{g}$ yields the *two-scale data-driven VMS-ROM (2S-DD-VMS-ROM)*:

$$\boxed{\left(\dot{\boldsymbol{u}}_r \,, \boldsymbol{\varphi}_i\right) = \left(\boldsymbol{f}(\boldsymbol{u}_r) \,, \boldsymbol{\varphi}_i\right) + \left(\boldsymbol{g}(\boldsymbol{u}_r) \,, \boldsymbol{\varphi}_i\right),} \qquad i = 1, \ldots, r. \tag{3.15}$$

We emphasize that, in contrast to the traditional VMS methodology, the 2S-DD-VMS-ROM framework allows *great flexibility* in choosing the *structure* of the closure term. For example,

for the NSE, the approximation (3.13) becomes: $\forall\, i = 1, \ldots, r$,

$$\begin{aligned} \text{VMS-ROM closure term} &= -\left[\left(\left(\boldsymbol{u}_R \cdot \nabla\right) \boldsymbol{u}_R \,, \boldsymbol{\varphi}_i\right) - \left(\left(\boldsymbol{u}_r \cdot \nabla\right) \boldsymbol{u}_r \,, \boldsymbol{\varphi}_i\right)\right] \\ &\approx \left(\boldsymbol{g}(\boldsymbol{u}_r) \,, \boldsymbol{\varphi}_i\right) \\ &= \left(\tilde{A}\, \boldsymbol{a} + \boldsymbol{a}^\top \tilde{B}\, \boldsymbol{a}\right)_i \,, \end{aligned} \tag{3.16}$$

where, for computational efficiency, we assume that the structures of $\boldsymbol{g}$ and $\boldsymbol{f}$ are similar. Thus, in the least squares problem (3.14), we solve for *all* the entries in the $r \times r$ matrix $\tilde{A}$ and the $r \times r \times r$ tensor $\tilde{B}$:

$$\min_{\tilde{A},\tilde{B}} \sum_{j=1}^{M} \Bigg\| -\left[\left(\left(\boldsymbol{u}_R{}^{FOM}(t_j) \cdot \nabla\right) \boldsymbol{u}_R{}^{FOM}(t_j) \,, \boldsymbol{\varphi}_i\right) - \left(\left(\boldsymbol{u}_r{}^{FOM}(t_j) \cdot \nabla\right) \boldsymbol{u}_r{}^{FOM}(t_j) \,, \boldsymbol{\varphi}_i\right)\right]$$

$$-\left(\tilde{A}\, \boldsymbol{a}^{FOM}(t_j) + \boldsymbol{a}^{FOM}(t_j)^\top \tilde{B}\, \boldsymbol{a}^{FOM}(t_j)\right) \Bigg\|^2 , \tag{3.17}$$

where $\boldsymbol{u}_R^{FOM}, \boldsymbol{u}_r^{FOM}$, and $\boldsymbol{a}^{FOM}$ are obtained from the available FOM data. Specifically, the values $\boldsymbol{a}^{FOM}(t_j)$, computed at snapshot time instances $\mathrm{t}_j, j = 1, \cdots, M$, are obtained by projecting the corresponding snapshots $\boldsymbol{u}(t_j)$ onto the ROM basis functions $\boldsymbol{\varphi}_i$ and using the orthogonality of the ROM basis functions: $\forall i = 1, \cdots, R$, $\forall j = 1, \cdots, M$,

$$a_i^{FOM}(t_j) = \Big(\boldsymbol{u}(t_j), \boldsymbol{\varphi}_i\Big). \tag{3.18}$$

In addition,

$$\boldsymbol{u}_R^{FOM}(t_j) = \sum_{k=1}^{R} a_k^{FOM}(t_j)\boldsymbol{\varphi}_k, \qquad \boldsymbol{u}_r^{FOM}(t_j) = \sum_{k=1}^{r} a_k^{FOM}(t_j)\boldsymbol{\varphi}_k. \tag{3.19}$$

The least squares problem (3.17) is *low-dimensional* since, for a small r value, seeks the optimal $(r^2 + r^3)$ entries in $\tilde{A}$ and $\tilde{B}$, respectively. Thus, (3.17) can be efficiently solved in the offline stage. For the NSE, the 2S-DD-VMS-ROM (3.15) takes the form

$$\boxed{\dot{\boldsymbol{a}} = (A + \tilde{A})\boldsymbol{a} + \boldsymbol{a}^\top (B + \tilde{B})\boldsymbol{a}\,,} \tag{3.20}$$

where A and B are the G-ROM operators in (3.10), and $\tilde{A}$ and $\tilde{B}$ are the VMS-ROM closure operators constructed in (3.17).

3.3.4 Three-Scale Data-Driven Variational Multiscale ROMs (3S-DD-VMS-ROM)

The 2S-DD-VMS-ROM (3.15) is based on the two-scale decomposition of $\boldsymbol{u}_R \in \boldsymbol{X}_1 \oplus \boldsymbol{X}_2$ into resolved and unresolved scales: $\boldsymbol{u}_R = \boldsymbol{u}_r + \boldsymbol{u}'$. The flexibility of the hierarchical structure of the ROM space allows a three-scale decomposition of $\boldsymbol{u}_R$, which yields a *three-scale*

data-driven VMS-ROM (3S-DD-VMS-ROM) that is more accurate than the 2S-DD-VMS-ROM (3.15). To construct the new 3S-DD-VMS-ROM, we first build three orthogonal spaces, $\boldsymbol{X}_1, \boldsymbol{X}_2$, and $\boldsymbol{X}_3$: $\boldsymbol{X}_1 := \text{span}\{\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_{r_1}\}$, which represents the *large resolved* ROM scales, $\boldsymbol{X}_2 := \text{span}\{\boldsymbol{\varphi}_{r_1+1}, \ldots, \boldsymbol{\varphi}_r\}$, which represents the *small resolved* ROM scales, and $\boldsymbol{X}_3 := \text{span}\{\boldsymbol{\varphi}_{r+1}, \ldots, \boldsymbol{\varphi}_R\}$, which represents the *unresolved* ROM scales. Next, we consider the best ROM approximation of $\boldsymbol{u}$ in the space $\boldsymbol{X}_1 \oplus \boldsymbol{X}_2 \oplus \boldsymbol{X}_3$, i.e., $\boldsymbol{u}_R \in \boldsymbol{X}_1 \oplus \boldsymbol{X}_2 \oplus \boldsymbol{X}_3$ defined as

$$\begin{aligned} \boldsymbol{u}_R &= \sum_{j=1}^{R} a_j \, \boldsymbol{\varphi}_j \\ &= \sum_{j=1}^{r_1} a_j \, \boldsymbol{\varphi}_j + \sum_{j=r_1+1}^{r} a_j \, \boldsymbol{\varphi}_j + \sum_{j=r+1}^{R} a_j \, \boldsymbol{\varphi}_j \\ &= \boldsymbol{u}_L + \boldsymbol{u}_S + \boldsymbol{u}' , \end{aligned} \tag{3.21}$$

where $\boldsymbol{u}_L \in \boldsymbol{X}_1$ represents the large resolved ROM component of $\boldsymbol{u}_R$, $\boldsymbol{u}_S \in \boldsymbol{X}_2$ represents the small resolved ROM component of $\boldsymbol{u}_R$, and $\boldsymbol{u}' \in \boldsymbol{X}_3$ represents the unresolved ROM component of $\boldsymbol{u}_R$. Thus, with the notation from Section 3.3.3, $\boldsymbol{u}_r = \boldsymbol{u}_L + \boldsymbol{u}_S$. We plug $\boldsymbol{u}_R$ in (3.3), and project the resulting equation onto both $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$:

$$\left(\dot{\boldsymbol{u}}_L , \boldsymbol{\varphi}_i\right) = \left(\boldsymbol{f}(\boldsymbol{u}_L + \boldsymbol{u}_S) , \boldsymbol{\varphi}_i\right) + \boxed{\left[\left(\boldsymbol{f}(\boldsymbol{u}_R) , \boldsymbol{\varphi}_i\right) - \left(\boldsymbol{f}(\boldsymbol{u}_L + \boldsymbol{u}_S) , \boldsymbol{\varphi}_i\right)\right]} , \quad \forall\, i = 1, \ldots, r_1, \tag{3.22}$$

$$\left(\dot{\boldsymbol{u}}_S , \boldsymbol{\varphi}_i\right) = \left(\boldsymbol{f}(\boldsymbol{u}_L + \boldsymbol{u}_S) , \boldsymbol{\varphi}_i\right) + \boxed{\left[\left(\boldsymbol{f}(\boldsymbol{u}_R) , \boldsymbol{\varphi}_i\right) - \left(\boldsymbol{f}(\boldsymbol{u}_L + \boldsymbol{u}_S) , \boldsymbol{\varphi}_i\right)\right]} , \quad \forall\, i = r_1 + 1, \ldots, r. \tag{3.23}$$

The two boxed terms in (3.22)–(3.23) are the *VMS-ROM closure terms*, which have *fundamentally different roles*: The VMS-ROM closure term in (3.22) models the interaction between the large resolved ROM modes and the small resolved ROM modes; the VMS-ROM closure term in (3.23) models the interaction between the small resolved ROM modes and the unresolved ROM modes. The new 3S-DD-VMS-ROM framework allows *great flexibility* in choosing the *structure* of the two VMS-ROM closure terms. For the NSE, we can use the

following approximations:

$$\begin{aligned}(\boldsymbol{\tau}_L)_i &:= -\left[\left(\left(\boldsymbol{u}_R\cdot\nabla\right)\boldsymbol{u}_R\,,\boldsymbol{\varphi}_i\right)-\left(\left((\boldsymbol{u}_L+\boldsymbol{u}_S)\cdot\nabla\right)(\boldsymbol{u}_L+\boldsymbol{u}_S)\,,\boldsymbol{\varphi}_i\right)\right]\\ &= \left(\tilde{A}_L\,\boldsymbol{a}+\boldsymbol{a}^\top\tilde{B}_L\,\boldsymbol{a}\right)_i,\qquad \forall\, i=1,\ldots,r_1\,, \end{aligned}\tag{3.24}$$

$$\begin{aligned}(\boldsymbol{\tau}_S)_i &:= -\left[\left(\left(\boldsymbol{u}_R\cdot\nabla\right)\boldsymbol{u}_R\,,\boldsymbol{\varphi}_i\right)-\left(\left((\boldsymbol{u}_L+\boldsymbol{u}_S)\cdot\nabla\right)(\boldsymbol{u}_L+\boldsymbol{u}_S)\,,\boldsymbol{\varphi}_i\right)\right]\\ &= \left(\tilde{A}_S\,\boldsymbol{a}+\boldsymbol{a}^\top\tilde{B}_S\,\boldsymbol{a}\right)_i\qquad \forall\, i=r_1+1,\ldots,r, \end{aligned}\tag{3.25}$$

where $\tilde{A}_L\in\mathbb{R}^{r_1\times r}$, $\tilde{A}_S\in\mathbb{R}^{(r-r_1)\times r}$, $\tilde{B}_L\in\mathbb{R}^{r_1\times r\times r}$, and $\tilde{B}_S\in\mathbb{R}^{(r-r_1)\times r\times r}$. To determine the entries in $\tilde{A}_L,\tilde{A}_S,\tilde{B}_L$, and $\tilde{B}_S$, we solve two least squares problems:

$$\min_{\tilde{A}_L,\tilde{B}_L}\ \sum_{j=1}^{M}\left\|\boldsymbol{\tau}_L^{FOM}-\left(\tilde{A}_L\,\boldsymbol{a}^{FOM}(t_j)+\boldsymbol{a}^{FOM}(t_j)^\top\tilde{B}_L\,\boldsymbol{a}^{FOM}(t_j)\right)\right\|^2,\tag{3.26}$$

$$\min_{\tilde{A}_S,\tilde{B}_S}\ \sum_{j=1}^{M}\left\|\boldsymbol{\tau}_S^{FOM}-\left(\tilde{A}_S\,\boldsymbol{a}^{FOM}(t_j)+\boldsymbol{a}^{FOM}(t_j)^\top\tilde{B}_S\,\boldsymbol{a}^{FOM}(t_j)\right)\right\|^2,\tag{3.27}$$

where $\boldsymbol{\tau}_L^{FOM},\boldsymbol{\tau}_S^{FOM}$, and $\boldsymbol{a}^{FOM}$ are obtained from the available FOM data.

For the NSE, the *three-scale data-driven VMS-ROM (3S-DD-VMS-ROM)* is

$$\begin{bmatrix}\dot{\boldsymbol{a}}_L\\ \dot{\boldsymbol{a}}_S\end{bmatrix} = A\,\boldsymbol{a}+\boldsymbol{a}^\top B\,\boldsymbol{a}+\begin{bmatrix}\tilde{A}_L\,\boldsymbol{a}+\boldsymbol{a}^\top\tilde{B}_L\,\boldsymbol{a}\\ \tilde{A}_S\,\boldsymbol{a}+\boldsymbol{a}^\top\tilde{B}_S\,\boldsymbol{a}\end{bmatrix},\tag{3.28}$$

where $\boldsymbol{a}^\top=[\boldsymbol{a}_L,\boldsymbol{a}_S]^\top$, A and B are the G-ROM operators in (3.10), and $\tilde{A}_L,\tilde{A}_S,\tilde{B}_L$, and $\tilde{B}_S$ are the VMS-ROM closure operators constructed in (3.26)–(3.27). Compared to the 2S-DD-VMS-ROM, in the 3S-DD-VMS-ROM we have *more flexibility* in choosing the VMS-ROM closure operators $\tilde{A}_L,\tilde{A}_S,\tilde{B}_L$, and $\tilde{B}_S$ in the least squares problems (3.26)–(3.27). For example, for $\tilde{A}_L,\tilde{B}_L$ we can specify physical constraints, sparsity patterns, or regularization parameters, that are different from those for $\tilde{A}_S,\tilde{B}_S$. Because of this increased flexibility, we expect that the 3S-DD-VMS-ROM (3.28) is *more accurate* than the 2S-DD-VMS-ROM (3.20).

3.4 Numerical Results

In this section, we perform a numerical investigation of the new DD-VMS-ROM framework. As noted in Section 3.3, the 2S-DD-VMS-ROM (3.20) was investigated in [85] under the name "data-driven filtered ROM" and in [50, 52] under the name "data-driven correction ROM." In [85], it was shown that the 2S-DD-VMS-ROM is more accurate than the standard G-ROM in the numerical simulation of 2D flow past a circular cylinder at Reynolds numbers $Re=100, Re=500$, and $Re=1000$. Furthermore, the 2S-DD-VMS-ROM was more

accurate and more efficient than other modern ROM closure models. In [52], it was shown that the 2S-DD-VMS-ROM is more accurate than the standard G-ROM in the numerical simulation of the quasi-geostrophic equations modeling the large scale ocean circulation.

Since the 2S-DD-VMS-ROM has already been shown to perform well, the focus of the current numerical investigation is on the new 3S-DD-VMS-ROM (3.28). Specifically, we investigate whether the 3S-DD-VMS-ROM is more accurate than the 2S-DDC-ROM. To this end, we consider four test cases: (i) the 1D viscous Burgers equation with viscosity coefficient $\nu = 10^{-3}$ (Section 3.4.2); (ii) a 2D flow past a circular cylinder at Reynolds numbers $Re = 100$, $Re = 500$, and $Re = 1000$ (Section 3.4.3); (iii) the quasi-geostrophic equations at Reynolds number $Re = 450$ and Rossby number $Ro = 0.0036$ (Section 3.4.4); and (iv) a 2D flow over a backward facing step at Reynolds number $Re = 1000$ (Section 3.4.5). For each test case, we investigate three ROMs: the 2S-DD-VMS-ROM (3.20), the new 3S-DD-VMS-ROM (3.28), and (for comparison purposes) the standard G-ROM (3.10). As a benchmark, we use the FOM results.

We test the ROMs in three different regimes:

(i) *Reconstructive* regime: The ROM basis and ROM operators A and B are constructed from FOM data obtained on the time interval $[0, T_1]$, and then the resulting ROMs are tested on the *same* time interval $[0, T_1]$. To construct the DD-VMS-ROM operators $\tilde{A}$ and $\tilde{B}$ (for the 2S-DD-VMS-ROM) and $\tilde{A}_L, \tilde{A}_S, \tilde{B}_L$, and $\tilde{B}_S$ (for the 3S-DD-VMS-ROM), we use different approaches for the four test cases: For the Burgers equation, quasi-geostrophic equations, and backward facing step test cases, we construct the DD-VMS-ROM operators by using FOM data from the entire time interval $[0, T_1]$. For the flow past a circular cylinder test case, for computational efficiency, we construct the DD-VMS-ROM operators from FOM data obtained on a shorter time interval, which does not significantly decrease the accuracy of the resulting DD-VMS-ROM. Specifically, we use FOM data for one period [50, 85], i.e., (i) from $t = 7$ to $t = 7.332$ for $Re = 100$, (ii) from $t = 7$ to $t = 7.442$ for $Re = 500$, and (iii) from $t = 13$ to $t = 13.268$ for $Re = 1000$.

(ii) *Cross-validation* regime: The ROM basis and ROM operators A and B are constructed from FOM data obtained on the time interval $[0, T_2]$, and then the resulting ROMs are tested on the time interval $[0, T_3]$, where $T_3 > T_2$. We note that the two time intervals are different, but they do overlap over $[0, T_2]$.

To construct the DD-VMS-ROM operators, we use different approaches for the two test cases: For the Burgers equation test case, we construct the DD-VMS-ROM operators by using FOM data from the entire time interval $[0, T_2]$. For the flow past a circular cylinder test case, for computational efficiency, we construct the DD-VMS-ROM operators from FOM data for one period [50, 85].

(iii) *Predictive* regime: The ROM basis and ROM operators A and B are constructed from FOM data obtained on the time interval $[0, T_2]$, and then the resulting ROMs are tested on the time interval $[T_2, T_3]$, where $T_3 > T_2$. We emphasize that the two time intervals are

completely different, without any overlap. To construct the DD-VMS-ROM operators, we use different approaches for the two test cases: For the Burgers equation test case, we construct the DD-VMS-ROM operators by using FOM data from the entire time interval $[0, T_2]$. For the flow past a circular cylinder test case, for computational efficiency, we construct the DD-VMS-ROM operators from FOM data for half a period [50, 85].

3.4.1 Computational Setting

In this section, we present the computational setting used in the numerical investigation.

First, as explained in detail on page B843 of [85], we rewrite the optimization problem (3.17) as the least squares problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^{(r^2+r^3)\times 1}} \| \boldsymbol{f} - E\,\boldsymbol{x} \|^2 , \tag{3.29}$$

where $\boldsymbol{x} \in \mathbb{R}^{(r^2+r^3)\times 1}$ contains all the entries of $\tilde{A}$ and $\tilde{B}$, and the vector $\boldsymbol{f} \in \mathbb{R}^{(M\,r)\times 1}$ and matrix $E \in \mathbb{R}^{(M\,r)\times(r^2+r^3)}$ are computed from $\boldsymbol{u}_R^{FOM}, \boldsymbol{u}_r^{FOM}$, and $\boldsymbol{a}^{FOM}$ (see (4.8) in [85]). The optimal $\tilde{A}$ and $\tilde{B}$ (i.e., the entries in $\boldsymbol{x}$ that solves the linear least squares problem (3.29)) are used to build the 2S-DD-VMS-ROM (3.20).

Furthermore, as explained on page B843 of [85], the least squares problem (3.29) is ill-conditioned. This ill-conditioning is common in data-driven least squares problems (see, e.g., [59]). To alleviate this ill-conditioning, we use the truncated singular value decomposition (SVD) [50, 85].

The algorithm for the 2S-DD-VMS-ROM (3.20) is presented in Algorithm 2. In most of our numerical experiments, we choose the optimal tolerance tol in the truncated SVD step of Algorithm 2. Specifically, for each value $1 \le m \le R$ (where R is the dimension of the snapshot matrix), we consider the truncated SVD approximation of dimension m, construct the operators $\tilde{A}_m$ and $\tilde{B}_m$, integrate the resulting 2S-DD-VMS-ROM in (3.33), and choose the $\widetilde{m}$ value yielding the lowest L^2 error. The only exception is in some of the numerical experiments for the Burgers equation (Section 3.4.2), where we fix $tol = tol_L$ or $tol = tol_S$ (see Tables 3.2–3.8).

The algorithm for the 3S-DD-VMS-ROM (3.28) is the same as Algorithm 2, except that we are using two different truncated SVDs to solve two different linear least squares problems, which correspond to the large and small resolved scales. Thus, we have two different control parameters, tol_L and tol_S. Similar to the 2S-DD-VMS-ROM , we rewrite the optimization problems (3.26) and (3.27) as the least squares problems

$$\min_{\boldsymbol{x}_L\in\mathbb{R}^{\left[r_1(r+r^2)\right]\times 1}} \| \boldsymbol{f}_L - E_L\,\boldsymbol{x}_L \|^2 , \tag{3.35}$$

Algorithm 2 2S-DD-VMS-ROM

1: Use all the entries of $\tilde{A}$ and $\tilde{B}$ in (3.20) to define vector of unknowns, $\boldsymbol{x}$.
2: Use $\boldsymbol{u}_R^{FOM}, \boldsymbol{u}_r^{FOM}$, and $\boldsymbol{a}^{FOM}$ to assemble the vector $\boldsymbol{f}$ and matrix E in (3.29).
3: Use the truncated SVD algorithm to solve the linear least squares problem (3.29).

(i) Calculate the SVD of E:

$$E = U\,\Sigma V^{\top}\,, \tag{3.30}$$

where the rank of matrix $E(\Sigma)$ is $\mathcal{M}$.

(ii) Specify tolerance $tol = \sigma_i\,, i = 1, \cdots, \mathcal{M}$.

(iii) Construct matrix $\widehat{\Sigma}^m$ from Σ as follows: $\widehat{\sigma}_m = \sigma_m$ if $\sigma_m \geq tol$, $m = 1, \cdots, \mathcal{M}$.

(iv) Construct $\widehat{E}^m$, the truncated SVD of E:

$$\widehat{E}^m = \widehat{U}^m\,\widehat{\Sigma}^m\left(\widehat{V}^m\right)^{\top}, \tag{3.31}$$

where $\widehat{U}^m$ and $\widehat{V}^m$ are the entries of U and V in (3.30) that correspond to $\widehat{\Sigma}^m$.

(v) The solution of the least squares problem (3.29) is

$$\boldsymbol{x} = \left(\widehat{V}^m\left(\widehat{\Sigma}^m\right)^{-1}\left(\widehat{U}^m\right)^{\top}\right)\,\boldsymbol{f}\,. \tag{3.32}$$

4: The 2S-DD-VMS-ROM (3.20) has the following form:

$$\boxed{\dot{\boldsymbol{a}} = \left(A + \tilde{A}^m\right)\,\boldsymbol{a} + \boldsymbol{a}^{\top}\,\left(B + \tilde{B}^m\right)\,\boldsymbol{a}}\,, \tag{3.33}$$

where $\tilde{A}^m$ and $\tilde{B}^m$ are the appropriate entries of $\boldsymbol{x}$ found in (3.32) with $tol = \sigma_m$.

5: Integrate the resulting 2S-DD-VMS-ROM in (3.20) over the given time domain and calculate the average L^2 error $\mathcal{E}^m(L^2)$ by using formula (3.39). The optimal $\widetilde{m}$ value (the optimal operators $\tilde{A}$ and $\tilde{B}$) is found by solving the following minimization problem:

$$\mathcal{E}^{\widetilde{m}}(L^2) = \min_{1 \leq m \leq R} \mathcal{E}^m(L^2)\,. \tag{3.34}$$

$$\min_{\boldsymbol{x} \in \mathbb{R}^{\left[(r-r_1)(r+r^2)\right]\times 1}} \|\,\boldsymbol{f}_S - E_S\,\boldsymbol{x}_S\,\|^2\,, \tag{3.36}$$

where the vectors $\boldsymbol{f}_L \in \mathbb{R}^{(M\,r_1)\times 1}$, $\boldsymbol{f}_S \in \mathbb{R}^{(M\,(r-r_1))\times 1}$, and the matrices $E_L \in \mathbb{R}^{(M\,r_1)\times(r_1(r+r^2))}$, $E_S \in \mathbb{R}^{(M\,(r-r_1))\times((r-r_1)(r+r^2))}$ are computed from $\boldsymbol{u}_R^{FOM}, \boldsymbol{u}_r^{FOM}$, and $\boldsymbol{a}^{FOM}$ (see (4.8) in [85]). Furthermore, $\boldsymbol{x}_L \in \mathbb{R}^{\left[r_1(r+r^2)\right]\times 1}$ contains all the entries of the operators $\tilde{A}_L$ and $\tilde{B}_L$, $\boldsymbol{x}_S \in \mathbb{R}^{\left[(r-r_1)(r+r^2)\right]\times 1}$ contains all the entries of the operators $\tilde{A}_S$ and $\tilde{B}_S$. The optimal $\tilde{A}_L$, $\tilde{B}_L$ and $\tilde{A}_S$, $\tilde{B}_S$ (i.e., the entries in $\boldsymbol{x}_L$ and $\boldsymbol{x}_S$ that solve the linear least squares problems (3.35) and

(3.36)) are used to build the 3S-DD-VMS-ROM (3.28). Again, to address the ill-conditioning of the least squares problems (3.35)–(3.36), we use the truncated SVD algorithm. The algorithm for the 3S-DD-VMS-ROM (3.28) is presented in Algorithm 3. We note that, if $tol_L = tol_S = tol$, the 2S-DD-VMS-ROM and 3S-DD-VMS-ROM yield the same results, since we solve the same minimization problem. Thus, the interesting case is when tol_L and/or tol_S are different from tol.

Algorithm 3 3S-DD-VMS-ROM

1: Choose $r_1, 1 \leq r_1 < r$, and use all the entries of $\tilde{A}_L$ and $\tilde{B}_L$ as well as $\tilde{A}_S$ and $\tilde{B}_S$ in (3.28) to define vectors of unknowns, $\boldsymbol{x}_L$ and $\boldsymbol{x}_S$, respectively.

2: Use $\boldsymbol{u}_R^{FOM}, \boldsymbol{u}_r^{FOM}$, and $\boldsymbol{a}^{FOM}$ to assemble the vectors $\boldsymbol{f}_L$ and $\boldsymbol{f}_S$, and the matrices E_L and E_S in (3.35) and (3.36).

3: Use the truncated SVD algorithm to solve the linear least squares problems (3.35) and (3.36).

(i) Calculate the SVD of E_L and E_S:

$$E_L = U_L \Sigma_L V_L^\top , \quad E_S = U_S \Sigma_S V_S^\top . \tag{3.37}$$

(ii) Specify tolerances $tol_L = \sigma_{L,i}, i = 1, \cdots, \mathcal{M}_L$, and $tol_S = \sigma_{S,j}, j = 1, \cdots, \mathcal{M}_S$, where $\mathcal{M}_L$ is the rank of Σ_L (E_L) and $\mathcal{M}_S$ is the rank of Σ_S (E_S).

(iii) Construct matrix $\widehat{\Sigma}_L^m$ from Σ_L as follows: $\widehat{\sigma}_{L,m_L} = \sigma_{m_L}$ if $\sigma_{m_L} \geq tol_L$, $m_L = 1, \cdots, \mathcal{M}_L$; construct matrix $\widehat{\Sigma}_L^m$ from Σ_L as follows: $\widehat{\sigma}_{S,m_L} = \sigma_{m_S}$ if $\sigma_{m_S} \geq tol_S$, $m_S = 1, \cdots, \mathcal{M}_S$.

(iv) Construct $\widehat{E}_L^{m_L}$ and $\widehat{E}_S^{m_S}$ with the truncated SVD of E_L and E_S.

(v) Construct the operators $\tilde{A}_L^{m_L}$ and $\tilde{B}_L^{m_L}$ as well as $\tilde{A}_S^{m_S}$ and $\tilde{B}_S^{m_S}$.

(vi) Integrate the resulting 3S-DD-VMS-ROM in (3.28) over the given time domain and calculate the average L^2 error $\mathcal{E}^{r_1,m_L,m_S}(L^2)$ by using formula (3.39).

4: Find the optimal $\widetilde{r}_1$, $\widetilde{m}_L$ and $\widetilde{m}_S$ values (i.e., the optimal operators $\tilde{A}_L, \tilde{B}_L$ and $\tilde{A}_S, \tilde{B}_S$ corresponding to the optimal r_1) by solving the following minimization problem:

$$\mathcal{E}^{\widetilde{r}_1,\widetilde{m}_L,\widetilde{m}_S}(L^2) = \min_{\substack{1\leq r_1<r \\ 1\leq m_L\leq \mathcal{M}_L \\ 1\leq m_S\leq \mathcal{M}_S}} \mathcal{E}^{r_1,m_L,m_S}(L^2) . \tag{3.38}$$

The focus of the current numerical investigation is on the numerical accuracy of the new DD-VMS-ROMs. Thus, we use all the available data to build the DD-VMS-ROM operators. We emphasize, however, that the computational cost of the construction of the DD-VMS-ROM operators can be significantly decreased by using the approach proposed on page B848

in [85].

To compare the ROMs' performance, in the Burgers equation, flow past a circular cylinder, and backward facing step test cases, we use the error metric

$$\text{average } L^2 \text{ norm: } \mathcal{E}(L^2) = \frac{1}{M}\sum_{j=1}^{M}\left\|\boldsymbol{u}_r(t_j) - \sum_{i=1}^{r}\left(\boldsymbol{u}^{\text{FOM}}(t_j),\varphi_i\right)\varphi_i\right\|_{L^2}, \tag{3.39}$$

whereas in the quasi-geostrophic equations test case we use the error metric (3.48). In the flow past a circular cylinder, quasi-geostrophic equations, and flow over a backward facing step test cases, we plot the time evolution of the ROM kinetic energy. Furthermore, in the quasi-geostrophic equations test case, we use the L^2 error of the time-averaged streamfunction, and plot the time-averaged streamfunction. Finally, in the backward facing step test case, we plot the time evolution of the y-component of the velocity, and the spectrum of the y-component of the velocity at a control point.

3.4.2 Burgers Equation

In this section, we investigate the 2S-DD-VMS-ROM (3.20) and the new 3S-DD-VMS-ROM (3.28) in the numerical simulation of the one-dimensional viscous Burgers equation:

$$\begin{cases} u_t - \nu u_{xx} + u u_x = 0\ , \quad x \in [0,1],\ t \in [0,1], \\ u(0,t) = u(1,t) = 0\ , \quad t \in [0,1], \end{cases} \tag{3.40}$$

with the initial condition

$$u_0(x) = \begin{cases} 1, & x \in (0,1/2], \\ 0, & x \in (1/2,1], \end{cases} \tag{3.41}$$

and $\nu = 10^{-3}$. This test problem has been used in [1, 36, 42, 85].

Snapshot Generation We generate the FOM results by using a linear FE spatial discretization with mesh size $h = 1/2048$ and a Crank-Nicolson time discretization with timestep size $\Delta t = 10^{-3}$.

ROM Construction We run the FOM from $t = 0$ to $t = 1$. To generate the ROM basis functions, we collect a total of 1000 snapshots for the reconstructive regime, and 700 snapshots for the cross-validation and predictive regimes. These snapshots are the solutions from $t = 0$ to $t = 1$ for the reconstructive regime, and $t = 0$ to $t = 0.7$ for the cross-validation and predictive regimes. To train $\tilde{A}$, $\tilde{B}$ (for the 2S-DD-VMS-ROM) and $\tilde{A}_L$, $\tilde{B}_L$ and $\tilde{A}_S$, $\tilde{B}_S$ (for the 3S-DD-VMS-ROM), we use FOM data on the time interval $[0,1]$ for the reconstructive regime, and FOM data on the time interval $[0,0.7]$ for the cross-validation and predictive regimes. We test all the ROMs on the time interval $[0,1]$ for the reconstructive and cross-validation regimes, and $[0.7,1]$ for the predictive regime.

Implementation Details To implement the 2S-DD-VMS-ROM (3.33), we use Algorithm 2. To implement the 3S-DD-VMS-ROM (3.28), we use Algorithm 3. For a fair comparison of the 2S-DD-VMS-ROM with the 3S-DD-VMS-ROM, we choose optimal tolerances in the two algorithms, i.e., optimal tol in Algorithm 2 and optimal tol_L and tol_s in Algorithm 3. We also investigate whether there is any relationship between the 2S-DD-VMS-ROM tolerance and the 3S-DD-VMS-ROM tolerances. To this end, we perform two sets of numerical experiments: (a) In the first set of experiments, we fix tol in the 2S-DD-VMS-ROM, choose $tol_L = tol$ in the 3S-DD-VMS-ROM, and search the optimal tol_S in the 3S-DD-VMS-ROM. (b) In the second set of experiments, we fix tol in the 2S-DD-VMS-ROM, choose $tol_S = tol$ in the 3S-DD-VMS-ROM, and search the optimal tol_L in the 3S-DD-VMS-ROM.

Numerical Results

In this section, we present numerical results for the Burgers equation (3.40) with $\nu = 10^{-3}$ in the reconstructive, cross-validation, and predictive regimes. In all the tables, we list the average L^2 error (3.39) for the G-ROM, the 2S-DD-VMS-ROM, and the new 3S-DD-VMS-ROM. We also list the tolerances used in the truncated SVD algorithm for the 2S-DD-VMS-ROM and 3S-DD-VMS-ROM, as well as the r_1 values for the 3S-DD-VMS-ROM.

In Table 3.1, we list the ROMs errors for the reconstructive regime with optimal tol in Algorithm 2 and optimal tol_L and tol_s in Algorithm 3. These results show that, for all r values, the 2S-DD-VMS-ROM and 3S-DD-VMS-ROM are several times (sometimes one or even two orders of magnitude) more accurate than the standard G-ROM. Overall, the 3S-DD-VMS-ROM is more accurate than the 2S-DD-VMS-ROM. For example, for $r = 7$, the 3S-DD-VMS-ROM is more than twice more accurate than the 2S-DD-VMS-ROM. We also note that, for low r values, the ROM errors do not seem to converge monotonically. We emphasize, however, that for large r values, we recover the expected asymptotic convergence.

r	G-ROM	2S-DD-VMS-ROM		3S-DD-VMS-ROM			
	$\mathcal{E}(L^2)$	tol	$\mathcal{E}(L^2)$	r_1	tol_S	tol_L	$\mathcal{E}(L^2)$
2	1.018e-01	1e-02	2.110e-03	1	1e-02	1e-02	2.110e-03
3	1.181e-01	1e-02	1.548e-03	1	1e-02	1e-02	1.548e-03
4	1.382e-01	1e-03	1.845e-03	1	1e-02	1e-03	1.162e-03
5	1.698e-01	7e-02	2.889e-03	4	1e-02	5e-04	1.577e-03
7	1.828e-01	1e-04	3.542e-03	4	1e-02	1e-04	1.688e-03
11	1.258e-01	1e-02	2.213e-03	4	1e-02	1e-04	1.675e-03
17	6.551e-02	1e-02	2.312e-03	4	1e-02	1e-04	1.971e-03

Table 3.1: Burgers equation, $\nu = 10^{-3}$, reconstructive regime, optimal tol, tol_S, and tol_L. Average L^2 error for G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values.

In Figure 3.1, we plot the time evolution of the solutions for the FOM projection, G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for the reconstructive regime. These plots show that both the 2S-DD-VMS-ROM and the 3S-DD-VMS-ROM are significantly more accurate than the standard G-ROM, as indicated by the results in Table 3.1.

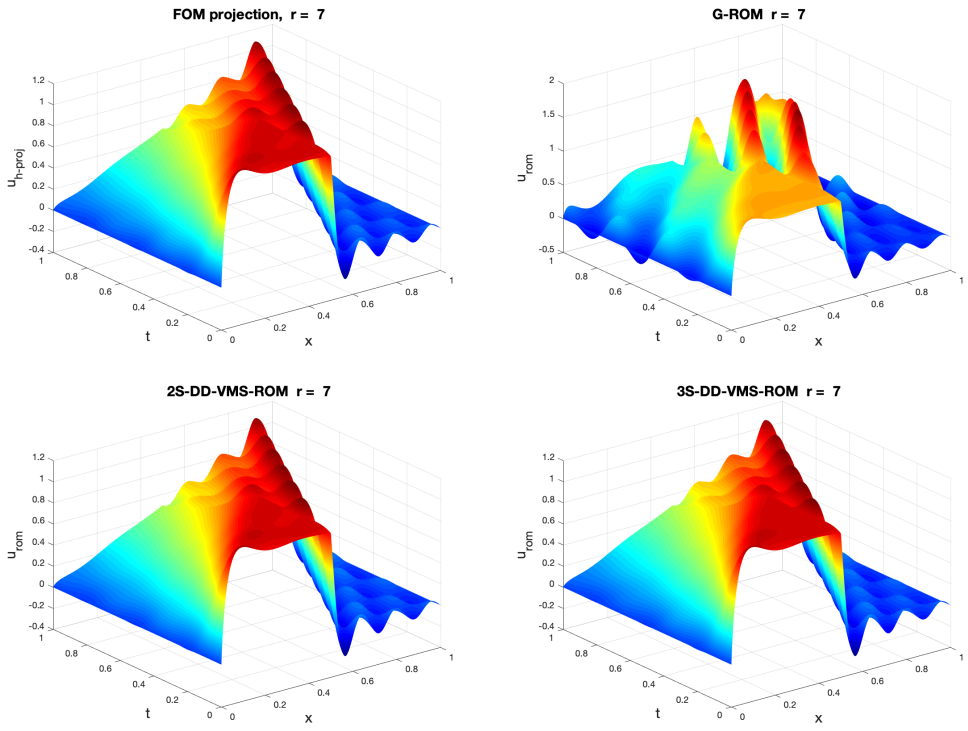

Figure 3.1: Burgers equation, $\nu = 10^{-3}$, reconstructive regime. FOM projection, G-ROM, 2S-DD-VMS-DDC-ROM, and 3S-DD-VMS-DDC-ROM plots for $r = 7$.

In Tables 3.2, 3.3, 3.4, and 3.5, we list the ROMs errors for the reconstructive regime with fixed tol in the 2S-DD-VMS-ROM, and $tol_L = tol$ and optimal tol_S in the 3S-DD-VMS-ROM. We also list the optimal value of tol_S. We consider the following values for $tol_L = tol$: 10^2 (Table 3.2), 10^1 (Table 3.3), 10^0 (Table 3.4), and 10^{-1} (Table 3.5). These results yield the following conclusions: For large $tol_L = tol$ values (i.e., 10^2 and 10^1), the 2S-DD-VMS-ROM is slightly more or as accurate as the G-ROM, whereas the 3S-DD-VMS-ROM is several times (and sometimes more than one order of magnitude) more accurate than the G-ROM and 2S-DD-VMS-ROM. For small $tol_L = tol$ values (i.e., 10^0 and 10^{-1}), the 2S-DD-VMS-ROM is several times (and sometimes more than one order of magnitude) more accurate than the G-ROM. Even in these cases, however, the 3S-DD-VMS-ROM is several times (and sometimes more than one order of magnitude) more accurate than the 2S-DD-VMS-ROM. Overall, the 3S-DD-VMS-ROM is by far the most accurate ROM.

r	G-ROM	2S-DD-VMS-ROM	3S-DD-VMS-ROM		
	$\mathcal{E}(L^2)$	$\mathcal{E}(L^2)$	r_1	tol_S	$\mathcal{E}(L^2)$
3	1.181e-01	1.181e-01	1	1e+00	1.609e-02
7	1.828e-01	1.828e-01	1	1e-01	6.241e-03
11	1.258e-01	1.258e-01	1	1e-01	4.955e-03
17	6.551e-02	6.551e-02	1	1e-02	2.826e-03

Table 3.2: Burgers equation, $\nu = 10^{-3}$, reconstructive regime, $tol = tol_L = 10^2$, and optimal tol_S. Average L^2 error for G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values.

r	G-ROM	2S-DD-VMS-ROM	3S-DD-VMS-ROM		
	$\mathcal{E}(L^2)$	$\mathcal{E}(L^2)$	r_1	tol_S	$\mathcal{E}(L^2)$
3	1.181e-01	7.278e-02	1	1e+00	1.322e-02
7	1.828e-01	1.755e-01	2	1e-03	3.915e-03
11	1.258e-01	1.229e-01	1	1e-03	1.787e-03
17	6.551e-02	6.456e-02	1	1e-02	2.310e-03

Table 3.3: Burgers equation, $\nu = 10^{-3}$, reconstructive regime, $tol = tol_L = 10^1$, and optimal tol_S. Average L^2 error for G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values.

r	G-ROM	2S-DD-VMS-ROM	3S-DD-VMS-ROM		
	$\mathcal{E}(L^2)$	$\mathcal{E}(L^2)$	r_1	tol_S	$\mathcal{E}(L^2)$
3	1.181e-01	1.333e-01	1	1e-02	5.292e-03
7	1.828e-01	2.590e-02	2	1e-03	3.549e-03
11	1.258e-01	3.607e-02	2	1e-02	2.045e-03
17	6.551e-02	5.029e-02	5	1e-02	2.237e-03

Table 3.4: Burgers equation, $\nu = 10^{-3}$, reconstructive regime, $tol = tol_L = 10^0$, and optimal tol_S. Average L^2 error for G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values.

In Tables 3.6, 3.7, and 3.8, we list the ROMs errors for the reconstructive regime with fixed tol in the 2S-DD-VMS-ROM, and $tol_S = tol$ and optimal tol_L in the 3S-DD-VMS-ROM. We also list the optimal value of tol_L. We consider the following values for $tol_S = tol$: 10^0 (Table 3.6), 10^{-1} (Table 3.7), and 10^{-2} (Table 3.8). These results yield the following conclusions: For all $tol_L = tol$ values and all r values, the 2S-DD-VMS-ROM is several times (and sometimes more than one order of magnitude) more accurate than the G-ROM. Furthermore, the 3S-DD-VMS-ROM is significantly (and sometimes several times) more

r	G-ROM	2S-DD-VMS-ROM	3S-DD-VMS-ROM		
	$\mathcal{E}(L^2)$	$\mathcal{E}(L^2)$	r_1	tol_S	$\mathcal{E}(L^2)$
3	1.181e-01	3.729e-03	1	1e-02	2.061e-03
7	1.828e-01	4.232e-03	4	1e-03	2.557e-03
11	1.258e-01	4.556e-03	2	1e-02	2.086e-03
17	6.551e-02	5.962e-03	5	1e-02	2.255e-03

Table 3.5: Burgers equation, $\nu = 10^{-3}$, reconstructive regime $tol = tol_L = 10^{-1}$, and optimal tol_S. Average L^2 error for G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values.

accurate than the 2S-DD-VMS-ROM. Overall, the 3S-DD-VMS-ROM is the most accurate ROM.

r	G-ROM	2S-DD-VMS-ROM	3S-DD-VMS-ROM		
	$\mathcal{E}(L^2)$	$\mathcal{E}(L^2)$	r_1	tol_L	$\mathcal{E}(L^2)$
3	1.181e-01	1.133e-02	2	1e-01	8.085e-03
7	1.828e-01	2.590e-02	6	1e-03	1.762e-02
11	1.258e-01	3.607e-02	10	1e-02	2.390e-02
17	6.551e-02	5.029e-02	16	1e-02	1.486e-02

Table 3.6: Burgers equation, $\nu = 10^{-3}$, reconstructive regime: $tol = tol_S = 10^{0}$ and optimal tol_L. Average L^2 error for G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values.

r	G-ROM	2S-DD-VMS-ROM	3S-DD-VMS-ROM		
	$\mathcal{E}(L^2)$	$\mathcal{E}(L^2)$	r_1	tol_L	$\mathcal{E}(L^2)$
3	1.181e-01	3.729e-03	2	1e-02	2.568e-03
7	1.828e-01	4.232e-03	4	1e-03	3.678e-03
11	1.258e-01	4.556e-03	10	1e-02	3.995e-03
17	6.551e-02	5.962e-03	16	1e-02	2.995e-03

Table 3.7: Burgers equation, $\nu = 10^{-3}$, reconstructive regime: $tol = tol_S = 10^{-1}$ and optimal tol_L. Average L^2 error for G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values.

r	G-ROM	2S-DD-VMS-ROM	3S-DD-VMS-ROM		
	$\mathcal{E}(L^2)$	$\mathcal{E}(L^2)$	r_1	tol_L	$\mathcal{E}(L^2)$
3	1.181e-01	1.548e-03	1	1e-02	1.548e-03
7	1.828e-01	1.062e-02	4	1e-04	1.688e-03
11	1.258e-01	2.213e-03	4	1e-04	1.675e-03
17	6.551e-02	2.312e-03	4	1e-04	1.974e-03

Table 3.8: Burgers equation, $\nu = 10^{-3}$, reconstructive regime: $tol = tol_S = 10^{-2}$ and optimal tol_L. Average L^2 error for G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values.

The results in Tables 3.2–3.8 suggest that there is no apparent relationship between the 2S-DD-VMS-ROM tolerance tol and the 3S-DD-VMS-ROM tolerances tol_L and tol_S. We intend to perform a more thorough investigation of potential relationships among these tolerances in a future study.

In Table 3.9, we list the ROMs errors for the cross-validation regime with optimal tol in Algorithm 2 and optimal tol_L and tol_s in Algorithm 3. These results show that, for all r values, the 2S-DD-VMS-ROM and 3S-DD-VMS-ROM are several times (sometimes even one order of magnitude) more accurate than the standard G-ROM. Overall, the 3S-DD-VMS-ROM is more accurate than the 2S-DD-VMS-ROM.

r	G-ROM	2S-DD-VMS-ROM		3S-DD-VMS-ROM			
	$\mathcal{E}(L^2)$	tol	$\mathcal{E}(L^2)$	r_1	tol_S	tol_L	$\mathcal{E}(L^2)$
3	2.015e-01	1e-01	2.028e-02	2	1e-01	1e+00	1.863e-02
7	1.796e-01	5e-02	1.400e-02	3	5e-02	1e+00	1.188e-02
11	1.163e-01	3e-02	8.981e-03	6	3e-02	1e+00	8.383e-03
17	6.897e-02	1e-02	8.542e-03	6	1e-02	1e+00	8.452e-03

Table 3.9: Burgers equation, $\nu = 10^{-3}$, cross-validation regime, optimal tol, tol_S, and tol_L. Average L^2 error for G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values.

In Table 3.10, we list the ROMs errors for the predictive regime with optimal tol in Algorithm 2 and optimal tol_L and tol_s in Algorithm 3. These results show that, for all r values, the 2S-DD-VMS-ROM and 3S-DD-VMS-ROM are several times (sometimes even one order of magnitude) more accurate than the standard G-ROM. Overall, the 3S-DD-VMS-ROM is more accurate than the 2S-DD-VMS-ROM.

r	G-ROM	2S-DD-VMS-ROM		3S-DD-VMS-ROM			
	$\mathcal{E}(L^2)$	tol	$\mathcal{E}(L^2)$	r_1	tol_S	tol_L	$\mathcal{E}(L^2)$
3	2.185e-01	1e-01	3.623e-02	2	1e-01	1e+00	3.029e-02
7	2.054e-01	3e-02	2.004e-02	6	5e-02	3e-02	1.428e-02
11	1.620e-01	3e-02	1.608e-02	10	5e-02	3e-02	1.418e-02
17	1.103e-01	1e-02	1.524e-02	6	1e-02	1e-01	1.506e-02

Table 3.10: Burgers equation, $\nu = 10^{-3}$, predictive regime, optimal tol, tol_S, and tol_L. Average L^2 error for G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values.

3.4.3 Flow Past A Cylinder

In this section, we investigate the 2S-DD-VMS-ROM (3.20) and the new 3S-DD-VMS-ROM (3.28) in the numerical simulation of a 2D channel flow past a circular cylinder at Reynolds numbers $Re = 100, Re = 500$, and $Re = 1000$.

Computational Setting As a mathematical model, we use the NSE (3.8)–(3.9). The computational domain is a 2.2×0.41 rectangular channel with a radius $= 0.05$ cylinder, centered at $(0.2, 0.2)$, see Figure 5.2.

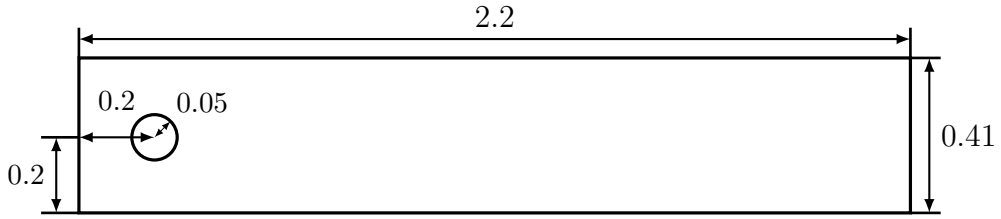

Figure 3.2: Geometry of the flow past a circular cylinder numerical experiment.

We prescribe no-slip boundary conditions on the walls and cylinder, and the following inflow and outflow profiles [38, 50, 51, 65]:

$$u_1(0, y, t) = u_1(2.2, y, t) = \frac{6}{0.41^2} y(0.41 - y), \tag{3.42}$$

$$u_2(0, y, t) = u_2(2.2, y, t) = 0, \tag{3.43}$$

where $\boldsymbol{u} = \langle u_1, u_2 \rangle$. There is no forcing and the flow starts from rest.

Snapshot Generation For the spatial discretization, we use the pointwise divergence-free, LBB stable (P_2, P_1^{disc}) Scott-Vogelius finite element pair on a barycenter refined regular

triangular mesh [40]. The mesh provides $103K$ (102962) velocity and $76K$ (76725) pressure degrees of freedom. We utilize the commonly used linearized BDF2 temporal discretization and a time step size $\Delta t = 0.002$ for both FOM and ROM time discretizations. On the first time step, we use a backward Euler scheme so that we have two initial time step solutions required for the BDF2 scheme.

ROM Construction The FOM simulations achieve the statistically steady state at different time instances for the three Reynolds numbers used in the numerical investigation: For $Re = 100$, after $t = 5s$; for $Re = 500$, after $t = 7s$; and for $Re = 1000$, after $t = 13s$. To build the ROM basis functions, we decided to use $10s$ of FOM data. Thus, to ensure a fair comparison of the numerical results at different Reynolds numbers, we collect FOM snapshots on the following time intervals: For $Re = 100$, from $t = 7$ to $t = 17$; for $Re = 500$, from $t = 7$ to $t = 17$; and for $Re = 1000$, from $t = 13$ to $t = 23$.

To train $\tilde{A}, \tilde{B}$ (for the 2S-DD-VMS-ROM) and $\tilde{A}_L, \tilde{B}_L$ and $\tilde{A}_S, \tilde{B}_S$ (for the 3S-DD-VMS-ROM), we use FOM data for one period in the reconstructive and cross-validation regimes, and FOM data for half a period in the predictive regime. We note that the period length of the statistically steady state is different for the three different Reynolds numbers: From $t = 7$ to $t = 7.332$ for $Re = 100$; from $t = 7$ to $t = 7.442$ for $Re = 500$; and from $t = 13$ to $t = 13.268$ for $Re = 1000$. Thus, the reconstructive and cross-validation regimes, we collect 167 snapshots for $Re = 100$; 222 snapshots for $Re = 500$; and 135 snapshots for $Re = 1000$. For the predictive regime, we collect 84 snapshots for $Re = 100$; 111 snapshots for $Re = 500$; and 68 snapshots for $Re = 1000$.

Numerical Results for $Re = 100$

In this section, we present numerical results for the flow past a cylinder at $Re = 100$.

In Table 3.11, for different r values, we list the average L^2 error (3.39) for the G-ROM, the 2S-DD-VMS-ROM, and the new 3S-DD-VMS-ROM in the reconstructive regime. We also list the r_1 values for the 3S-DD-VMS-ROM. These results show that, for all r values, the 2S-DD-VMS-ROM and 3S-DD-VMS-ROM are several times (sometimes one and even two orders of magnitude) more accurate than the standard G-ROM. Furthermore, the 3S-DD-VMS-ROM is generally more accurate than the 2S-DD-VMS-ROM, especially for large r values: For example, for $r = 8$, the 3S-DD-VMS-ROM is *more than twice more accurate* than the 2S-DD-VMS-ROM. We also note that the ROM errors in Table 3.11 converge to 0 according to an even/odd pattern: The ROM errors for even r values converge to 0 and the ROM errors for odd r values also converge to 0. This behavior is related to the flow past a cylinder configuration, in which the ROM modes appear in pairs. We emphasize, however, that for large r values, we recover the asymptotic convergence that does not depend on the odd/even r values, just as in the Burgers equation test case (Section 3.4.2).

r	G-ROM	2S-DD-VMS-ROM	3S-DD-VMS-ROM	
	$\mathcal{E}(L^2)$	$\mathcal{E}(L^2)$	r_1	$\mathcal{E}(L^2)$
2	9.902e-02	5.118e-04	1	5.088e-04
3	1.029e-01	3.208e-02	2	3.018e-02
4	5.840e-02	1.553e-03	2	1.479e-03
5	6.492e-02	2.270e-02	4	2.191e-02
6	1.370e-02	5.336e-04	1	4.804e-04
7	1.403e-02	6.038e-03	6	5.817e-03
8	1.214e-02	9.302e-04	6	4.415e-04

Table 3.11: Flow past a cylinder, $Re = 100$, reconstructive regime. Average L^2 errors for G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values.

In Table 3.12, for different r values, we list the average L^2 error (3.39) for the G-ROM, the 2S-DD-VMS-ROM, and the new 3S-DD-VMS-ROM in the cross-validation regime. These results show that, for all r values, the 2S-DD-VMS-ROM and 3S-DD-VMS-ROM are several times (sometimes even two orders of magnitude) more accurate than the standard G-ROM. Furthermore, the 3S-DD-VMS-ROM is generally more accurate than the 2S-DD-VMS-ROM, especially for large r values: For example, for $r = 8$, the 3S-DD-VMS-ROM is almost *three times more accurate* than the 2S-DD-VMS-ROM.

r	G-ROM	2S-DD-VMS-ROM	3S-DD-VMS-ROM	
	$\mathcal{E}(L^2)$	$\mathcal{E}(L^2)$	r_1	$\mathcal{E}(L^2)$
2	4.891e-01	1.536e-03	1	1.458e-03
3	4.088e-01	3.514e-02	2	3.106e-02
4	9.291e-02	2.187e-03	2	2.015e-03
5	1.013e-01	2.279e-02	4	2.220e-02
6	3.270e-02	5.113e-04	2	4.921e-04
7	3.059e-02	7.476e-03	1	7.260e-03
8	3.600e-02	1.221e-03	6	4.385e-04

Table 3.12: Flow past a cylinder, $Re = 100$, cross-validation regime. Average L^2 errors for G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values.

In Table 3.13, for different r values, we list the average L^2 error (3.39) for the G-ROM, the 2S-DD-VMS-ROM, and the new 3S-DD-VMS-ROM in the predictive regime. These results show that, for all r values, the 2S-DD-VMS-ROM and 3S-DD-VMS-ROM are several times (sometimes even one order of magnitude) more accurate than the standard G-ROM. Furthermore, the 3S-DD-VMS-ROM is generally more accurate than the 2S-DD-VMS-ROM: Specifically, for $r \geq 6$, the 3S-DD-VMS-ROM is at least *twice more accurate* than the 2S-DD-VMS-ROM.

r	G-ROM	2S-DD-VMS-ROM	3S-DD-VMS-ROM	
	$\mathcal{E}(L^2)$	$\mathcal{E}(L^2)$	r_1	$\mathcal{E}(L^2)$
2	3.883e-01	8.172e-02	1	7.388e-02
3	3.616e-01	3.374e-02	1	3.141e-02
4	1.366e-01	8.127e-03	2	4.115e-03
5	1.464e-01	4.248e-02	3	2.602e-02
6	1.348e-01	5.946e-03	4	1.051e-03
7	1.291e-01	1.529e-02	3	6.613e-03
8	9.638e-02	6.798e-03	6	3.170e-03

Table 3.13: Flow past a cylinder, $Re = 100$, predictive regime. Average L^2 errors for G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values.

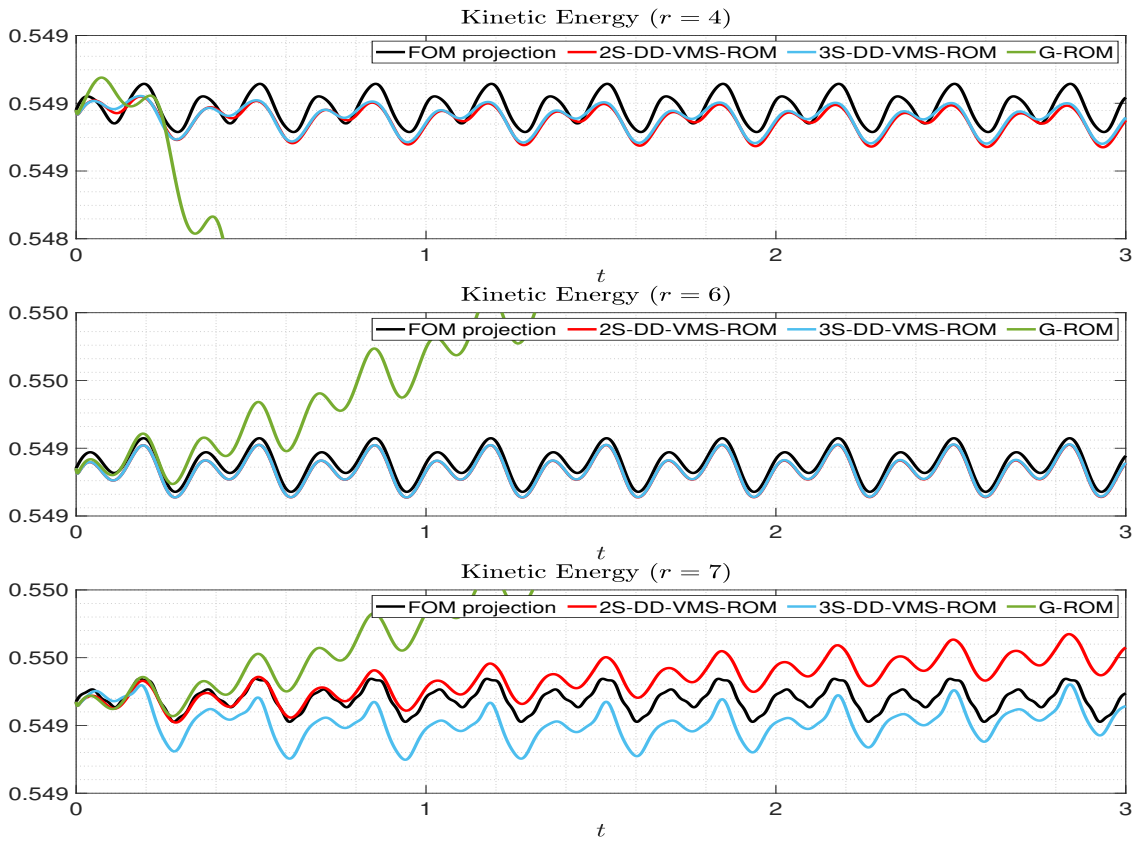

Figure 3.3: Flow past a cylinder, $Re = 100$, reconstructive regime. Time evolution of the kinetic energy for FOM projection, G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values.

In Figure 3.3, for $r = 4, 6, 7$, we plot the time evolution of the kinetic energy of the FOM,

the G-ROM, the 2S-DD-VMS-ROM, and the new 3S-DD-VMS-ROM in the reconstructive regime. These plots support the conclusions in Table 3.11: Both the 3S-DD-VMS-ROM and the 2S-DD-VMS-ROM accurately approximate the FOM kinetic energy and are significantly more accurate than the standard G-ROM. Furthermore, 3S-DD-VMS-ROM is slightly more accurate than the 2S-DD-VMS-ROM, especially for $r = 7$.

In Figure 3.4, for $r = 4, 6, 7$, we plot the time evolution of the kinetic energy of the FOM the G-ROM, the 2S-DD-VMS-ROM, and the new 3S-DD-VMS-ROM in the cross-validation regime. For all cases, the evolution of the G-ROM kinetic energy is very inaccurate. In contrast, for $r = 4$ and $r = 6$, both the 3S-DD-VMS-ROM and the 2S-DD-VMS-ROM successfully reproduce the FOM kinetic energy. For $r = 7$, the 3S-DD-VMS-ROM accurately approximates the FOM kinetic energy between $t = 0$ and $t = 4$. For $t \geq 4$, although the 3S-DD-VMS-ROM and 2S-DD-VMS-ROM kinetic energy approximations are not as accurate, they are still much more accurate than the G-ROM kinetic energy approximation.

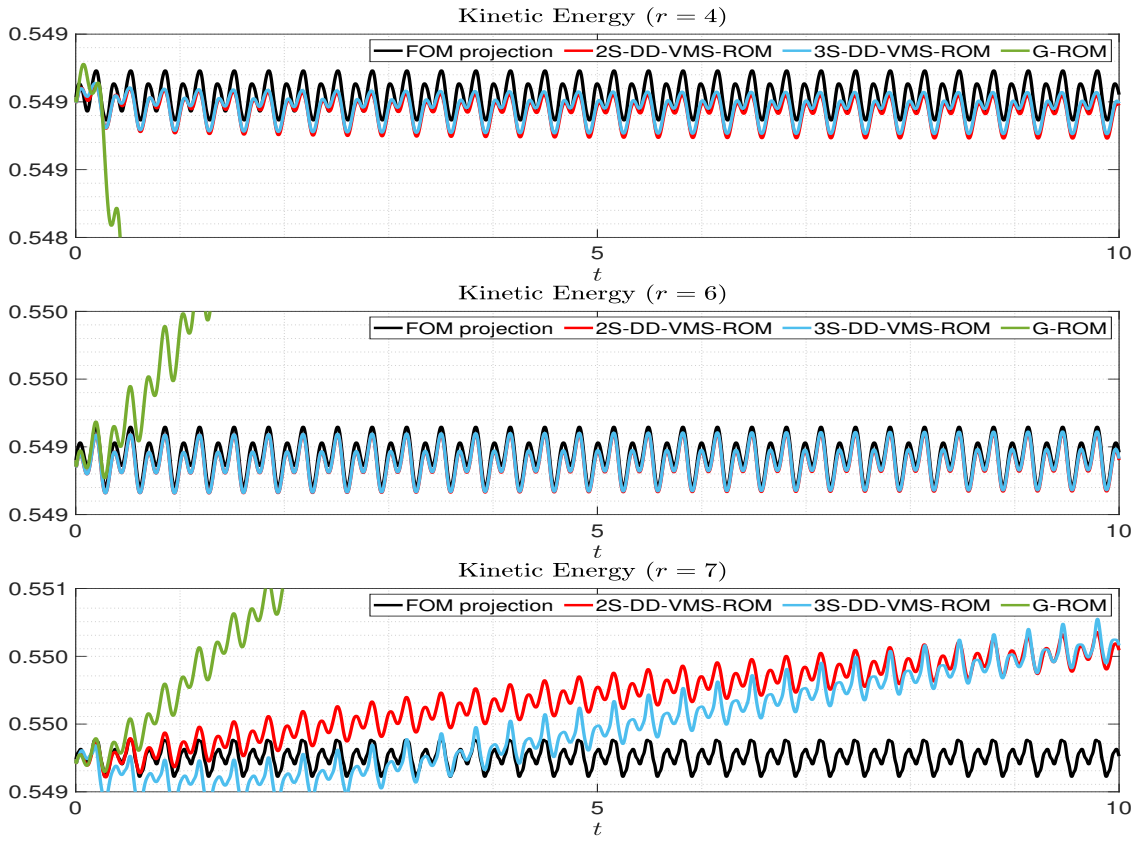

Figure 3.4: Flow past a cylinder, $Re = 100$, cross-validation regime. Time evolution of the kinetic energy for FOM projection, G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values.

In Figure 3.5, for $r = 4, 6, 7$, we plot the time evolution of the kinetic energy of the FOM, the G-ROM, the 2S-DD-VMS-ROM, and the new 3S-DD-VMS-ROM in the predictive regime. For all r values, the G-ROM kinetic energy approximation is very inaccurate. In contrast, the new 3S-DD-VMS-ROM accurately approximates the exact FOM kinetic energy for $r = 4, 6, 7$. The 2S-DD-VMS-ROM kinetic energy approximation is accurate for $r = 6$, but not for $r = 4$ and, especially, for $r = 7$.

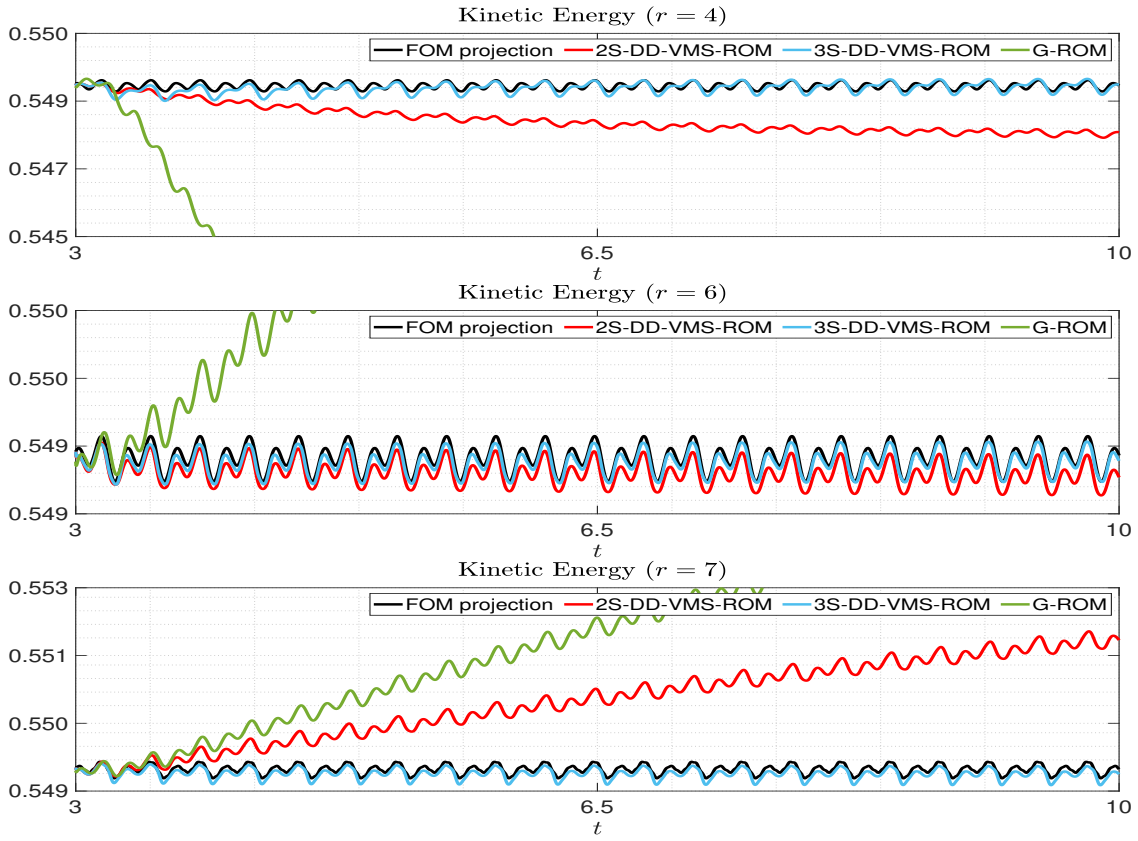

Figure 3.5: Flow past a cylinder, $Re = 100$, predictive regime. Time evolution of the kinetic energy for FOM projection, G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values.

The errors listed in Tables 3.11–3.13 and the plots in Figures 3.3–3.5 show that, in the reconstructive, cross-validation, and predictive regimes, the 3S-DD-VMS-ROM is consistently the most accurate ROM. Furthermore, the 3S-DD-VMS-ROM is more accurate than the 2S-DD-VMS-ROM, especially in the predictive regime.

Numerical Results for $Re = 500$

In this section, we present numerical results for the flow past a cylinder at $Re = 500$.

In Table 3.14, for different r values, we list the average L^2 error (3.39) for the G-ROM, the 2S-DD-VMS-ROM, and the new 3S-DD-VMS-ROM in the reconstructive regime. We also list the r_1 values for the 3S-DD-VMS-ROM. These results show that, for all r values, the 2S-DD-VMS-ROM and 3S-DD-VMS-ROM are several times (sometimes more than one order of magnitude) more accurate than the standard G-ROM. Furthermore, the 3S-DD-VMS-ROM is generally more accurate than the 2S-DD-VMS-ROM. For example, for $r = 2$, the 3S-DD-VMS-ROM is almost twice more accurate as the 2S-DD-VMS-ROM.

r	G-ROM	2S-DD-VMS-ROM	3S-DD-VMS-ROM	
	$\mathcal{E}(L^2)$	$\mathcal{E}(L^2)$	r_1	$\mathcal{E}(L^2)$
2	2.892e-01	7.029e-03	1	3.937e-03
3	3.344e-01	8.138e-02	2	7.517e-02
4	3.478e-01	4.195e-03	3	4.145e-03
5	3.795e-01	6.811e-02	2	5.915e-02
6	6.338e-02	3.864e-03	2	3.294e-03
7	5.738e-02	1.789e-02	2	1.563e-02
8	5.339e-02	5.734e-03	6	4.809e-03

Table 3.14: Flow past a cylinder, $Re = 500$, reconstructive regime. Average L^2 errors for G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values.

r	G-ROM	2S-DD-VMS-ROM	3S-DD-VMS-ROM	
	$\mathcal{E}(L^2)$	$\mathcal{E}(L^2)$	r_1	$\mathcal{E}(L^2)$
2	1.071e+00	2.015e-02	1	1.501e-02
3	8.280e-01	1.101e-01	2	8.428e-02
4	6.258e-01	1.218e-02	3	4.648e-03
5	6.440e-01	1.557e-01	3	7.329e-02
6	1.898e-01	5.733e-03	3	4.056e-03
7	1.531e-01	3.550e-02	2	2.033e-02
8	1.678e-01	9.050e-03	1	5.480e-03

Table 3.15: Flow past a cylinder, $Re = 500$, cross-validation regime. Average L^2 errors for G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values.

In Table 3.15, for different r values, we list the average L^2 error (3.39) for the G-ROM, the 2S-DD-VMS-ROM, and the new 3S-DD-VMS-ROM in the cross-validation regime. We also list the r_1 values for the 3S-DD-VMS-ROM. These results show that, for all r values,

the 2S-DD-VMS-ROM and 3S-DD-VMS-ROM are several times (sometimes even two orders of magnitude) more accurate than the standard G-ROM. Furthermore, the 3S-DD-VMS-ROM is generally more accurate than the 2S-DD-VMS-ROM. Specifically, for $r = 4, 5, 8$, the 3S-DD-VMS-ROM is almost twice more accurate than the 2S-DD-VMS-ROM.

In Table 3.16, for different r values, we list the average L^2 error (3.39) for the G-ROM, the 2S-DD-VMS-ROM, and the new 3S-DD-VMS-ROM in the predictive regime. We also list the r_1 values for the 3S-DD-VMS-ROM. These results show that, for all r values, the 2S-DD-VMS-ROM and 3S-DD-VMS-ROM are several times (sometimes even more than one order of magnitude) more accurate than the standard G-ROM. More importantly, for all r values (but especially for large r values), the 3S-DD-VMS-ROM is significantly more accurate than the 2S-DD-VMS-ROM: For example, for $r = 5, 6, 7$, and 8, the 3S-DD-VMS-ROM is *more than twice more accurate* than the 2S-DD-VMS-ROM.

r	G-ROM	2S-DD-VMS-ROM	3S-DD-VMS-ROM	
	$\mathcal{E}(L^2)$	$\mathcal{E}(L^2)$	r_1	$\mathcal{E}(L^2)$
2	7.351e-01	1.004e-01	1	1.004e-01
3	7.088e-01	8.838e-02	2	8.497e-02
4	5.871e-01	8.785e-03	1	8.785e-03
5	6.231e-01	9.735e-02	2	3.640e-02
6	1.293e-01	2.288e-02	4	9.051e-03
7	1.069e-01	2.816e-02	6	1.480e-02
8	1.130e-01	1.402e-02	6	5.544e-03

Table 3.16: Flow past a cylinder, $Re = 500$, predictive regime. Average L^2 errors for G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values.

In Figure 3.6, for $r = 4, 6, 7$, we plot the time evolution of the kinetic energy of the FOM, the G-ROM, the 2S-DD-VMS-ROM, and the new 3S-DD-VMS-ROM in the reconstructive regime. For all r values, the G-ROM kinetic energy approximation is very inaccurate. In contrast, the new 3S-DD-VMS-ROM accurately approximates the exact FOM kinetic energy for $r = 4, 6, 7$. The 2S-DD-VMS-ROM kinetic energy approximation is accurate for $r = 4$ and $r = 6$, but not for $r = 7$.

In Figure 3.7, for $r = 4, 6, 7$, we plot the time evolution of the kinetic energy of the FOM, the G-ROM, the 2S-DD-VMS-ROM, and the new 3S-DD-VMS-ROM in the cross-validation regime. For all r values, the G-ROM kinetic energy approximation is very inaccurate. In contrast, the new 3S-DD-VMS-ROM accurately approximates the exact FOM kinetic energy for $r = 4, 6, 7$. The 2S-DD-VMS-ROM kinetic energy approximation is accurate for $r = 4$ and $r = 6$, but not for $r = 7$.

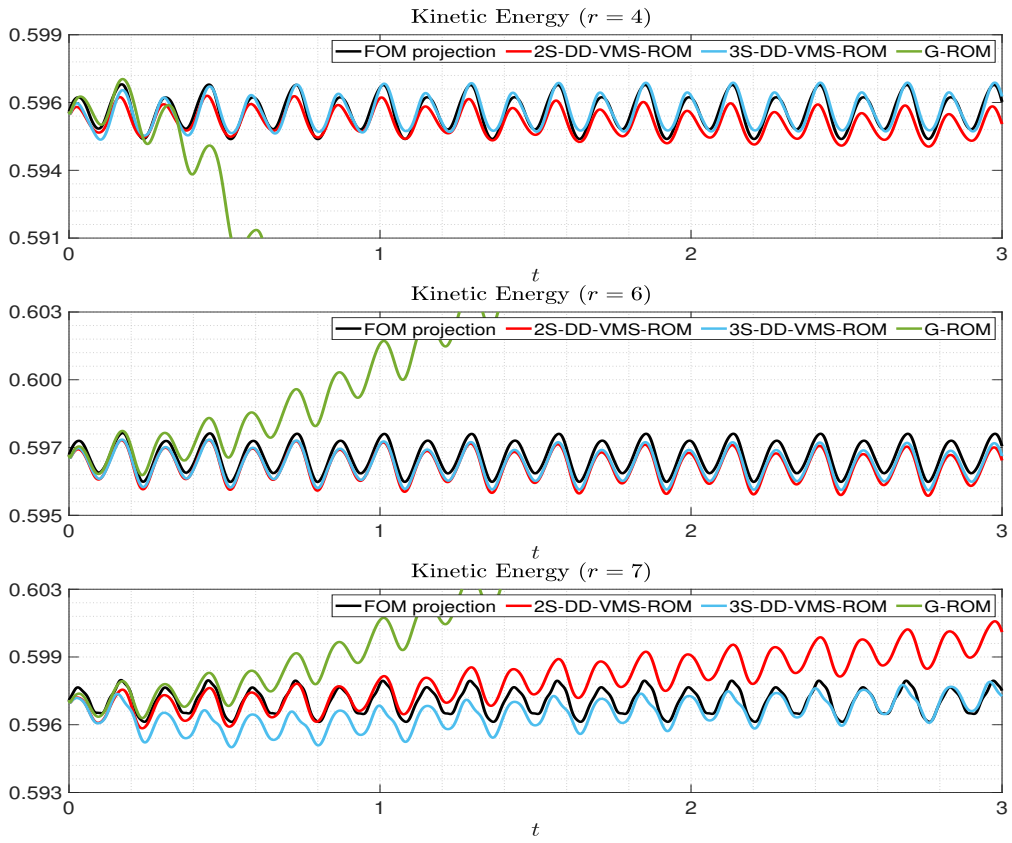

Figure 3.6: Flow past a cylinder, $Re = 500$, reconstructive regime. Time evolution of the kinetic energy for FOM projection, G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values.

In Figure 3.8, for $r = 4, 6, 7$, we plot the time evolution of the kinetic energy of the FOM, the G-ROM, the 2S-DD-VMS-ROM, and the new 3S-DD-VMS-ROM in the predictive regime. For all r values, the G-ROM kinetic energy approximation is very inaccurate. In contrast, the new 3S-DD-VMS-ROM accurately approximates the FOM kinetic energy for $r = 6$ and $r = 7$. For $r = 6$ and $r = 7$, the 2S-DD-VMS-ROM kinetic energy approximation is less accurate than the 3S-DD-VMS-ROM kinetic energy approximation but more accurate than the G-ROM kinetic energy approximation. For $r = 4$, both the 3S-DD-VMS-ROM and the 2S-DD-VMS-ROM kinetic energy approximations are accurate.

The errors listed in Tables 3.14–3.16 and the plots in Figures 3.6–3.8 show that, in the reconstructive, cross-validation, and predictive regimes, the 3S-DD-VMS-ROM is consistently the most accurate ROM. Furthermore, the 3S-DD-VMS-ROM is more accurate than the 2S-DD-VMS-ROM, especially in the predictive regime.

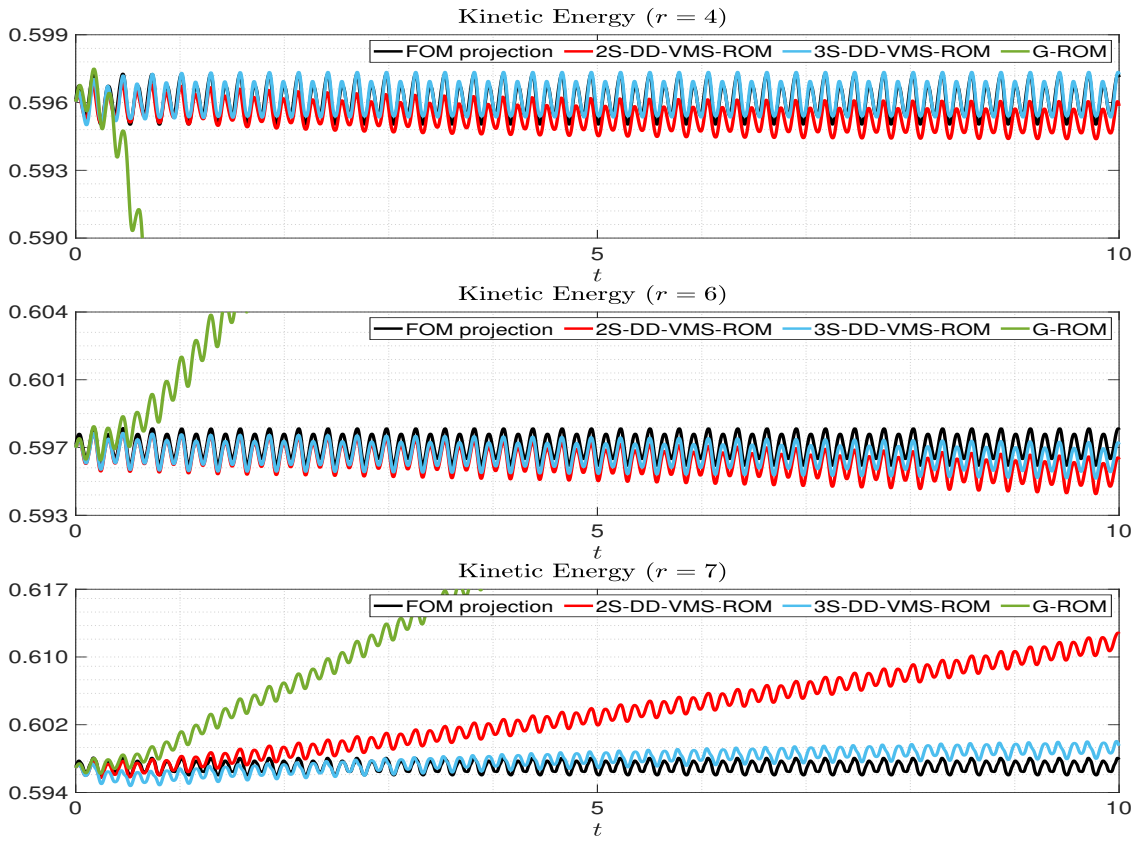

Figure 3.7: Flow past a cylinder, $Re = 500$, cross-validation regime. Time evolution of the kinetic energy for FOM projection, G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values.

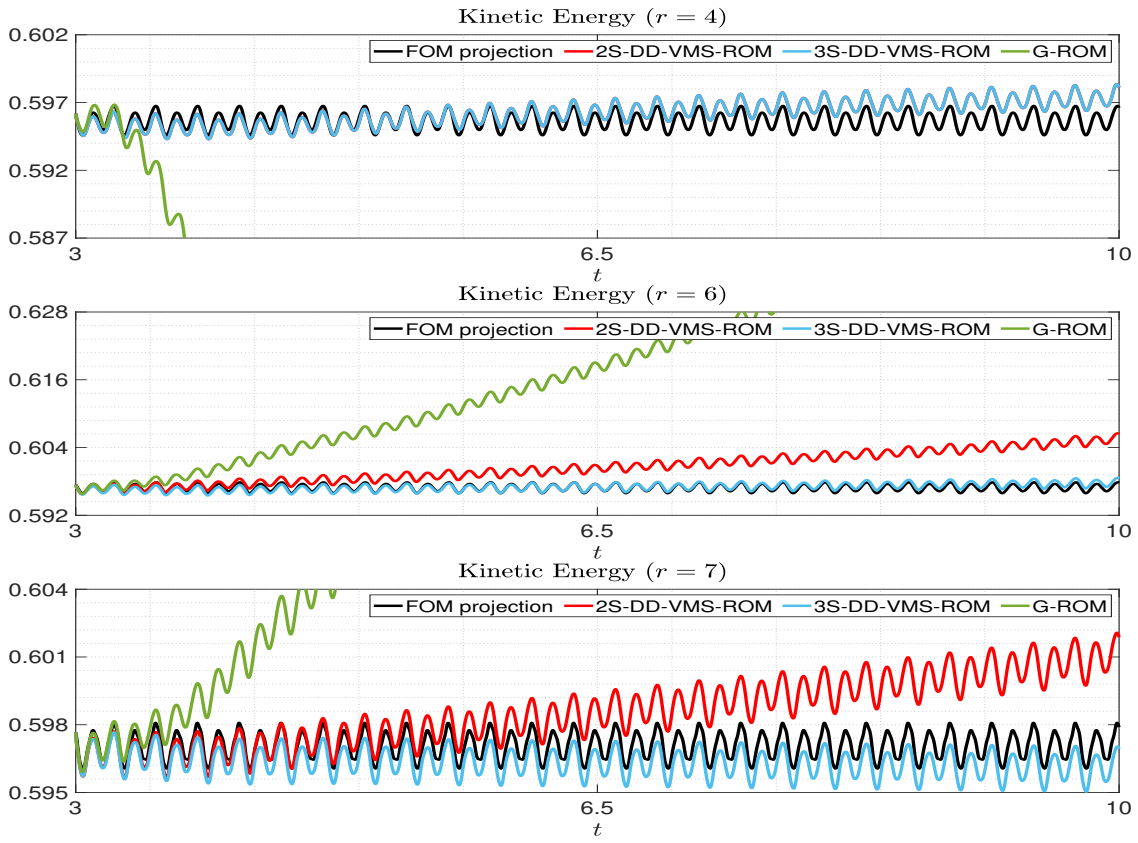

Figure 3.8: Flow past a cylinder, $Re = 500$, predictive regime. Time evolution of the kinetic energy for FOM projection, G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values.

Numerical Results for $Re = 1000$

In this section, we present numerical results for the flow past a cylinder at $Re = 1000$.

In Table 3.17, for different r values, we list the average L^2 error (3.39) for the G-ROM, the 2S-DD-VMS-ROM, and the new 3S-DD-VMS-ROM in the reconstructive regime. We also list the r_1 values for the 3S-DD-VMS-ROM. These results show that, for all r values, the 2S-DD-VMS-ROM and 3S-DD-VMS-ROM are several times (sometimes even more than one order of magnitude) more accurate than the standard G-ROM. Furthermore, the 3S-DD-VMS-ROM is generally more accurate than the 2S-DD-VMS-ROM. For example, for $r = 5$ and $r = 8$, the 3S-DD-VMS-ROM is *almost twice more accurate* than the 2S-DD-VMS-ROM.

r	G-ROM	2S-DD-VMS-ROM	3S-DD-VMS-ROM	
	$\mathcal{E}(L^2)$	$\mathcal{E}(L^2)$	r_1	$\mathcal{E}(L^2)$
2	4.937e-01	6.704e-03	1	6.692e-03
3	5.112e-01	6.804e-02	1	6.794e-02
4	5.980e-01	1.287e-02	2	9.869e-03
5	6.579e-01	1.794e-01	3	9.184e-02
6	1.503e-01	1.086e-02	4	8.210e-03
7	1.365e-01	2.848e-02	5	2.235e-02
8	7.076e-02	7.550e-03	4	4.836e-03

Table 3.17: Flow past a cylinder, $Re = 1000$, reconstructive regime. Average L^2 errors for G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values.

In Table 3.18, for different r values, we list the average L^2 error (3.39) for the G-ROM, the 2S-DD-VMS-ROM, and the new 3S-DD-VMS-ROM in the cross-validation regime. These results show that, for all r values, the 2S-DD-VMS-ROM and 3S-DD-VMS-ROM are several times (sometimes even two orders of magnitude) more accurate than the standard G-ROM. Furthermore, the 3S-DD-VMS-ROM is generally more accurate than the 2S-DD-VMS-ROM, especially for large r values. In particular, for $r = 5$, the 3S-DD-VMS-ROM is *almost five times more accurate* than the 2S-DD-VMS-ROM.

r	G-ROM	2S-DD-VMS-ROM	3S-DD-VMS-ROM	
	$\mathcal{E}(L^2)$	$\mathcal{E}(L^2)$	r_1	$\mathcal{E}(L^2)$
2	1.509e+00	1.504e-02	1	1.503e-02
3	8.595e-01	8.024e-02	1	8.024e-02
4	6.583e-01	2.538e-02	2	1.503e-02
5	7.095e-01	5.156e-01	3	1.026e-01
6	5.562e-01	3.132e-02	4	1.018e-02
7	4.760e-01	6.482e-02	6	3.505e-02
8	2.692e-01	1.691e-02	5	5.791e-03

Table 3.18: Flow past a cylinder, $Re = 1000$, cross-validation regime. Average L^2 errors for G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values.

In Table 3.19, for different r values, we list the average L^2 error (3.39) for the G-ROM, the 2S-DD-VMS-ROM, and the new 3S-DD-VMS-ROM in the predictive regime. We also list the r_1 values for the 3S-DD-VMS-ROM. These results show that, for all r values, the 2S-DD-VMS-ROM and 3S-DD-VMS-ROM are significantly (sometimes several times) more accurate than the standard G-ROM. More importantly, for all r values (but especially for large r values), the 3S-DD-VMS-ROM is significantly more accurate than the 2S-DD-VMS-

ROM: For example, for $r = 6$, the 3S-DD-VMS-ROM is *more than five times more accurate* than the 2S-DD-VMS-ROM.

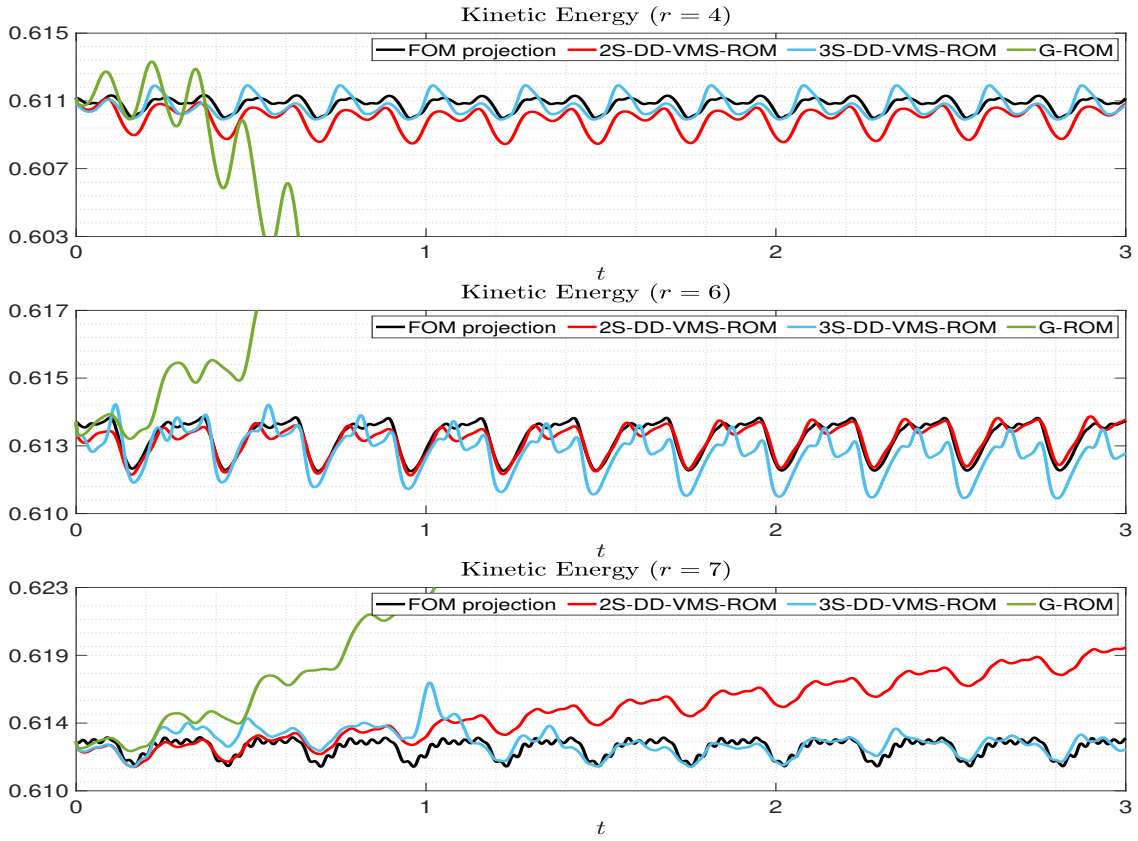

Figure 3.9: Flow past a cylinder, $Re = 1000$, reconstructive regime. Time evolution of the kinetic energy for FOM projection, G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values.

In Figure 3.9, for $r = 4, 6, 7$, we plot the time evolution of the kinetic energy of the FOM, the G-ROM, the 2S-DD-VMS-ROM, and the new 3S-DD-VMS-ROM in the reconstructive regime. For all cases, the evolution of the G-ROM kinetic energy is very inaccurate. In contrast, for $r = 4$ and $r = 6$, both the 3S-DD-VMS-ROM and the 2S-DD-VMS-ROM successfully reproduce the FOM kinetic energy. For $r = 7$, 3S-DD-VMS-ROM kinetic energy yields small oscillations for $0 \leq t \leq 1$, but it quickly converges to the FOM kinetic energy after $t > 1$. On the other hand, the 2S-DD-VMS-ROM kinetic energy approximation is not accurate.

r	G-ROM	2S-DD-VMS-ROM	3S-DD-VMS-ROM	
	$\mathcal{E}(L^2)$	$\mathcal{E}(L^2)$	r_1	$\mathcal{E}(L^2)$
2	1.146e+00	3.857e-01	1	2.860e-01
3	9.217e-01	4.522e-01	1	1.357e-01
4	7.207e-01	1.679e-01	2	7.070e-02
5	7.281e-01	5.620e-01	3	2.331e-01
6	3.545e-01	2.279e-01	2	3.733e-02
7	3.027e-01	2.273e-01	3	7.922e-02
8	1.587e-01	5.849e-02	3	4.394e-02

Table 3.19: Flow past a cylinder, $Re = 1000$, predictive regime. Average L^2 errors for G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values.

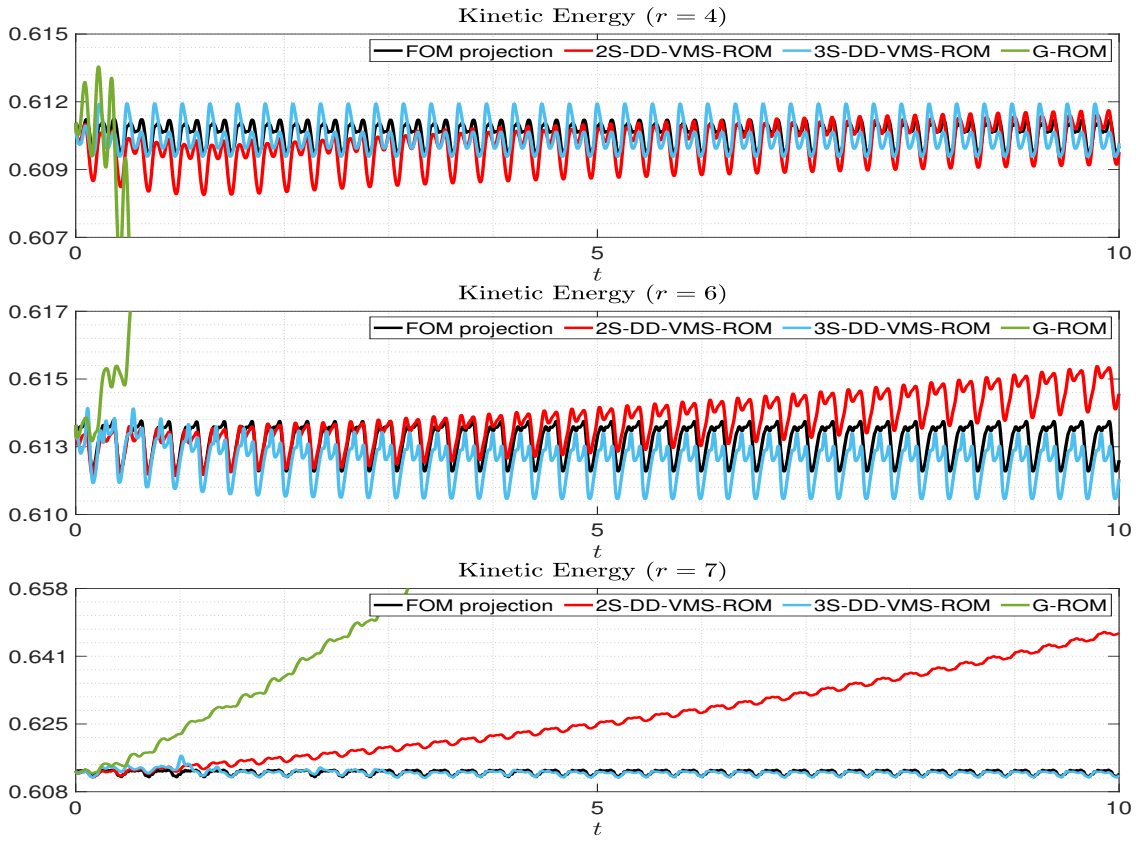

Figure 3.10: Flow past a cylinder, $Re = 1000$, cross-validation regime. Time evolution of the kinetic energy for FOM projection, G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values.

In Figure 3.10, for $r = 4, 6, 7$, we plot the time evolution of the kinetic energy of the FOM,

the G-ROM, the 2S-DD-VMS-ROM, and the new 3S-DD-VMS-ROM in the cross-validation regime. For all cases, the evolution of the G-ROM kinetic energy is very inaccurate. In contrast, for all cases, the 3S-DD-VMS-ROM successfully reproduces the exact FOM kinetic energy. The 2S-DD-VMS-ROM kinetic energy is accurate for $r = 4$, but not for $r = 6$ and, especially, for $r = 7$.

In Figure 3.11, for three r values, we plot the time evolution of the kinetic energy of the G-ROM, the 2S-DD-VMS-ROM, and the new 3S-DD-VMS-ROM in the predictive regime. For all cases, the evolution of the G-ROM kinetic energy is very inaccurate. For $r = 4$, the 3S-DD-VMS-ROM kinetic energy approximation is accurate, whereas the 2S-DD-VMS-ROM and the G-ROM kinetic energies are inaccurate. For $r = 6$ and $r = 7$, although the 3S-DD-VMS-ROM kinetic energy approximations are not as accurate, they are still much more accurate than the 2S-DD-VMS-ROM and, especially, the G-ROM kinetic energy approximations.

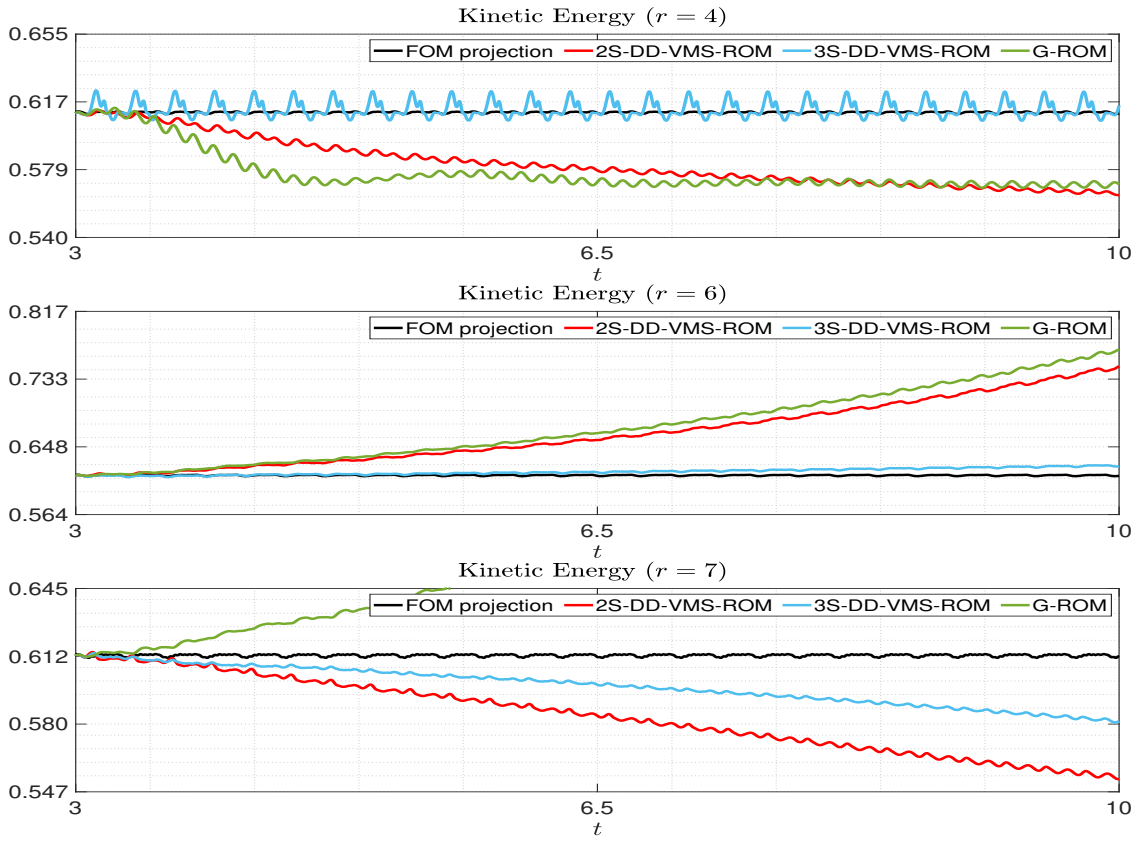

Figure 3.11: Flow past a cylinder, $Re = 1000$, predictive regime. Time evolution of the kinetic energy for FOM projection, G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values.

The errors listed in Tables 3.17–3.19 and the plots in Figures 3.9–3.11 show that, in the reconstructive, cross-validation, and predictive regimes, the 3S-DD-VMS-ROM is consistently the most accurate ROM. Furthermore, the 3S-DD-VMS-ROM is more accurate than the 2S-DD-VMS-ROM, especially in the predictive regime.

3.4.4 Quasi-Geostrophic Equations (QGE)

In this section, we investigate the 2S-DD-VMS-ROM (3.20) and the new 3S-DD-VMS-ROM (3.28) in the numerical simulation of the quasi-geostrophic equations (QGE)

$$\frac{\partial \omega}{\partial t} + J(\omega, \psi) - Ro^{-1}\frac{\partial \psi}{\partial x} = Re^{-1}\Delta\omega + Ro^{-1}F, \tag{3.44}$$

$$\omega = -\Delta\psi, \tag{3.45}$$

which are used to model the large scale ocean circulation [48, 82]. In (3.44)–(3.45), ω is the vorticity, ψ is the streamfunction, Re is the Reynolds number, and Ro is the Rossby number.

Computational Setting We follow [22, 52, 71, 75] and consider a symmetric double-gyre wind forcing given by

$$F = \sin(\pi(y-1)), \tag{3.46}$$

the computational domain $\Omega = [0,1] \times [0,2]$, the time domain $[0, 80]$, and the parameters $Re = 450$ and $Ro = 0.0036$. We also assume that ψ and ω satisfy homogeneous Dirichlet boundary conditions:

$$\psi(t,x,y) = 0, \qquad \omega(t,x,y) = 0 \qquad \text{for} \quad (x,y) \in \partial\Omega \ \text{ and } \ t \geq 0. \tag{3.47}$$

Snapshot Generation For the FOM discretization, we use a spectral method with a 257×513 spatial resolution and an explicit Runge-Kutta method. We follow [52, 71, 75] and run the FOM on the time interval $[0, 80]$. The flow displays a transient behavior on the time interval $[0, 10]$, and then converges to a statistically steady state on the time interval $[10, 80]$. We record the FOM solutions on the time interval $[10, 80]$ every 10^{-2} simulation time units, which ensures that the snapshots used in the construction of the ROM basis are equally spaced.

ROM Construction To construct the ROM basis, we follow the procedure described in Section 3.2 in [52] (see also [71, 75]). First, we collect 701 equally spaced FOM vorticity snapshots in the time interval $[10, 80]$ at equidistant time intervals. Next, for computational efficiency, we interpolate the FOM vorticity onto a uniform mesh with the resolution 257×513 over the spatial domain $\Omega = [0,1] \times [0,2]$, i.e., with a mesh size $\Delta x = \Delta y = 1/256$. Finally, we

use the interpolated snapshots and solve the corresponding eigenvalue problem to generate the ROM basis.

To train $\tilde{A}, \tilde{B}$ (for the 2S-DD-VMS-ROM) and $\tilde{A}_L, \tilde{B}_L$ and $\tilde{A}_S, \tilde{B}_S$ (for the 3S-DD-VMS-ROM), we use the same FOM data that was used to generate the ROM basis. Furthermore, to increase the computational efficiency of the 2S-DD-VMS-ROM and 3S-DD-VMS-ROM, we replace the R-dimensional FOM data with its d-dimensional approximation, where the parameter d satisfies $1 \leq d \leq R$ (for details, see Section 5.3 in [85], Section 4.3 in [50], and Section 3.2 in [52]). Specifically, we replace $\boldsymbol{\tau}^{FOM}$ with $\boldsymbol{\tau}^{d}$ (for the 2S-DD-VMS-ROM) and $\boldsymbol{\tau}_L^{FOM}$ and $\boldsymbol{\tau}_S^{FOM}$ with $\boldsymbol{\tau}_L^{d}$ and $\boldsymbol{\tau}_S^{d}$, respectively (for the 3S-DD-VMS-ROM). In our QGE numerical simulations, we choose $d = 3r$ to maintain a good balance between numerical accuracy and computational efficiency.

Numerical Results

Next, we present results for the 2S-DD-VMS-ROM (3.20) and the new 3S-DD-VMS-ROM (3.28) in the numerical simulation of the QGE (3.44)–(3.45). For clarity of presentation, we consider only the reconstructive regime.

To assess the ROM performance, we follow [52] and use the L^2 error of the time-averaged ROM streamfunction over the time interval $[10, 80]$:

$$\overline{\mathcal{E}}(L^2) = \left\| \overline{\psi^{FOM}(\boldsymbol{f}x, \cdot)} - \overline{\psi^{ROM}(\boldsymbol{f}x, \cdot)} \right\|_{L^2}^2, \tag{3.48}$$

where $\overline{(\cdot)}$ denotes the time average over the time interval $[10, 80]$, and $\boldsymbol{f}x = (x, y)$. In Table 3.20, for different r values, we list $\overline{\mathcal{E}}(L^2)$ for the G-ROM, the 2S-DD-VMS-ROM, and the new 3S-DD-VMS-ROM. We also list the r_1 values used for the 3S-DD-VMS-ROM. These results show that, for all r values, the 2S-DD-VMS-ROM and 3S-DD-VMS-ROM are orders of magnitude (sometimes two and even three orders of magnitude) more accurate than the standard G-ROM. Furthermore, the 3S-DD-VMS-ROM is generally more accurate than the 2S-DD-VMS-ROM: For example, for $r = 10$, $r = 15$, $r = 20$, and $r = 25$, the 3S-DD-VMS-ROM is about *three times more accurate* than the 2S-DD-VMS-ROM.

r	G-ROM	2S-DD-VMS-ROM	3S-DD-VMS-ROM	
	$\overline{\mathcal{E}}(L^2)$	$\overline{\mathcal{E}}(L^2)$	r_1	$\overline{\mathcal{E}}(L^2)$
10	3.734e+02	5.174e-01	5	1.996e-01
15	1.035e+02	3.853e-01	8	1.260e-01
20	1.371e+01	1.653e-01	9	5.175e-02
25	3.491e+00	3.434e-01	10	5.640e-02

Table 3.20: QGE, $Re = 450$, $Ro = 0.0036$, reconstructive regime. L^2 errors of the time-averaged streamfunction for G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values.

In Figure 3.12, for $r = 10$ and $r = 20$, we plot the time evolution of the kinetic energy of the FOM, the G-ROM, the 2S-DD-VMS-ROM, and the new 3S-DD-VMS-ROM in the reconstructive regime. These plots support the conclusions in Table 3.20: For $r = 10$, the G-ROM kinetic energy takes off very quickly and stabilizes at a level which is roughly 200 times higher than the FOM kinetic energy on average. In contrast, both the 2S-DD-VMS-ROM and the 3S-DD-VMS-ROM produce kinetic energies of the same order of magnitude as the FOM kinetic energy. Furthermore, the 3S-DD-VMS-ROM performs better than the 2S-DD-VMS-ROM in reproducing the peaks and the peak frequencies. As expected, for larger r values, the G-ROM's performance improves. For example, for $r = 20$, the G-ROM, the 2S-DD-VMS-ROM, and the 3S-DD-VMS-ROM kinetic energies perform similarly. We note, however, that for later times (e.g., on the time interval $[60, 80]$), the G-ROM kinetic energy is somewhat higher than the FOM kinetic energy, while the 2S-DD-VMS-ROM and 3S-DD-VMS-ROM kinetic energies are closer to the FOM kinetic energy.

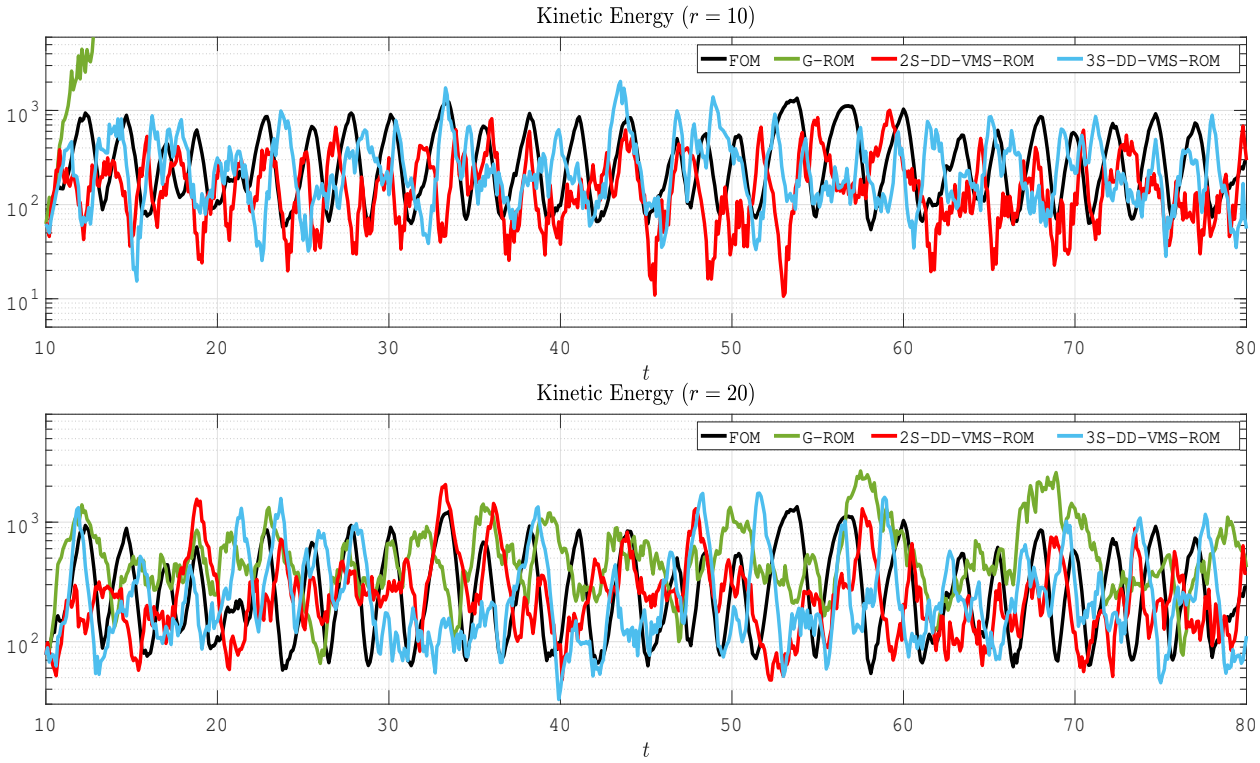

Figure 3.12: QGE, $Re = 450$, $Ro = 0.0036$, reconstructive regime. Time evolution of the kinetic energy for FOM, G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values.

We follow [52, 71] and, in Figure 3.13, for $r = 10$ and $r = 20$, we plot the time-average of the streamfunction ψ over the time interval $[10, 80]$ for the FOM, G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM. We emphasize that we use the same scale for the FOM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM plots. The plots in Figure 3.13 support the conclusions

in Table 3.20: For both $r = 10$ and $r = 20$, the 2S-DD-VMS-ROM and 3S-DD-VMS-ROM successfully reproduce the four gyre structure in the time-averaged streamfunction, whereas the G-ROM fails. Furthermore, the 3S-DD-VMS-ROM is more accurate than the 2S-DD-VMS-ROM.

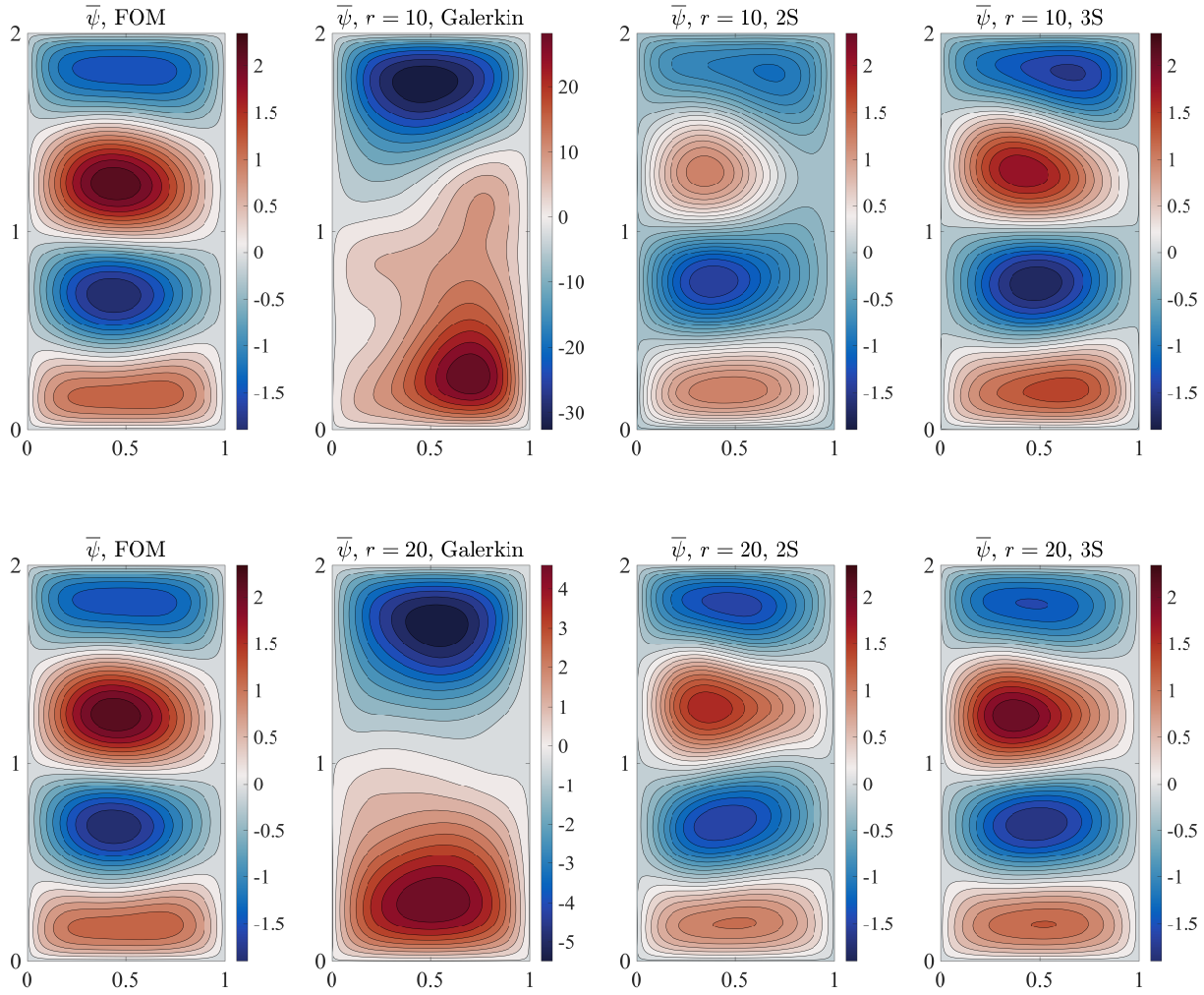

Figure 3.13: QGE, $Re = 450$, $Ro = 0.0036$, reconstructive regime. Time-averaged streamfunction ψ over the interval $[10, 80]$ for FOM, G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values.

The errors listed in Table 3.20 and the plots in Figures 3.12–3.13 show that, in the reconstructive regime, the 3S-DD-VMS-ROM is consistently the most accurate ROM.

3.4.5 Backward Facing Step

In this section, we investigate the 2S-DD-VMS-ROM (3.20) and the new 3S-DD-VMS-ROM (3.28) in the numerical simulation of a two-dimensional flow over a backward facing step at $Re = 1000$.

Computational Setting As a mathematical model, we use the NSE (3.8)–(3.9). We use the same computational domain as that used in Section 4.4 [5] and Section 8.2.2 in [67], i.e., a 44×9 rectangle with a unit height step placed at $(4, 0)$ (see the top plot in Figure 3.14).

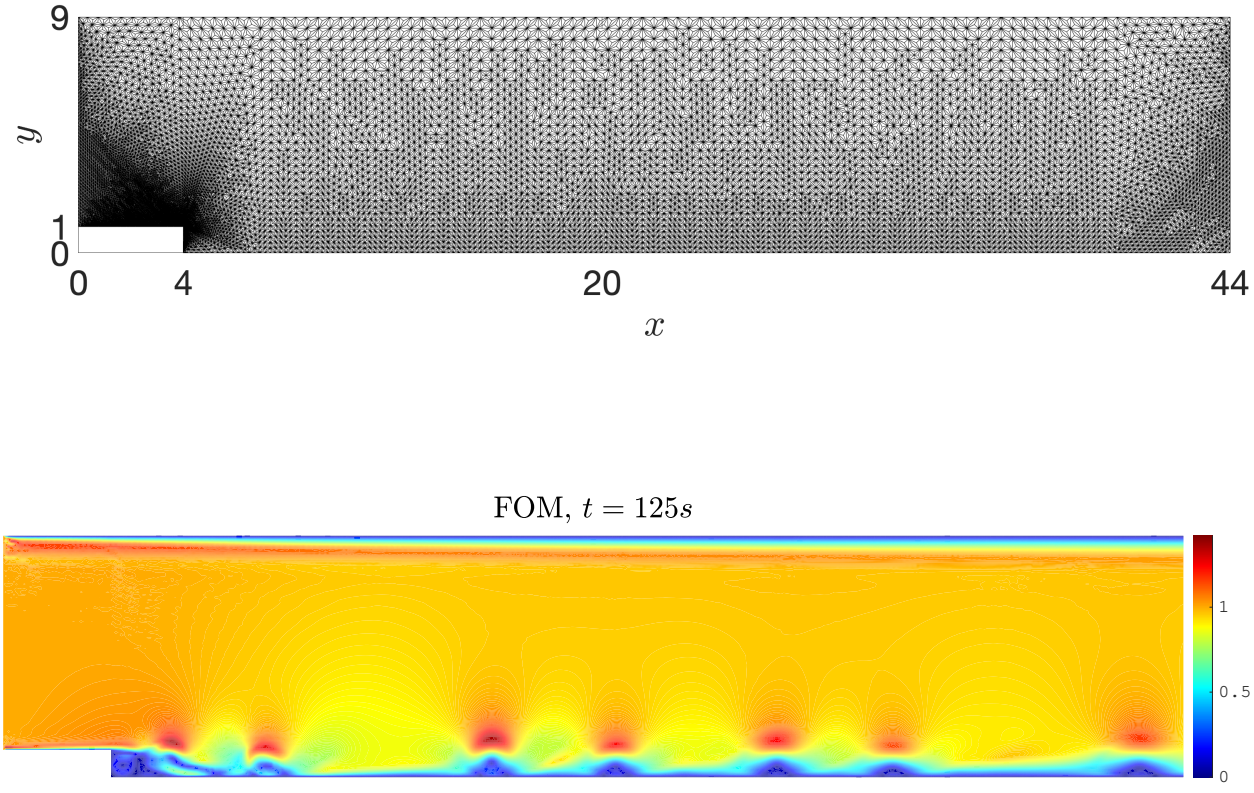

Figure 3.14: Backward facing step, $Re = 1000$. Geometry and finite element mesh (top). Magnitude of FOM velocity field at $t = 125$ (bottom).

Snapshot Generation For the spatial discretization, we use a barycenter refinement mesh of a Delaunay generated triangulation, which allows for (P_2, P_1^{disc}) Scott-Vogelius elements to be LBB stable (for details, see [41]). The mesh (see the top plot in Figure 3.14) has 209508 velocity and 156285 pressure degrees of freedom. We use the linearized BDF2 method and a time step size $\Delta t = 0.05$ for both FOM and ROM time discretizations. On the first time step, we use the backward Euler method so that we have two initial time step solutions required for the BDF2 scheme. For illustration purposes, in Figure 3.14 (the bottom plot), we display the magnitude of the FOM velocity field at $t = 125$.

In Figure 3.15, we plot the time evolution of the FOM kinetic energy on the time interval $[100, 150]$. This plot shows that the flow over a backward facing step that we consider is not periodic or periodic-like. The numerical results in the remainder of this section will show that this setting is more challenging for reduced order modeling than the other three test problems considered in Sections 3.4.2–3.4.4.

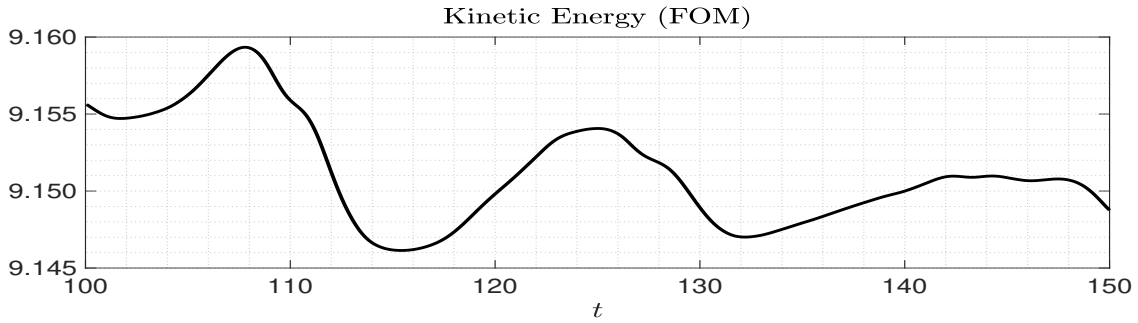

Figure 3.15: Backward facing step, $Re = 1000$. Time evolution of the FOM kinetic energy.

ROM Construction To build the ROM basis functions, we follow [67] and collect 1000) equally spaced FOM snapshots on the time interval $[100.05, 150]$.

To train $\tilde{A}, \tilde{B}$ (for the 2S-DD-VMS-ROM) and $\tilde{A}_L, \tilde{B}_L$ and $\tilde{A}_S, \tilde{B}_S$ (for the 3S-DD-VMS-ROM), we use the same FOM data that was used to generate the ROM basis. Furthermore, to increase the computational efficiency of the 2S-DD-VMS-ROM and 3S-DD-VMS-ROM, we use the approach described in Section 3.4.4 and replace $\boldsymbol{\tau}^{FOM}$ with $\boldsymbol{\tau}^{3r}$ (for the 2S-DD-VMS-ROM) and $\boldsymbol{\tau}_L^{FOM}$ and $\boldsymbol{\tau}_S^{FOM}$ with $\boldsymbol{\tau}_L^{3r}$ and $\boldsymbol{\tau}_S^{3r}$, respectively (for the 3S-DD-VMS-ROM). To further reduce the computational cost of the 3S-DD-VMS-ROM, we adopt a generic way in choosing r_1 for large r values (i.e., $r \geq 30$) and let $r_1 = \lfloor r/2 \rfloor$.

Numerical Results

Next, we present results for the 2S-DD-VMS-ROM (3.20) and the new 3S-DD-VMS-ROM (3.28) in the numerical simulation of the flow over a backward facing step at $Re = 1000$. For clarity

of presentation, we consider only the reconstructive regime.

In Table 3.21, for different r values, we list the average L^2 error (3.39) for the G-ROM, the 2S-DD-VMS-ROM, and the new 3S-DD-VMS-ROM. We also list the r_1 values for the 3S-DD-VMS-ROM. These results show that, for all r values, both the 2S-DD-VMS-ROM and the 3S-DD-VMS-ROM are about 30% more accurate than the standard G-ROM. Furthermore, the 3S-DD-VMS-ROM is consistently more accurate than the 2S-DD-VMS-ROM. This improvement is significant for low r values (i.e., $2 \leq r \leq 15$), and modest for large r values (i.e., $20 \leq r \leq 60$).

r	G-ROM	2S-DD-VMS-ROM	3S-DD-VMS-ROM	
	$\mathcal{E}(L^2)$	$\mathcal{E}(L^2)$	r_1	$\mathcal{E}(L^2)$
2	1.0270e+00	9.6593e-01	1	8.6129e-01
5	1.4864e+00	1.1671e+00	1	1.1070e+00
10	1.8401e+00	1.5064e+00	2	1.2932e+00
15	1.4733e+00	1.0909e+00	9	7.4297e-01
20	1.0392e+00	7.5813e-01	3	7.0704e-01
30	9.1723e-01	7.7908e-01	15	7.5835e-01
40	4.4118e-01	2.9694e-01	20	2.7753e-01
50	2.5578e-01	1.6002e-01	25	1.5586e-01
60	1.6772e-01	1.1679e-01	30	1.1276e-01

Table 3.21: Backward facing step, $Re = 1000$, reconstructive regime. Average L^2 errors for G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values.

We follow [5] (see also [67]) and, in Figure 3.16, for $r = 15$, we plot a pointwise quantity, i.e., the time evolution of the y-component of the velocity, v, for the FOM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM at the point with coordinates $(19, 1)$, which is physically located behind the step. This plot shows that both the 2S-DD-VMS-ROM and the 3S-DD-VMS-ROM are significantly more accurate than the G-ROM. Furthermore, the 3S-DD-VMS-ROM is more accurate than the 2S-DD-VMS-ROM.

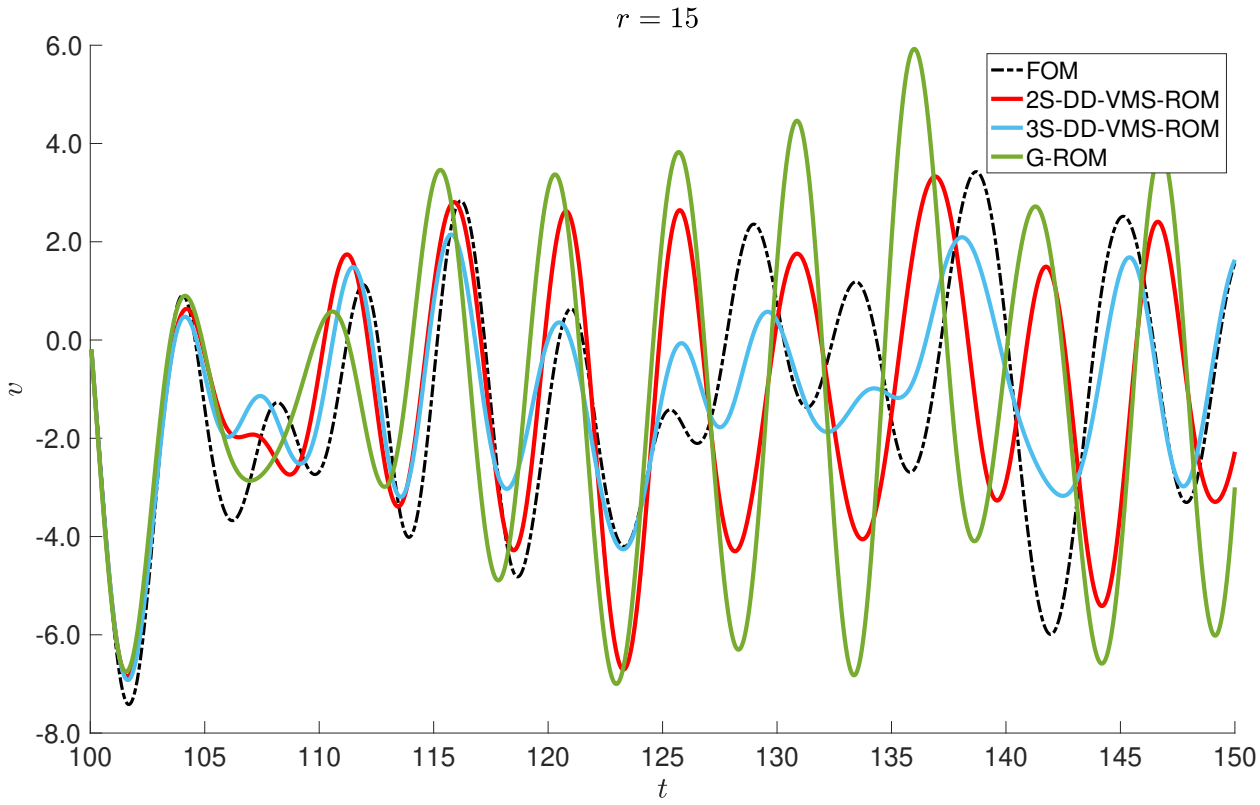

Figure 3.16: Backward facing step, $Re = 1000$, reconstructive regime. Time evolution of the y-component of the velocity, v, of FOM, G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM with $r = 15$ at the point with coordinates $(19, 1)$.

In Figure 3.17, for $r = 30, 40$, and 60, we plot the time evolution of the kinetic energy of the FOM, G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM. These plots support the conclusions in Table 3.21. Specifically, for low r values (i.e., $r = 30$), the G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM results are relatively inaccurate. However, for medium r values (i.e., $r = 40$), the 2S-DD-VMS-ROM and 3S-DD-VMS-ROM results are significantly more accurate than the G-ROM results. As expected, for high r values (i.e., $r = 60$), the 2S-DD-VMS-ROM, 3S-DD-VMS-ROM, and G-ROM results are all accurate. Furthermore, for $r = 40$ the 3S-DD-VMS is more accurate than the 2S-DD-VMS-ROM. For $r = 30$ and $r = 60$, the 2S-DD-VMS-ROM and the 3S-DD-VMS-ROM perform similarly.

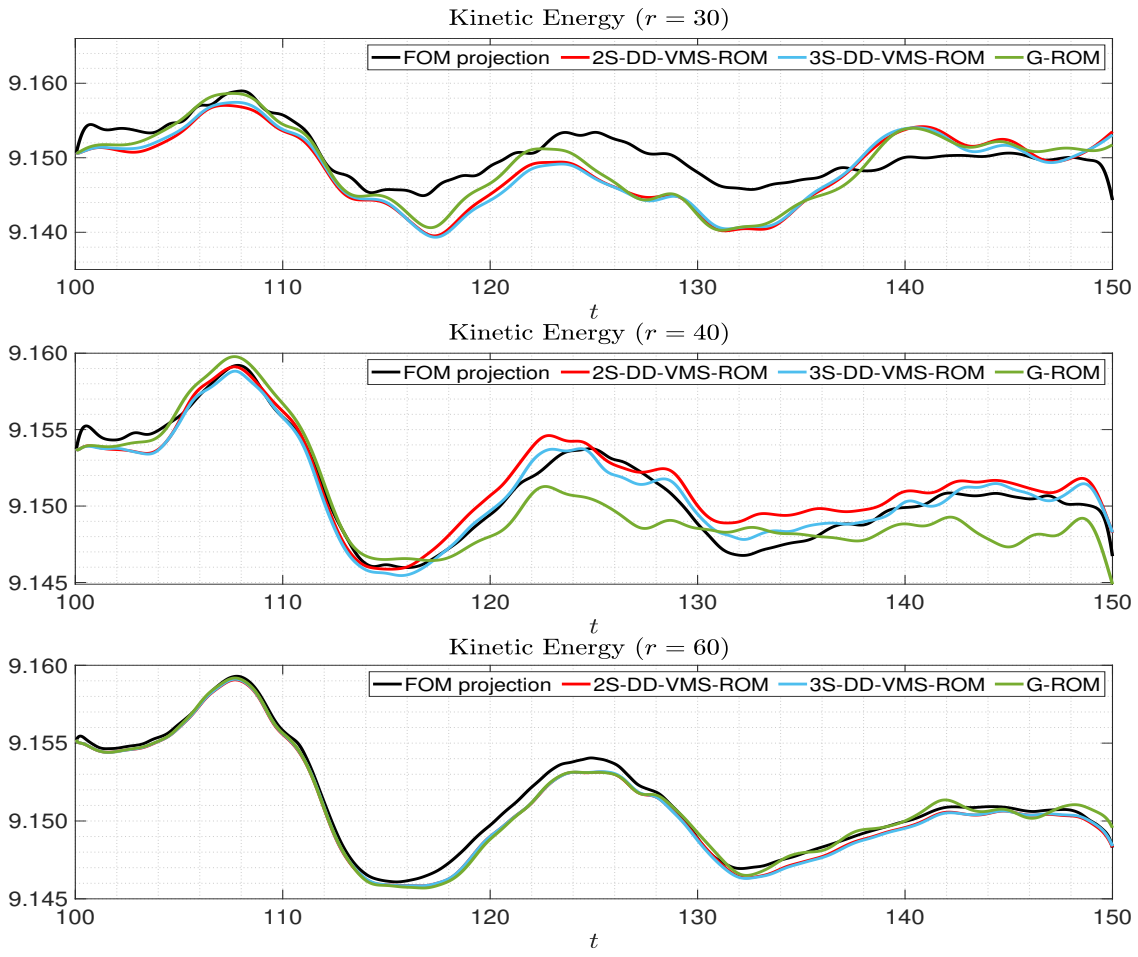

Figure 3.17: Backward facing step, $Re = 1000$, reconstructive regime. Time evolution of the kinetic energy for FOM projection, G-ROM, 2S-DD-VMS-ROM and 3S-DD-VMS-ROM for different r values.

We follow [5, 67] and, in Figure 3.18, for $r = 15$, we plot a pointwise quantity, i.e., the spectrum of the $y-$component of the velocity, v, for the FOM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM at the point with coordinates $(19, 1)$. This plot shows that the 2S-DD-VMS-ROM spectrum is more accurate than the G-ROM spectrum. Furthermore, the 3S-DD-VMS-ROM spectrum is more accurate than the 2S-DD-VMS-ROM spectrum.

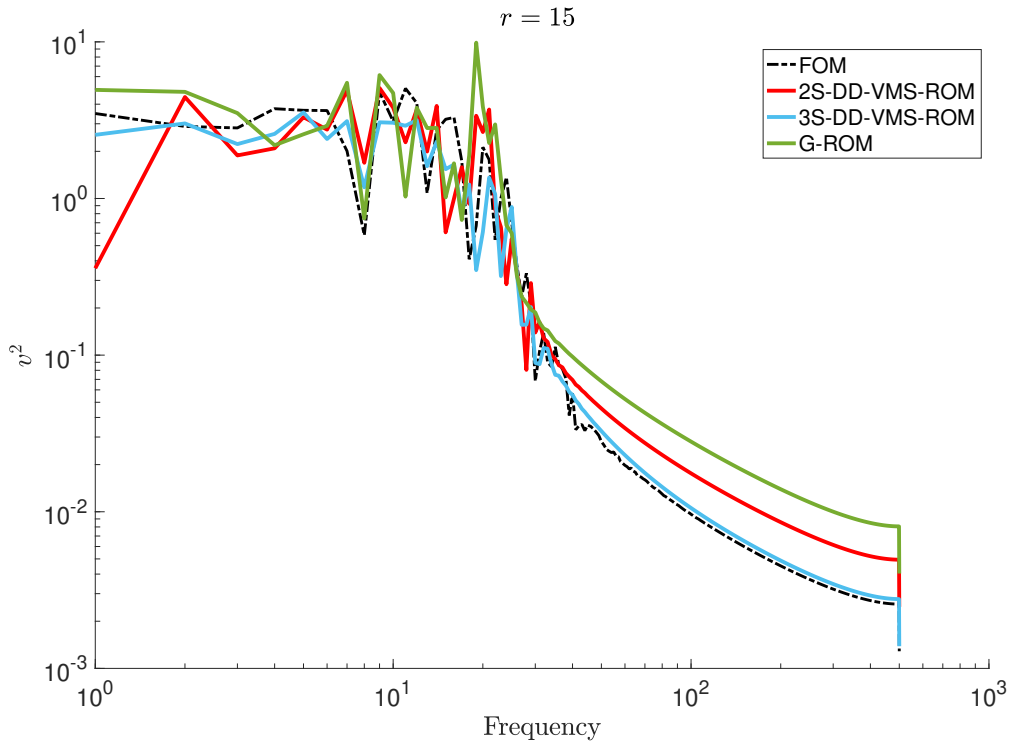

Figure 3.18: Backward facing step, $Re = 1000$, reconstructive regime. The spectrum of the y-component of the velocity for FOM, G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM with $r = 15$ at the point with coordinates $(19, 1)$.

The errors listed in Table 3.21 and the plots in Figures 3.16–3.18 show that, in the reconstructive regime, both the 2S-DD-VMS-ROM and the 3S-DD-VMS-ROM are more accurate than the G-ROM. Furthermore, the 3S-DD-VMS-ROM is more accurate than the 2S-DD-VMS-ROM. However, for the backward facing step test problem, this improvement is not as significant as for the other three test cases investigated in Sections 3.4.2–3.4.4.

3.4.6 Qualitative Comparison of 2S-DD-VMS-ROM and 3S-DD-VMS-ROM

In the previous sections, we performed a quantitative comparison of the 2S-DD-VMS-ROM and the 3S-DD-VMS-ROM in the numerical simulation of the Burgers equation (Section 3.4.2), the flow past a cylinder (Section 3.4.3), the QGE (Section 3.4.4), and the flow over a backward facing step (Section 3.4.5). In all our numerical simulations, the 3S-DD-VMS-ROM was more accurate than the 2S-DD-VMS-ROM, although this improvement was less significant for the flow over a backward facing step. In this section, we present a qualitative comparison of the 2S-DD-VMS-ROM and 3S-DD-VMS-ROM.

We believe that the 3S-DD-VMS-ROM is more accurate than the 2S-DD-VMS-ROM in our numerical tests because the 3S-DD-VMS-ROM is more *flexible* than the 2S-DD-VMS-ROM. Specifically, the 2S-DD-VMS-ROM has only one control parameter in the truncated SVD used in Algorithm 2, i.e., the tolerance tol. The 3S-DD-VMS-ROM, on the other hand, has two control parameters in the truncated SVD used in Algorithm 3: (i) the tolerance tol_L used in the truncated SVD for the least squares problem for the large resolved scales, and (ii) the tolerance tol_S used in the truncated SVD for the least squares problem for the small resolved scales. Thus, in principle, by choosing optimal values for the two modeling parameters in the 3S-DD-VMS-ROM (i.e., tol_L and tol_S), we can obtain more accurate results than those obtained with the 2S-DD-VMS-ROM, which has only one modeling parameter (i.e., tol). The truncated SVD components of the 2S-DD-VMS-ROM and 3S-DD-VMS-ROM algorithms aim at alleviating the ill-conditioning that is common in data-driven least squares problems (see, e.g., [50, 59, 85]). Our numerical investigation shows that the tolerances used in the truncated SVD have a significant effect on the 2S-DD-VMS-ROM and 3S-DD-VMS-ROM results. Furthermore, our numerical results confirm that having more flexibility in choosing the two tolerances in the 3S-DD-VMS-ROM yields more accurate results.

For example, for the Burgers equation, the results in Table 3.4 show that, for $r = 3$, choosing two different tolerances in the 3S-DD-VMS-ROM (i.e., $tol_L = 10^0$ and $tol_S = 10^{-2}$) yields more accurate results than the 2S-DD-VMS-ROM, which uses only one tolerance (i.e., $tol = 10^0$). Indeed, the 3S-DD-VMS-ROM average L^2 error is more than an order of magnitude lower than the 2S-DD-VMS-ROM average L^2 error.

The flow past a circular cylinder test case yields similar conclusions. We follow [5] and, in Figure 3.19, for $r = 5$, we plot the time evolution of the y-component of the velocity, v, of the FOM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM at the point with coordinates $(0.43, 0.2)$, which is physically located behind the circular cylinder. The plot in Figure 3.19 clearly shows that choosing two different tolerances in the 3S-DD-VMS-ROM algorithm yields more accurate results than the 2S-DD-VMS-ROM, which uses only one tolerance.

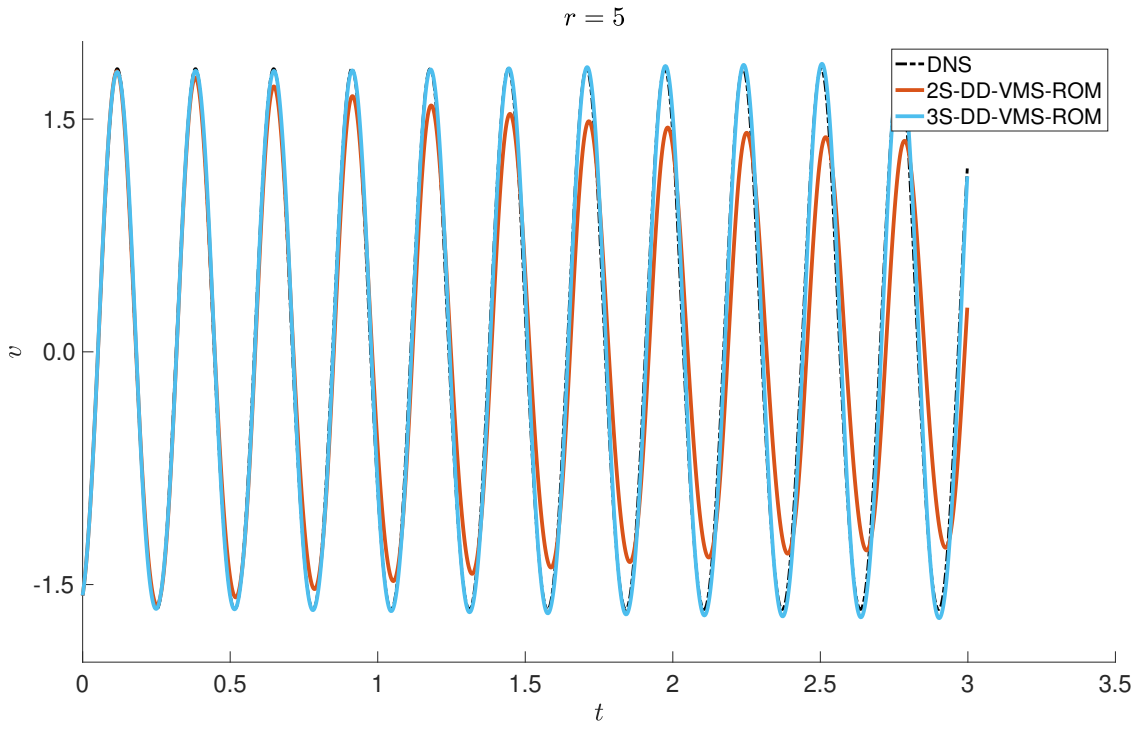

Figure 3.19: Flow past a cylinder, $Re = 1000$, reconstructive regime. Time evolution of the y-component of the velocity, v, of the FOM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM with $r = 5$ at the point with coordinates $(0.43, 0.2)$.

Furthermore, we follow [5] and, for the 2S-DD-VMS-ROM and 3S-DD-VMS-ROM, in Figure 3.20 we plot the first component of the vectors $\boldsymbol{\tau}^{FOM}$ and $\boldsymbol{\tau}^{ROM}$ with the FOM and ROM representations of the VMS-ROM closure terms, which are defined in (3.17) for the 2S-DD-VMS-ROM and in (3.24)–(3.25) for the 3S-DD-VMS-ROM. Specifically, at each time step t_j, $j = 1, \ldots, M$,

$$\begin{aligned} \boldsymbol{\tau}^{FOM}(t_j) = \; - \; & \left[\left(\left(\boldsymbol{u}_R^{FOM}(t_j) \cdot \nabla \right) \boldsymbol{u}_R^{FOM}(t_j) \, , \boldsymbol{\varphi}_i \right) \right. \\ & \left. - \left(\left(\boldsymbol{u}_r^{FOM}(t_j) \cdot \nabla \right) \boldsymbol{u}_r^{FOM}(t_j) \, , \boldsymbol{\varphi}_i \right) \right] , \end{aligned} \tag{3.49}$$

where $\boldsymbol{u}_R^{FOM}(t_j)$ and $\boldsymbol{u}_r^{FOM}(t_j)$ are defined in (3.19), and

$$\boldsymbol{\tau}^{ROM}(t_j) = \tilde{A} \, \boldsymbol{a}^{ROM}(t_j) + \boldsymbol{a}^{ROM}(t_j)^\top \, \tilde{B} \, \boldsymbol{a}^{ROM}(t_j) \, , \tag{3.50}$$

where $\tilde{A}$ and $\tilde{B}$ are the DD-VMS-ROM operators, and $\boldsymbol{a}^{ROM}(t_j)$ is the ROM solution at time step t_j. The plot in Figure 3.20 shows that the first component of the 2S-DD-VMS-ROM and 3S-DD-VMS-ROM closure terms are different. Thus, we conclude that the tolerance used in the truncated SVD has a significant effect on the ROM closure model and on the corresponding ROM results (as illustrated in Figure 3.19).

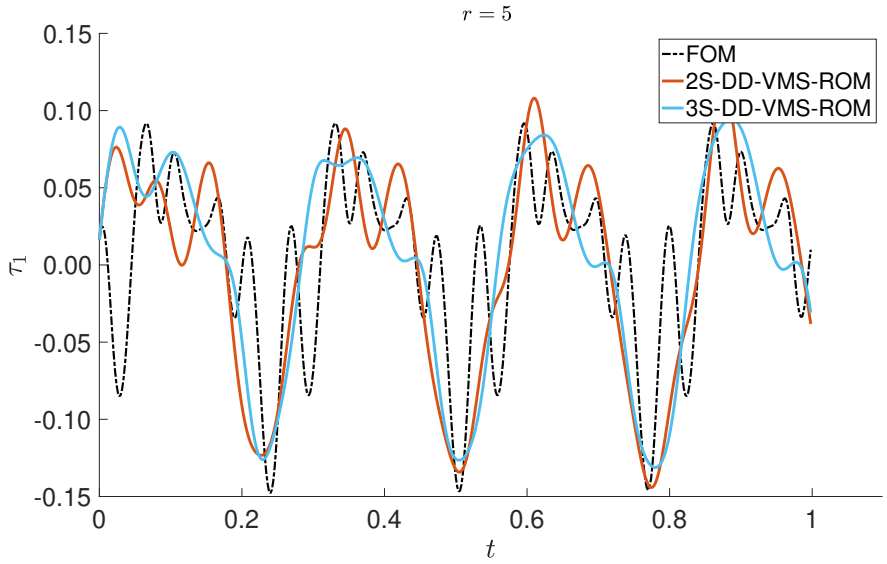

Figure 3.20: Flow past a cylinder, $Re = 1000$, reconstructive regime. Time evolution of the first component of the subscales for the FOM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM with $r = 5$.

For the QGE test case, the results in Table 3.20 show that choosing two different tolerances in the 3S-DD-VMS-ROM yields more accurate results than the 2S-DD-VMS-ROM, which uses only one tolerance. For example, for $r = 25$, the 3S-DD-VMS-ROM L^2 error is more than six times lower than the 2S-DD-VMS-ROM L^2 error.

For the backward facing step test case, the results in Table 3.21 (see also Figures 3.16–3.18) support the same conclusion. Indeed, the 3S-DD-VMS-ROM (which uses two different tolerances) is more accurate than the 2S-DD-VMS-ROM (which uses only one tolerance). This improvement is significant for low r values (i.e., $2 \leq r \leq 15$), and modest for large r values (i.e., $20 \leq r \leq 60$).

We emphasize that both the quantitative comparisons (in Sections 3.4.2–3.4.5) and the qualitative comparison in this section are only valid for the 2S-DD-VMS-ROM and the 3S-DD-VMS-ROM. Thus, our conclusions do not carry over to other types of VMS-ROMs, e.g., [5, 8, 18, 24, 35, 37, 67, 68, 69, 78, 81, 84]. In particular, we do not perform a general comparison of two-scale VMS-ROMs and three-scale VMS-ROMs. Instead, we take a more modest step and compare two specific examples from the two classes, i.e., the 2S-DD-VMS-ROM and the 3S-DD-VMS-ROM, respectively. We believe that extending to the ROM setting two-scale and three-scale VMS models developed for classical numerical discretizations (see, e.g., the surveys in [3, 14, 39, 64]), and comparing the resulting two-scale and three-scale VMS-ROMs is a worthy research endeavor that could yield conclusions that are different from the conclusions drawn from our numerical investigation (see, e.g., [2] for the

finite element setting). This, however, is beyond the scope of this paper.

3.5 Conclusions and Outlook

In this paper, we propose a new data-driven variational multiscale reduced order model (DD-VMS-ROM) framework. We construct the new DD-VMS-ROM framework in two steps: In the first step, we leverage the VMS methodology and the hierarchical structure of the ROM basis to provide explicit mathematical formulas for the interaction among the ROM spatial scales. In the second step, we use the available full order model (FOM) data to construct structural VMS-ROM closure models for the interactions among scales. We investigate two DD-VMS-ROMs: (i) The two-scale DD-VMS-ROM (2S-DD-VMS-ROM) considers two scales: resolved scales and unresolved scales. For the 2S-DD-VMS-ROM, we construct one ROM closure model for the interaction between the resolved and unresolved scales. (ii) The three-scale DD-VMS-ROM (3S-DD-VMS-ROM) considers three scales: resolved large scales, resolved small scales, and unresolved scales. For the 3S-DD-VMS-ROM, we construct one ROM closure model for the interaction between the resolved large and resolved small scales, and another ROM closure model for the interaction between resolved small scales and unresolved scales. We test the 2S-DD-VMS-ROM and 3S-DD-VMS-ROM in the numerical simulation of four test cases: (i) the 1D Burgers equation with viscosity coefficient $\nu = 10^{-3}$; (ii) a 2D flow past a circular cylinder at Reynolds numbers $Re = 100$, $Re = 500$, and $Re = 1000$; (iii) the quasi-geostrophic equations at Reynolds number $Re = 450$ and Rossby number $Ro = 0.0036$; and (iv) a 2D flow over a backward facing step at Reynolds number $Re = 1000$. We consider the reconstructive regime for all the test cases, and the cross-validation and predictive regimes for the Burgers equation and the 2D flow past a circular cylinder test cases. The numerical results show that both the 2S-DD-VMS-ROM and the 3S-DD-VMS-ROM are more accurate than the standard Galerkin ROM (G-ROM). Furthermore, the 3S-DD-VMS-ROM is more accurate than the 2S-DD-VMS-ROM, although this improvement is less significant for the flow over a backward facing step.

We intend to pursue several research avenues in the development of the new DD-VMS-ROM framework. The first research direction that we plan to investigate is finding the optimal parameter r_1 and the optimal tolerances tol_L and tol_S in the new 3S-DD-VMS-ROM. In this paper, we used a trial and error approach to find these parameters. We intend to investigate whether providing rigorous error estimates [28, 36, 61] or leveraging physical insight [30] can provide parameters that yield more accurate results. Another research direction that we plan to pursue is the development of new DD-VMS-ROM closure models by leveraging ideas from VMS methods for finite element discretizations (see, e.g., Section 8.8 in [39]), e.g., the time-dependent subscale-orthogonal methods [13, 67, 68]. We also plan to explore different topological structures for the ROM closure term. In the present study, we assume that the structure of the ROM closure model function $\boldsymbol{g}$ is similar to the structure of the Galerkin model function $\boldsymbol{f}$ and we utilize a least squares approach to determine the shape

of $\boldsymbol{g}$. We emphasize that, without loss of generality, our DD-VMS-ROM framework can be formulated by utilizing a supervised machine learning approach [63, 72, 73, 74], a topic that we would like to explore in the future. Finally, we intend to explore the extension of the new DD-VMS-ROM to the Petrov-Galerkin framework [10, 11, 23, 58].

Bibliography

[1] M. Ahmed and O. San. Stabilized principal interval decomposition method for model reduction of nonlinear convective systems with moving shocks. *Comp. Appl. Math.*, 37(5):6870–6902, 2018.

[2] N. Ahmed and V. John. An assessment of two classes of variational multiscale methods for the simulation of incompressible turbulent flows. *Comput. Methods Appl. Mech. Engrg.*, 365:112997, 2020.

[3] N. Ahmed, T. C. Rebollo, V. John, and S. Rubino. A review of variational multiscale methods for the simulation of turbulent incompressible flows. *Arch. Comput. Method. E.*, 24(1):115–164, 2017.

[4] J. Baiges, R. Codina, I. Castanar, and E. Castillo. A finite element reduced order model based on adaptive mesh refinement and artificial neural networks. 2019.

[5] J. Baiges, R. Codina, and S. Idelsohn. Reduced-order subscales for POD models. *Comput. Methods Appl. Mech. Engrg.*, 291:173–196, 2015.

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

[7] M. Benosman, J. Borggaard, O. San, and B. Kramer. Learning-based robust stabilization for reduced-order models of 2D and 3D Boussinesq equations. *Appl. Math. Model.*, 49:162–181, 2017.

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

[9] S. L. Brunton and J. N. Kutz. *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press, 2019.

[10] K. Carlberg, M. Barone, and H. Antil. Galerkin v. least-squares Petrov–Galerkin projection in nonlinear model reduction. *J. Comput. Phys.*, 330:693–734, 2017.

[11] K. Carlberg, C. Bou-Mosleh, and C. Farhat. Efficient non-linear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximations. *Int. J. Num. Meth. Eng.*, 86(2):155–181, 2011.

[12] M. D. Chekroun, H. Liu, and J. C. McWilliams. Variational approach to closure of nonlinear dynamical systems: Autonomous case. *J. Stat. Phys.*, pages 1–88, 2019.

[13] R. Codina. Stabilized finite element approximation of transient incompressible flows using orthogonal subscales. *Comput. Methods Appl. Mech. Engrg.*, 191(39-40):4295–4321, 2002.

[14] R. Codina, S. Badia, J. Baiges, and J. Principe. Variational multiscale methods in computational fluid dynamics. *Encyclopedia of Computational Mechanics Second Edition*, pages 1–28, 2018.

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

[16] D. T. Crommelin and A. J. Majda. Strategies for model reduction: comparing different optimal bases. *J. Atmos. Sci.*, 61:2206–2217, 2004.

[17] V. DeCaria, T. Iliescu, W. Layton, M. McLaughlin, and M. Schneier. An artificial compression reduced order model. *SIAM J. Numer. Anal.*, 2020. accepted.

[18] F. G. Eroglu, S. Kaya, and L. G. Rebholz. A modular regularized variational multiscale proper orthogonal decomposition for incompressible flows. *Comput. Meth. Appl. Mech. Eng.*, 325:350–368, 2017.

[19] L. Fick, Y. Maday, A. T. Patera, and T. Taddei. A stabilized POD model for turbulent flows over a range of Reynolds numbers: Optimal parameter sampling and constrained projection. *J. Comp. Phys.*, 371:214–243, 2018.

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

[21] A. Gouasmi, E. J. Parish, and K. Duraisamy. A priori estimation of memory effects in reduced-order models of nonlinear systems using the Mori–Zwanzig formalism. *Proc. R. Soc. A*, 473(2205):20170385, 2017.

[22] R. J. Greatbatch and B. T. Nadiga. Four-gyre circulation in a barotropic model with double-gyre wind forcing. *J. Phys. Oceanogr.*, 30(6):1461–1471, 2000.

[23] S. Grimberg, C. Farhat, and N. Youkilis. On the stability of projection-based model order reduction for convection-dominated laminar and turbulent flows. *arXiv preprint,http://arxiv.org/abs/2001.10110*, 2020.

[24] F. Güler Eroğlu, S. Kaya, and L. G. Rebholz. Decoupled modular regularized VMS-POD for Darcy-Brinkman equations. *IAENG Int. J. Appl. Math.*, 2019.

[25] M. Gunzburger, N. Jiang, and M. Schneier. An ensemble-proper orthogonal decomposition method for the nonstationary Navier-Stokes equations. *SIAM J. Numer. Anal.*, 55(1):286–304, 2017.

[26] J. Harlim, S. W. Jiang, S. Liang, and H. Yang. Machine learning for prediction with missing dynamics. *arXiv preprint,* http://arxiv.org/abs/1910.05861, 2019.

[27] D. Hartmann, M. Herz, and U. Wever. Model order reduction a key technology for digital twins. In *Reduced-order modeling (ROM) for simulation and optimization*, pages 167–179. Springer, Cham, Switzerland, 2018.

[28] J. S. Hesthaven, G. Rozza, and B. Stamm. *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*. Springer, 2015.

[29] S. Hijazi, G. Stabile, A. Mola, and G. Rozza. Data-driven POD-Galerkin reduced order model for turbulent flows. *arXiv preprint,* http://arxiv.org/abs/1907.09909, 2019.

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

[31] T. J. R. Hughes, G. R. Feijóo, L. Mazzei, and J.-B. Quincy. The variational multiscale method – a paradigm for computational mechanics. *Comput. Methods Appl. Mech. Engrg.*, 166(1):3–24, 1998.

[32] T. J. R. Hughes, L. Mazzei, and K. E. Jansen. Large eddy simulation and the variational multiscale method. *Comput. Vis. Sci.*, 3:47–59, 2000.

[33] T. J. R. Hughes, L. Mazzei, A. Oberai, and A. Wray. The multiscale formulation of large eddy simulation: Decay of homogeneous isotropic turbulence. *Phys. Fluids*, 13(2):505–512, 2001.

[34] T. J. R. Hughes, A. Oberai, and L. Mazzei. Large eddy simulation of turbulent channel flows by the variational multiscale method. *Phys. Fluids*, 13(6):1784–1799, 2001.

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

[36] T. Iliescu and Z. Wang. Are the snapshot difference quotients needed in the proper orthogonal decomposition? *SIAM J. Sci. Comput.*, 36(3):A1221–A1250, 2014.

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

[38] V. John. Reference values for drag and lift of a two dimensional time-dependent flow around a cylinder. *Int. J. Num. Meth. Fluids*, 44:777–788, 2004.

[39] V. John. *Finite element methods for incompressible flow problems*. Springer, 2016.

[40] V. John, A. Linke, C. Merdon, M. Neilan, and L. G. Rebholz. On the divergence constraint in mixed finite element methods for incompressible flows. *SIAM Rev.*, 2016.

[41] V. John, A. Linke, C. Merdon, M. Neilan, and L. G. Rebholz. On the divergence constraint in mixed finite element methods for incompressible flows. *SIAM Review*, 59(3):492–544, 2017.

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

[43] K. K. Lin and F. Lu. Data-driven model reduction, Wiener projections, and the Mori-Zwanzig formalism. *arXiv preprint arXiv:1908.07725*, 2019.

[44] J.-C. Loiseau and S. L. Brunton. Constrained sparse Galerkin regression. *J. Fluid Mech.*, 838:42–67, 2018.

[45] F. Lu, K. K. Lin, and A. J. Chorin. Data-based stochastic model reduction for the Kuramoto–Sivashinsky equation. *Phys. D*, 340:46–57, 2017.

[46] A. J. Majda and N. Chen. Model error, information barriers, state estimation and prediction in complex multiscale systems. *Entropy*, 20(9):644, 2018.

[47] A. J. Majda and J. Harlim. Physics constrained nonlinear regression models for time series. *Nonlinearity*, 26(1):201, 2012.

[48] A. J. Majda and X. Wang. *Nonlinear dynamics and statistical theories for basic geophysical flows*. Cambridge University Press, Cambridge, 2006.

[49] R. Maulik, A. Mohan, B. Lusch, S. Madireddy, and P. Balaprakash. Time-series learning of latent-space dynamics for reduced-order model closure. *arXiv preprint, http://arxiv.org/abs/1906.07815*, 2019.

[50] M. Mohebujjaman, L. G. Rebholz, and T. Iliescu. Physically-constrained data-driven correction for reduced order modeling of fluid flows. *Int. J. Num. Meth. Fluids*, 89(3):103–122, 2019.

[51] M. Mohebujjaman, L. G. Rebholz, X. Xie, and T. Iliescu. Energy balance and mass conservation in reduced order models of fluid flows. *J. Comput. Phys.*, 346:262–277, 2017.

[52] C. Mou, H. Liu, D. R. Wells, and T. Iliescu. Data-driven correction reduced order models for the quasi-geostrophic equations: A numerical investigation. *Int. J. Comput. Fluid Dyn.*, 2020. to appear.

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

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

[55] A. A. Oberai and J. Jagalur-Mohan. Approximate optimal projection for reduced-order models. *Int. J. Num. Meth. Engng.*, 105(1):63–80, 2016.

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

[57] S. Pagani, A. Manzoni, and K. Carlberg. Statistical closure modeling for reduced-order models of stationary systems by the ROMES method. *arXiv preprint, http://arxiv.org/abs/1901.02792*, 2019.

[58] E. J. Parish, C. Wentland, and K. Duraisamy. The adjoint Petrov-Galerkin method for non-linear model reduction. *arXiv preprint arXiv:1810.03455*, 2019.

[59] B. Peherstorfer and K. Willcox. Data-driven operator inference for nonintrusive projection-based model reduction. *Comput. Methods Appl. Mech. Engrg.*, 306:196–215, 2016.

[60] S. Perotto, A. Reali, P. Rusconi, and A. Veneziani. HIGAMod: A Hierarchical IsoGeometric Approach for MODel reduction in curved pipes. *Comput. & Fluids*, 142:21–29, 2017.

[61] A. Quarteroni, A. Manzoni, and F. Negri. *Reduced Basis Methods for Partial Differential Equations: An Introduction*, volume 92. Springer, 2015.

[62] S. M. Rahman, S. Pawar, O. San, A. Rasheed, and T. Iliescu. A nonintrusive reduced order modeling framework for quasigeostrophic turbulence. *Phys. Rev. E*, 100:053306, 2019.

[63] S. M. Rahman, O. San, and A. Rasheed. A hybrid approach for model order reduction of barotropic quasi-geostrophic turbulence. *Fluids*, 3(4):86, 2018.

[64] U. Rasthofer and V. Gravemeier. Recent developments in variational multiscale methods for large-eddy simulation of turbulent flow. *Arch. Comput. Method. E.*, 25(3):647–690, 2018.

[65] L. Rebholz and M. Xiao. Improved accuracy in algebraic splitting methods for Navier-Stokes equations. *SIAM J. Sci. Comput.*, 39(4):A1489–A1513, 2017.

[66] T. C. Rebollo, E. D. Avila, M. G. Mármol, F. Ballarin, and G. Rozza. On a certified Smagorinsky reduced basis turbulence model. *SIAM J. Numer. Anal.*, 55(6):3047–3067, 2017.

[67] R. Reyes and R. Codina. Projection-based reduced order models for flow problems: A variational multiscale approach. *Comput. Methods Appl. Mech. Engrg.*, 363:112844, 2020.

[68] R. Reyes, R. Codina, J. Baiges, and S. Idelsohn. Reduced order models for thermally coupled low mach flows. *Adv. Model. Simul. Eng. Sci.*, 5(1):28, 2018.

[69] J. P. Roop. A proper-orthogonal decomposition variational multiscale approximation method for a generalized Oseen problem. *Adv. Numer. Anal.*, 2013, 2013.

[70] P. Sagaut. *Large Eddy Simulation for Incompressible Flows.* Scientific Computation. Springer-Verlag, Berlin, third edition, 2006.

[71] O. San and T. Iliescu. A stabilized proper orthogonal decomposition reduced-order model for large scale quasigeostrophic ocean circulation. *Adv. Comput. Math.*, pages 1289–1319, 2015.

[72] O. San and R. Maulik. Extreme learning machine for reduced order modeling of turbulent geophysical flows. *Phys. Rev. E*, 97(4):042322, 2018.

[73] O. San and R. Maulik. Machine learning closures for model order reduction of thermal fluids. *Appl. Math. Model.*, 60:681–710, 2018.

[74] O. San and R. Maulik. Neural network closures for nonlinear model order reduction. *Adv. Comput. Math.*, 44(6):1717–1750, 2018.

[75] 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.

[76] T. P. Sapsis and P. F. J. Lermusiaux. Dynamically orthogonal field equations for continuous stochastic dynamical systems. *Phys. D*, 238(23-24):2347–2360, 2009.

[77] S. Shah and E. Bou-Zeid. Very-large-scale motions in the atmospheric boundary layer educed by snapshot proper orthogonal decomposition. *Bound.-Lay. Meteorol.*, 153(3):355–387, 2014.

[78] G. Stabile, F. Ballarin, G. Zuccarino, and G. Rozza. A reduced order variational multiscale approach for turbulent flows. *Adv. Comput. Math.*, pages 1–20, 2019.

[79] R. Ştefănescu, A. Sandu, and I. M. Navon. POD/DEIM reduced-order strategies for efficient four dimensional variational data assimilation. *J. Comput. Phys.*, 295:569–595, 2015.

[80] K. Taira, M. S. Hemati, S. L. Brunton, Y. Sun, K. Duraisamy, S. Bagheri, S. T. M. Dawson, and C.-A. Yeh. Modal analysis of fluid flows: Applications and outlook. *AIAA J.*, pages 1–25, 2019.

[81] A. Tello, R. Codina, and J. Baiges. Fluid structure interaction by means of variational multiscale reduced order models. *Int. J. Num. Meth. Eng.*, 2019.

[82] G. K. Vallis. *Atmospheric and Oceanic Fluid Dynamics: Fundamentals and Large-scale Circulation*. Cambridge University Press, 2006.

[83] Z. Y. Wan, P. Vlachas, P. Koumoutsakos, and T. Sapsis. Data-assisted reduced-order modeling of extreme events in complex dynamical systems. *PloS One*, 13(5):e0197704, 2018.

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

[85] X. Xie, M. Mohebujjaman, L. G. Rebholz, and T. Iliescu. Data-driven filtered reduced order modeling of fluid flows. *SIAM J. Sci. Comput.*, 40(3):B834–B857, 2018.

[86] M. Zhang and R. J. A. M. Stevens. Characterizing the coherent structures within and above large wind farms. *Bound.-Lay. Meteorol.*, 174:61–80, 2020.

Chapter 4

Long-Time Reynolds Averaging of ROMs for Fluid Flows

This chapter has been published in the journal *Mathematics in Engineering*. [*]

In that paper, my contribution was performing numerical experiments for the Burgers equation in Section 4.4.

[*]L.C. Berselli, T. Iliescu, **B. Koc**, and R. Lewandowski. Long-time Reynolds averaging of reduced order models for fluid flows: Preliminary results. Mathematics in Engineering, 2(1):1–25, 2020.

In this chapter, we combine some results on the long-time averaging of fluid equations with the recent techniques for reduced order model (ROM) development. In this preliminary work we start proving some analytical results that characterize the time-averaged effect of the exchange of energy between various modes, both in the case of the computable decomposition made with proper orthogonal decomposition (POD) type basis functions and with the abstract basis made with eigenfunctions. We will show that the results obtainable with a generic (but computable) basis are less precise than those obtainable with the abstract spectral basis, the difference coming from the lack of orthogonality of the gradients of the POD basis functions.

We then provide a few numerical examples. Concerning the analytical results we will prove partial results for the energy exchange between large and small scales, showing the difference between the use of a spectral type basis and a POD one. In particular, we are interested in results connected to the statistical equilibrium problem, which can be deduced in a computable way by a long-time averaging of the solutions. We want to investigate the possible forward and backward average transfer of energy. The properties of a turbulent flow are computable (and relevant) only in an average sense. In this respect, we want to follow the classical approach dating back to Stokes, Reynolds, and Prandtl of considering long-time averages of the solution as the key quantity to be computed or observed. Therefore, we do not need to consider statistical averaging and link it with time averaging by means of (unproved) ergodic hypotheses.

To introduce the problem that we will consider, we recall that a Newtonian incompressible flow (with constant density) can be described by the Navier-Stokes equations (NSE in the sequel)

$$\begin{cases} \partial_t \boldsymbol{u} - \nu \Delta \boldsymbol{u} + (\boldsymbol{u} \cdot \nabla)\, \boldsymbol{u} + \nabla p = \boldsymbol{f} & \text{in } \Omega \times (0,T), \\ \nabla \cdot \boldsymbol{u} = 0 & \text{in } \Omega \times (0,T), \\ \boldsymbol{u} = \boldsymbol{0} & \text{on } \Gamma \times (0,T), \\ \boldsymbol{u}(\cdot,0) = \boldsymbol{u}(0) & \text{in } \Omega, \end{cases} \tag{4.1}$$

with Dirichlet conditions when the motion takes place in a smooth and bounded domain $\Omega \subset \mathbb{R}^3$ with solid walls $\Gamma := \partial\Omega$. The unknowns are the velocity field $\boldsymbol{u}$ and the scalar pressure p, while the positive constant $\nu > 0$ is the kinematic viscosity. The key parameter to detect the nature of the problem is the non-dimensional Reynolds number, which is defined as

$$Re = \frac{UL}{\nu},$$

where U and L are a characteristic velocity and length of the problem. In realistic problems, the Reynolds number can be extremely large (in many cases of the order of 10^8, but up to the order of 10^{20} in certain geophysical problems). For simplicity in the notation, we use the viscosity as a control parameter and assume that it is very small. Hence, the effect of the regularization (similar to the diffusion in heat transfer) due to the Laplacian is negligible and the behavior of solutions is really turbulent and rather close to the motion of ideal fluids.

Due to the well-known difficulties in performing direct numerical simulations (DNS), it is nowadays a well-established technique that of trying to reduce the computational efforts by simulating only the largest scales, which, for limited computational and experimental resources, are the only ones computable and observable.

In this framework, the large eddy simulation (LES) methods, which emerged in the last 30 years, are among the most popular, and they found a very relevant role within both theorists' and practitioners' communities. For recent LES reviews, see, for instance, the monographs [2, 5, 27, 39].

The LES methods are, in many cases, very well developed and both theoretically and computationally appealing, especially for problems without boundaries, but many difficulties and open problems arise when facing solid boundaries. In most cases the design of efficient LES methods is guided by deep properties of the solutions, as emerging from deep theorems of mathematical analysis. Furthermore, the ultimate goal of having a golden standard is far from being obtained, and large families of methods (wave-number asymptotics, differential filters, α-models) attracted the interest of different communities, spanning from the pure mathematicians, to the applied geophysicist and mechanical engineers, to the computational practitioners. For these reasons, we believe that it is important to have some well-defined and clearly stated guidelines, that can be adapted to different problems. In this way the methods can be improved with insight not only from experts in modeling, but also from mathematicians, physicists, and computational scientists.

In this respect, we point out that very recently the use of other (more flexible and computationally simpler) ways of finding approximate systems has become very popular. The LES methods itself can be specialized or even glued with other ways of determining much smaller approximate systems, which are computable in a very efficient way. For instance, reduced order modeling is increasingly becoming an accepted paradigm, in which applications to fluids are still being developed [15, 25, 35, 36, 38].

The basic ansatz at the basis of the use of these models is the approximation of the velocity by a truncation of the series

$$\boldsymbol{u} = \sum_{k=1}^{\infty} \boldsymbol{u}_k \boldsymbol{w}_k,$$

where $\{\boldsymbol{w}_k\}_{k\in\mathbb{N}}$ is a basis constructed by using the POD, not necessarily made with eigenfunctions of the Stokes operator, and the coefficients of the L^2-projections are evaluated as follows

$$\boldsymbol{u}_k = \frac{\int_\Omega \boldsymbol{u}\cdot\boldsymbol{w}_k\,dx}{\int_\Omega |\boldsymbol{w}_k|^2\,dx}.$$

The appealing property of this approach is that the choice of the basis is adapted to and determined by the problem itself. Generally the basis is chosen after a preliminary numerical computation, hence it contains at least the basic features of the solution and geometry of the problem to be studied. The other basic fact is that the kinetic energy of the problem is the

key quantity under consideration; in fact the L^2-projection is generally used to determine the approximate velocity and also the energy content in the basis construction. To determine the number $r \in \mathbb{N}$ such that the solution is projected over the space generated by the (orthogonal) functions $V_r := \text{span}\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_r\}$, generally it is assumed that the projection of the solution over V_r contains a large percentage (say 80%) of the total kinetic energy of the underlying system.

It turns out that a basis associated with the problem at hand can greatly improve the effectiveness of the ROM. Its proper choice can be of great interest in applications to fluid flows [41, 42, 44]. The present paper combines mathematical results on the long-time behavior of fluid flows (especially in the case of statistical equilibrium) with reduced order modeling. The main goal of this approach is that it provides a mathematical description of both the long-time averages of ROMs and the energy exchange between ROM modes. Furthermore, preliminary numerical results that support the theoretical developments are presented. Specifically, we are extending to the POD setting the results for statistical solutions by Foias et al. [11, 12] and those more recently obtained for time-averages in [3, 28]. In this respect, the main theoretical results of this paper, stated in Theorems 4.6 and 4.7 below, can be viewed as a spectral version of the results of [3, 28]. These results show that low frequency modes yield a Reynolds stress that is dissipative in the mean, its total spatial mean work being larger than the long time average of the dissipation of the fluctuations, which is consistent with observations and results in [3, 11]. However, the analysis shows that the triad interaction between high and low frequency modes yields an additional non positive term in the budget between the Reynolds stress of high modes and the corresponding mean dissipation. This term may be non dissipative and may permit an inverse energy cascade, which is not in contradiction with the fact that the total Reynolds stress is dissipative in mean.

The paper is organized as follows: In Section 4.1, we outline the general framework for ROMs of fluid flows, and we display the exchange of energy between large scales and small scales for two ROM bases: the POD and the Stokes eigenfunctions. In Section 4.2, we present some preliminaries on long-time averages. In Section 4.3, we prove the main theoretical results for the average transfer of energy for ROMs constructed with the POD and the Stokes eigenfunctions. In Section 4.4, we perform a preliminary numerical study in which we investigate the theoretical results in the numerical simulation of the one-dimensional Burgers equation. Finally, in Section 4.5, we draw conclusions and outline future research directions.

4.1 Reduced order modeling

As outlined in the introduction, one key quantity in the pure and applied analysis of the Navier-Stokes equations is the kinetic energy, since it is a meaningful physical quantity and the analysis of its budget is at the basis of abstract existence results for weak solutions (cf. Constantin and Foias [6]) and also of the conventional turbulence theories of Kolmogorov [19].

It is well-known that after testing the NSE (4.1) by $\boldsymbol{u}$ and integrating over the space-time, one (formally) obtains the global energy balance

$$\frac{1}{2}\|\boldsymbol{u}(t)\|^2 + \nu \int_0^t \|\nabla \boldsymbol{u}(s)\|^2 \, ds = \frac{1}{2}\|\boldsymbol{u}(0)\|^2 + \int_0^t \int_\Omega \boldsymbol{f} \cdot \boldsymbol{u} \, dxds.$$

At present, it is possible to prove that the above balance holds true with the sign of "less or equal" for the class of weak (or turbulent) solutions for which there are results of existence globally in time, without restrictions on the viscosity and on the size of square summable initial velocity $\boldsymbol{u}(0)$ and external force $\boldsymbol{f}$. It is of fundamental importance in many problems in pure mathematics to understand under which hypotheses the equality holds true. We are now focusing on the "global energy" which is an averaged quantity, since it is the integral of the square modulus of the velocity over the entire domain. We also point out that at the other extreme one can deduce, without the integration over the domain, the point-wise relation

$$\partial_t \frac{|\boldsymbol{u}|^2}{2} + |\nabla \boldsymbol{u}|^2 - \Delta \frac{|\boldsymbol{u}|^2}{2} + \operatorname{div} \left(\frac{\boldsymbol{u}|\boldsymbol{u}|^2}{2} + p\,\boldsymbol{u} \right) = \boldsymbol{f} \cdot \boldsymbol{u}.$$

In between there is the so called "local energy" which can be obtained by multiplying the NSE by $\boldsymbol{u}\,\phi$, where ϕ is a bump function, before integrating in the space-time variables. The goal is to show that

$$\int_0^T \int_\Omega |\nabla \boldsymbol{u}|^2 \phi \, dxdt = \int_0^T \int_\Omega \left[\frac{|\boldsymbol{u}|^2}{2} (\partial_t \phi + \Delta \phi) + \left(\frac{|\boldsymbol{u}|^2}{2} + p \right) \boldsymbol{u} \cdot \nabla \phi + \boldsymbol{f} \cdot \boldsymbol{u}\, \phi \right] dxdt$$

holds (at least with the inequality sign) for all smooth scalar functions $\phi \in C_0^\infty((0,T) \times \Omega)$ such that $\phi \geq 0$. The validity of such an inequality is one of the requirements to prove *partial regularity results*, but it is also one of the conditions to be satisfied by weak solutions constructed by numerical or LES methods. In this respect, see Guermond, Oden and Prudhomme [14] and [4].

In this paper we study the global energy in the perspective that it can be reconstructed in a computable way or it can be well approximated by the POD basis functions $\{\boldsymbol{w}_k\}$.

The fact that the functions $\{\boldsymbol{w}_k\}$ can be constructed to be orthonormal with respect to the scalar product in $(L^2(\Omega)^3, \|\cdot\|)$ allows us to evaluate the kinetic energy easily by the following numerical series

$$\mathrm{E}(\boldsymbol{u}) = \frac{1}{2} \sum_{k=1}^{\infty} \|\boldsymbol{u}_k\|^2.$$

Since we are going to use only a reduced number of ROM modes, it is relevant to consider the energy contained in functions described by a restricted set of indices. Hence, following the notation in [12], if we define

$$\boldsymbol{u}_{m',m''} := \sum_{k=m'}^{m''} \boldsymbol{u}_k \boldsymbol{w}_k,$$

then the kinetic energy content of $\boldsymbol{u}_{m',m''}$ is simply evaluated as follows

$$\mathrm{E}(\boldsymbol{u}_{m',m''}) = \frac{1}{2} \sum_{k=m'}^{m''} \|\boldsymbol{u}_k\|^2.$$

We want to investigate the energy transfer between the various modes, together with averaged long-time behavior associated with this splitting.

We are going to adapt well-known studies on the decomposition in small and large eddies. This would be the case if the functions $\boldsymbol{w}_k$ are chosen to be the eigenfunctions of the Stokes operator, hence associated with large and small frequencies. In our setting the basis is determined by the solution of a simplified problem, which can be treated computationally, and the basis functions are orthonormal in $L^2(\Omega)$, but we cannot expect that their gradients are also orthogonal.

For the NSE, the standard ROM is constructed as follows:

(i) choose modes $\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_d\}$, which represent the recurrent spatial structures of the given flow;

(ii) choose the dominant modes $\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_m\}$, with $m \leq d$, as basis functions for the ROM;

(iii) use a Galerkin truncation $\boldsymbol{u}_m = \sum_{j=1}^{m} a_j \, \boldsymbol{w}_j$;

(iv) replace $\boldsymbol{u}$ with $\boldsymbol{u}_m$ in the NSE;

(v) use a Galerkin projection of NSE ($\boldsymbol{u}_m$) onto the ROM space $V_m := \mathrm{span}\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_m\}$ to obtain a low-dimensional dynamical system, which represents the ROM:

$$\dot{a} = A\, a + a^\top B\, a\,,$$

where a is the vector of unknown ROM coefficients and A, B are ROM operators;

(vi) in an offline stage, compute the ROM operators;

(vii) in an online stage, repeatedly use the ROM (for various parameter settings and/or longer time intervals).

Hence, there is a very natural splitting of the velocity field $\boldsymbol{u}$ into two components, the part coherent with the basis expansion associated with the more energetic modes ($\boldsymbol{y}$), and the remainder ($\boldsymbol{z}$). This can be formalized as follows:

$$\boldsymbol{u} = \boldsymbol{y} + \boldsymbol{z},$$

where

$$\boldsymbol{y} = \sum_{k=1}^{m} \boldsymbol{u}_k \boldsymbol{w}_k = P_m \boldsymbol{u} \qquad \text{and} \qquad \boldsymbol{z} = \sum_{k=m+1}^{+\infty} \boldsymbol{u}_k \boldsymbol{w}_k = (I - P_m)\, \boldsymbol{u} =: Q_m\, \boldsymbol{u}. \tag{4.2}$$

In (4.2), $m \in \mathbb{N}$ can be selected computationally (based, e.g., on the relative kinetic energy content in the first m POD-modes, but other choices relative to the enstrophy are possible) in order to have a significant amount of the energy content of the flow in $\boldsymbol{y}$. Furthermore, P_m is the projection operator over the subspace V_m spanned by the first m-functions $\{\boldsymbol{w}_k\}_{k=1,\dots,m}$.

We observe that we are considering the functions $\{\boldsymbol{w}_k\}_k$ as divergence-free, at least in a weak and/or approximate sense. Even if generally they are not "exactly divergence-free", numerically we can consider that they have vanishing divergence, by assuming that the solution of the problems used to construct the basis is accurate enough to have negligible divergence. Hence, in the computations that will follow, we will drop the pressure terms by a standard Leray projection. It will be nevertheless interesting to extend our study to bases that are not divergence-free, e.g., those derived by the combination of ROM with artificial compressibility methods [8, 9, 13].

In addition, we consider the external force as stationary, that is $\boldsymbol{f} = \boldsymbol{f}(\mathbf{x}) \in L^2(\Omega)^3$ and we look for conditions holding at statistical equilibrium. Our purpose is to determine –if possible– the long-time behavior of $\boldsymbol{y}$ and to analyze the energy budget between low and high modes in the orthogonal decomposition determined by the functions $\boldsymbol{w}_k$.

As usual in many problems fluid mechanics, we use the Hilbert space functional setting with

$$\begin{aligned}
&\mathcal{V} = \{\boldsymbol{\varphi} \in \mathcal{D}(\Omega)^3,\ \nabla \cdot \boldsymbol{\varphi} = 0\},\\
&H = \left\{\boldsymbol{u} \in L^2(\Omega)^3,\ \nabla \cdot \boldsymbol{u} = 0,\ \boldsymbol{u} \cdot \mathbf{n} = 0 \text{ on } \Gamma\right\},\\
&V = \left\{\boldsymbol{u} \in H_0^1(\Omega)^3,\ \nabla \cdot \boldsymbol{u} = 0\right\},
\end{aligned}$$

where $\mathbf{n}$ denotes the outward normal unit vector. Moreover, V' is the topological dual space of V. We will also denote by $< \cdot, \cdot >$ the duality pairing between V and V'. We recall that $\mathcal{V}$ is dense in H and V for their respective topologies [10, 29].

Once we project $L^2(\Omega)^3$ over the subspace H of divergence-free and tangential vector fields by means of the Leray projection operator P, we have the following abstract (functional) equation in H

$$\frac{d\boldsymbol{u}}{dt} + \nu A\boldsymbol{u} + B(\boldsymbol{u}, \boldsymbol{u}) = \mathrm{P}\, \boldsymbol{f},$$

where $A := -\mathrm{P}\,\Delta$, while $B(\boldsymbol{u},\boldsymbol{u}) := \mathrm{P}\,((\boldsymbol{u}\cdot\nabla)\,\boldsymbol{u})$. As usual in this analysis (see for instance [12]), we can start by assuming that the input force can be decomposed within a finite sum of basis functions (or that it belongs to V_m, which will be clarified in the next section, in particular by Theorem 4.2), hence

$$\mathbf{P}_m\,\boldsymbol{f} = \boldsymbol{f}.$$

We split the Navier-Stokes equations into a coupled system for $\boldsymbol{y} \in P_m H$ and $\boldsymbol{z} \in (P_m H)^{\perp} = Q_m H$ as follows

$$\begin{aligned} &\frac{d\boldsymbol{y}}{dt} - \nu\, P_m(\Delta\boldsymbol{u}) + P_m B(\boldsymbol{y}+\boldsymbol{z},\boldsymbol{y}+\boldsymbol{z}) = P_m\,\boldsymbol{f},\\ &\frac{d\boldsymbol{z}}{dt} - \nu\, Q_m(\Delta\boldsymbol{u}) + Q_m B(\boldsymbol{y}+\boldsymbol{z},\boldsymbol{y}+\boldsymbol{z}) = \mathbf{0}, \end{aligned} \tag{4.3}$$

where we used that both P_m and Q_m commute with the time derivative.

Once we evaluate the kinetic energy, since $P_m\boldsymbol{y} = \boldsymbol{y}$ and $Q_m\,\boldsymbol{z} = \boldsymbol{z}$ we get (by integrating by parts and by using the fact that functions vanish at the boundary) that

$$\begin{aligned} -\nu\int_\Omega P_m(\Delta\boldsymbol{u})\cdot\boldsymbol{y}\,dx &= -\nu\int_\Omega(\Delta\boldsymbol{u})\cdot\boldsymbol{y}\,dx = -\nu\int_\Omega(\Delta\boldsymbol{y}+\Delta\boldsymbol{z})\cdot\boldsymbol{y}\,dx = \nu\,\|\nabla\boldsymbol{y}\|^2 + \nu\int_\Omega\nabla\boldsymbol{y}:\nabla\boldsymbol{z}\,dx,\\ -\nu\int_\Omega Q_m(\Delta\boldsymbol{u})\cdot\boldsymbol{z}\,dx &= -\nu\int_\Omega(\Delta\boldsymbol{u})\cdot\boldsymbol{z}\,dx = -\nu\int_\Omega(\Delta\boldsymbol{y}+\Delta\boldsymbol{z})\cdot\boldsymbol{z}\,dx = \nu\,\|\nabla\boldsymbol{z}\|^2 + \nu\int_\Omega\nabla\boldsymbol{y}:\nabla\boldsymbol{z}\,dx. \end{aligned}$$

In this way we can obtain the following equality and inequality

$$\begin{aligned} &\frac{1}{2}\frac{d}{dt}\|\boldsymbol{y}\|^2 + \nu\|\nabla\boldsymbol{y}\|^2 + \nu(\nabla\boldsymbol{y},\nabla\boldsymbol{z}) + (B(\boldsymbol{y}+\boldsymbol{z},\boldsymbol{y}+\boldsymbol{z}),\boldsymbol{y}) = (\boldsymbol{f},\boldsymbol{y}),\\ &\frac{1}{2}\frac{d}{dt}\|\boldsymbol{z}\|^2 + \nu\|\nabla\boldsymbol{z}\|^2 + \nu(\nabla\boldsymbol{y},\nabla\boldsymbol{z}) + (B(\boldsymbol{y}+\boldsymbol{z},\boldsymbol{y}+\boldsymbol{z}),\boldsymbol{z}) \le \mathbf{0}, \end{aligned} \tag{4.4}$$

These are the two basic balance equations that we will use to infer the behavior and transfer of the kinetic energy between $\boldsymbol{y}$ and $\boldsymbol{z}$. Notice that the balance relation for $\boldsymbol{y}$, involving just a finite combination of rather smooth functions is an equality, while the second one is an inequality. In fact, the second one can be derived by a limiting argument and in the limit the lower semi-continuity of the norm will produce the inequality.

Since the tri-linear term $(B(\boldsymbol{u},\boldsymbol{u}),\boldsymbol{w})$ is skew-symmetric with respect to the last two variables, we obtain from (4.4)

$$\begin{aligned} &\frac{1}{2}\frac{d}{dt}\mathrm{E}(\boldsymbol{y}) + \nu\|\nabla\boldsymbol{y}\|^2 + \nu(\nabla\boldsymbol{y},\nabla\boldsymbol{z}) = (B(\boldsymbol{y},\boldsymbol{y}),\boldsymbol{z}) - (B(\boldsymbol{z},\boldsymbol{z}),\boldsymbol{y}) + (\boldsymbol{f},\boldsymbol{y}),\\ &\frac{1}{2}\frac{d}{dt}\mathrm{E}(\boldsymbol{z}) + \nu\|\nabla\boldsymbol{z}\|^2 + \nu(\nabla\boldsymbol{y},\nabla\boldsymbol{z}) \le -(B(\boldsymbol{y},\boldsymbol{y}),\boldsymbol{z}) + (B(\boldsymbol{z},\boldsymbol{z}),\boldsymbol{y}). \end{aligned} \tag{4.5}$$

This is a formal setting, which is obviously true for strong solutions of the NSE, where the inequality in (4.5) is an equality. When considering weak solutions, the integral $(B(\boldsymbol{z},\boldsymbol{z}),\boldsymbol{z})$ might be not defined in $L^1(0,T)$ for regularity issues. However, one can still rigorously derive (4.5) by a double frequency truncation or a regularization of the operator B by considering $(B(\boldsymbol{z}\star\rho_\varepsilon,\boldsymbol{z}),\boldsymbol{z})$ for a given standard mollifier ρ_ε and passing to the limit when $\varepsilon\to 0$. Details are standard and out of the scope of the present paper.

We observe that $-(B(\boldsymbol{y},\boldsymbol{y}),\boldsymbol{z})$ is the energy flux induced in the more energetic terms by the inertial forces associated to less energetic modes, while $-(B(\boldsymbol{z},\boldsymbol{z}),\boldsymbol{y})$ is the energy flux induced in the less energetic terms by the inertial forces associated to more energetic modes. In a schematic way we can decompose the rate of transfer of kinetic energy $\mathrm{e}_m(\boldsymbol{u})$ into two terms as follows

$$\mathrm{e}_m(\boldsymbol{u}) := e^{\uparrow}(\boldsymbol{u}) - e^{\downarrow}(\boldsymbol{u}) \qquad \text{with} \qquad e^{\uparrow}(\boldsymbol{u}) := -(B(\boldsymbol{y},\boldsymbol{y}),\boldsymbol{z}), \qquad e^{\downarrow}(\boldsymbol{u}) := (B(\boldsymbol{z},\boldsymbol{z}),\boldsymbol{y}). \tag{4.6}$$

We also use the following notation:

$$\mathcal{E}_m(\boldsymbol{u}) := -\nu(\nabla\boldsymbol{y},\nabla\boldsymbol{z}). \tag{4.7}$$

Hence, we can rewrite (4.5) as follows

$$\begin{aligned} &\frac{1}{2}\frac{d}{dt}\mathrm{E}(\boldsymbol{y}) + \nu\|\nabla\boldsymbol{y}\|^2 = \mathcal{E}_m(\boldsymbol{u}) - \mathrm{e}_m(\boldsymbol{u}) + (\boldsymbol{f},\boldsymbol{y}),\\ &\frac{1}{2}\frac{d}{dt}\mathrm{E}(\boldsymbol{z}) + \nu\|\nabla\boldsymbol{z}\|^2 \le \mathcal{E}_m(\boldsymbol{u}) + \mathrm{e}_m(\boldsymbol{u}). \end{aligned} \tag{4.8}$$

Remark 4.1. We recall that apart from extremely simple geometries and provided one is willing to use in a systematic way special functions as the Bessel ones or the spherical harmonics (which are nevertheless time consuming in their evaluation), the explicit calculations in numerical tests will not be so easy to be obtained in a precise and efficient way. Hence, the solution of (4.3) and the long-time integration of its solution pose hard numerical problems.

We point out for the reader that we have a first fundamental difference with respect to the classical splitting based on the use of a spectral basis (which will be recalled in Section 4.1.1), where the latter integral vanishes exactly. For this reason, in the next section we will show the derivation of the corresponding system of equations, which holds when the eigenfunctions are used.

4.1.1 On the spectral decomposition

In this section, we compare the results of the previous section with the well-established ones that can be proved if the spectral decomposition (i.e., that made with eigenfunctions of the

Stokes operator $\{\mathcal{W}_k\}$) is used instead of utilizing a generic POD basis. We recall that, by classical results about compact operators in Hilbert spaces there exists a sequence of smooth functions $\{\mathcal{W}_k\}$ (and their regularity is depending on the smoothness of the bounded domain Ω) and an increasing sequence of positive numbers $\{\lambda_k\}$ such that

$$A\mathcal{W}_k = \lambda_k \mathcal{W}_k \qquad \text{and} \qquad \int_\Omega \mathcal{W}_k \cdot \mathcal{W}_j \, dx = \delta_{kj}.$$

Since each function $\mathcal{W}_k$ solves the following Stokes system $A\mathcal{W}_k = \lambda_k \mathcal{W}_k$, it follows by an integration by parts that

$$\int_\Omega \nabla \mathcal{W}_k : \nabla \mathcal{W}_j \, dx = 0 \qquad \text{for } k \neq j,$$

hence also the V-orthogonality of the family $\{\mathcal{W}_k\}_{k\in\mathbb{N}}$.

We consider now the usual decomposition by eigenfunctions associated with low and high frequencies

$$\boldsymbol{u} = \boldsymbol{y} + \boldsymbol{z} := \sum_{k=1}^{m} c_k \mathcal{W}_k + \sum_{k=m+1}^{\infty} c_k \mathcal{W}_k = \mathbf{P}_m \, \boldsymbol{u} + \mathbf{Q}_m \, \boldsymbol{u},$$

where $\mathbf{P}_m$ is the projection over the subspace generated by $\{\mathcal{W}_k\}_{k=1,\dots,m}$. Our main result is based on a standard result about the projector $\mathbf{P}_m$, that can be found in [30, App. A.4,Thm. 4.11]:

Theorem 4.2. *The projector $\mathbf{P}_m$ can be defined as a continuous endomorphism over V, H and V', and one has*

$$\|\mathbf{P}_m\|_{\mathcal{L}(V,V)} \leq 1, \qquad \|\mathbf{P}_m\|_{\mathcal{L}(H,H)} \leq 1, \qquad \|\mathbf{P}_m\|_{\mathcal{L}(V',V')} \leq 1.$$

The result is mainly based on the regularity of solutions of elliptic equations, and thanks to this fact, it is possible to decompose the equations for the velocity, which yields,

$$\begin{aligned} -\nu \int_\Omega \mathbf{P}_m(\Delta \boldsymbol{u}) \cdot \boldsymbol{y} \, dx = -\nu \int_\Omega \Delta \boldsymbol{y} \cdot \boldsymbol{y} \, dx = \nu \, \|\nabla \boldsymbol{y}\|^2, \\ -\nu \int_\Omega \mathbf{Q}_m \, (\Delta \boldsymbol{u}) \cdot \boldsymbol{z} \, dx = -\nu \int_\Omega \Delta \boldsymbol{z} \cdot \boldsymbol{z} \, dx = \nu \, \|\nabla \boldsymbol{z}\|^2, \end{aligned}$$

since $\mathbf{P}_m \Delta \boldsymbol{u} = \Delta \mathbf{P}_m \, \boldsymbol{u} = \Delta \boldsymbol{y}$ and also $\mathbf{Q}_m(\Delta \boldsymbol{u}) = \Delta \mathbf{Q}_m \, \boldsymbol{u} = \Delta \boldsymbol{z}$.

Thus, we directly obtain the system

$$\begin{aligned} &\frac{1}{2}\frac{d}{dt}\|\boldsymbol{y}\|^2 + \nu \|\nabla \boldsymbol{y}\|^2 + (B(\boldsymbol{y}+\boldsymbol{z}, \boldsymbol{y}+\boldsymbol{z}), \boldsymbol{y}) = (\boldsymbol{f}, \boldsymbol{y}), \\ &\frac{1}{2}\frac{d}{dt}\|\boldsymbol{z}\|^2 + \nu \|\nabla \boldsymbol{z}\|^2 + (B(\boldsymbol{y}+\boldsymbol{z}, \boldsymbol{y}+\boldsymbol{z}), \boldsymbol{z}) \leq \mathbf{0}, \end{aligned} \tag{4.9}$$

which can be rewritten also as

$$\begin{aligned} &\frac{1}{2}\frac{d}{dt}\mathrm{E}(\boldsymbol{y}) + \nu\|\nabla \boldsymbol{y}\|^2 = -\mathrm{e}_m(\boldsymbol{u}) + (\boldsymbol{f}, \boldsymbol{y}), \\ &\frac{1}{2}\frac{d}{dt}\mathrm{E}(\boldsymbol{z}) + \nu\|\nabla \boldsymbol{z}\|^2 \leq \mathrm{e}_m(\boldsymbol{u}). \end{aligned} \tag{4.10}$$

We note that the equations in (4.10) do not contain the term $\mathcal{E}_m(\boldsymbol{u})$, whereas the equations in (4.8) contain $\mathcal{E}_m(\boldsymbol{u})$. This will have important consequences in the analysis in Section 4.3.

4.2 Preliminaries on long-time averages

Since we consider long-time averages for the NSE, we must consider solutions which are global-in-time (defined for all positive times). Due to the well-known open problems related to the NSE, this forces us to restrict ourselves to Leray-Hopf weak solutions [6, 29]. By using a natural setting, we take the initial datum $\boldsymbol{u}(0)$ in H. The classical Leray-Hopf result of existence (but not uniqueness) of a global weak solution $\boldsymbol{u}$ to the NSE holds when $\boldsymbol{f} \in V'$, and the velocity $\boldsymbol{u}$ satisfies

$$\boldsymbol{u} \in L^2(\mathbb{R}_+; V) \cap L^\infty(\mathbb{R}_+; H).$$

Notice that we consider in this paper the case where $\boldsymbol{f}$ is time-independent, for simplicity. However, the following results can be extended to the case where $\boldsymbol{f} = \boldsymbol{f}(t)$ is time dependent, for $\boldsymbol{f}$ belonging to a suitable class (see [3]).

In order to properly set what we mean by long-time-averaging, let $\psi : \mathbb{R}^+ \times \Omega \to \mathbb{R}^N$ be any tensor field related to a given turbulent flow (N being its order). The time-average of ψ over a time interval $[0, t]$ is defined by

$$M_t(\psi)(x) := \frac{1}{t}\int_0^t \psi(s, x)\, ds \qquad \text{for } t > 0. \tag{4.11}$$

According to the standard turbulence modeling process, we then apply the averaging operator M_t to NSE (4.1) and also to (4.3), to study the limits when $t \to +\infty$. We recall that the long-time averages represent one of the few observable and computable quantities associated to a highly variable turbulent flow. We will adopt the following standard notation for the long-time average of any field ψ

$$\overline{\psi}(x) := \lim_{t\to+\infty} M_t(\psi)(x),$$

whenever the limit exists. (Without too much restrictions, we can suppose that the limits we write do exist, at least after extracting sub-sequences leaving the mathematical difficul-

ties, which can be treated with generalized Banach limits, for a more general and abstract framework.) Within this theory we can decompose the velocity as follows

$$\boldsymbol{u} = \overline{\boldsymbol{u}} + \boldsymbol{u}',$$

where $\boldsymbol{u}' := \boldsymbol{u} - \overline{\boldsymbol{u}}$ represents the so-called turbulent fluctuations.

We recall that time-averaging has been introduced by O. Reynolds [37], at least for large values of t, and the ideas have been widely developed by L. Prandtl [32] in the case of turbulent channel flows. The same ideas have been also later considered in the case of fully developed homogeneous and isotropic turbulence, such as grid-generated turbulence. In this case the velocity field is postulated as oscillating around a mean smoother steady state, see for instance G.-K. Batchelor [1]. For further details on the role of time averaging in turbulence, after the work of Stokes and Reynolds, we can recall a few recent papers and books [2, 3, 5, 11, 18, 26, 28], where aspects of computation and modeling are studied.

We now observe that, by taking the time-averages of the NSE we have the following estimates, see [28, Prop. 2.1]

$$\begin{aligned} &\|\boldsymbol{u}(t)\|^2 \le \|\boldsymbol{u}(0)\|^2 \,\mathrm{e}^{-\nu C_P t} + \frac{\|\boldsymbol{f}\|^2}{\nu^2 C_P}\big(1 - \mathrm{e}^{-\nu C_P t}\big), \qquad \forall\, t > 0 \\ &\frac{1}{t}\int_0^t \|\nabla \boldsymbol{u}(s)\|^2 \, ds \le \frac{\|\boldsymbol{f}\|^2}{\nu^2} + \frac{\|\boldsymbol{u}(0)\|^2}{\nu t}, \qquad \forall\, t > 0, \end{aligned} \tag{4.12}$$

where C_P is the best constant in the Poincaré inequality

$$C_P \|\boldsymbol{u}\|^2 \le \|\nabla \boldsymbol{u}\|^2 \qquad \forall\, \boldsymbol{u} \in H_0^1(\Omega).$$

The above inequalities show that both $\|\boldsymbol{u}(t)\|^2$ and $\frac{1}{t}\int_0^t \|\nabla \boldsymbol{u}(s)\|^2 \, ds$ are uniformly bounded for all $t \ge 1$ (any other positive time will be enough), hence we have the following result:

Theorem 4.3 (cf. [3, 28])**.** *Let $\boldsymbol{u}(0) \in H$, $\boldsymbol{f} \in V'$, and let $\boldsymbol{u}$ be a global-in-time weak solution to the NSE (4.1). Then, there exist*

1. *a sequence $\{t_n\}_{n\in\mathbb{N}}$ such that $\lim_{n\to\infty} t_n = +\infty$;*
2. *a vector field $\overline{\boldsymbol{u}} \in V$;*
3. *a vector field $\boldsymbol{B} \in L^{3/2}(\Omega)^3$;*
4. *a second order tensor field $\boldsymbol{\sigma}^{(\mathrm{R})} \in L^3(\Omega)^9$;*

such that it holds:

i) When $n \to \infty$,

$$\begin{aligned} &M_{t_n}(\boldsymbol{u}) \rightharpoonup \overline{\boldsymbol{u}} \qquad \text{weakly in } V, \\ &M_{t_n}\big((\boldsymbol{u}\cdot\nabla)\,\boldsymbol{u}\big) \rightharpoonup \boldsymbol{B} \qquad \text{weakly in } L^{3/2}(\Omega)^3, \\ &M_{t_n}(\boldsymbol{u}'\otimes\boldsymbol{u}') \rightharpoonup \boldsymbol{\sigma}^{(\mathrm{R})} \qquad \text{weakly in } L^3(\Omega)^9; \end{aligned}$$

ii) The Reynolds averaged equations:

$$\begin{cases} (\overline{\boldsymbol{u}}\cdot\nabla)\,\overline{\boldsymbol{u}} - \nu\Delta\overline{\boldsymbol{u}} + \nabla\overline{p} + \nabla\cdot\boldsymbol{\sigma}^{(\mathrm{R})} = \overline{\boldsymbol{f}} & \text{in } \Omega, \\ \nabla\cdot\overline{\boldsymbol{u}} = 0 & \text{in } \Omega, \\ \overline{\boldsymbol{u}} = \boldsymbol{0} & \text{on } \Gamma, \end{cases} \tag{4.13}$$

hold true in the weak sense;

iii) The equality $\boldsymbol{B} - (\overline{\boldsymbol{u}}\cdot\nabla)\,\overline{\boldsymbol{u}} = \nabla\cdot\boldsymbol{\sigma}^{(\mathrm{R})}$ *is valid in* $\mathcal{D}'(\Omega)$*;*

iv) The following energy balance (equality) holds true

$$\nu\,\|\nabla\overline{\boldsymbol{u}}\|^2 + (\nabla\cdot\boldsymbol{\sigma}^{(\mathrm{R})}, \overline{\boldsymbol{u}}) = <\overline{\boldsymbol{f}}, \overline{\boldsymbol{u}}>;$$

v) The tensor $\boldsymbol{\sigma}^{(\mathrm{R})}$ *is dissipative on the average or, more precisely, the following inequality*

$$\epsilon := \nu\,\overline{\|\nabla\boldsymbol{u}'\|^2} \leq \int_\Omega (\nabla\cdot\boldsymbol{\sigma}^{(\mathrm{R})})\cdot\overline{\boldsymbol{u}}\,dx, \tag{4.14}$$

holds true.

It is important to observe that the long-time limit is characterized by the solution of the system (4.13), which is similar to the Navier-Stokes equations, but which contains an extra term, coming from the effect of fluctuations, which has the mean effect of increasing the dissipation.

We observe that this is related to the long-time behavior of solutions close to statistical equilibrium. The study of the long-time behavior dates back to pioneering works of Foias and Prodi on deterministic statistical solutions, see, for instance [12]. Their interest is mainly devoted to finding measure in the space of initial data to be connected with the long-time limits. Here, we follow a slightly different path, as in [3, 28], in order to characterize in a less technical way the long-time behavior, without resorting to any ergodic-type result and also with the perspective that long time averages are computable or at least can be approximated in a clear way.

4.3 Average transfer of energy at equilibrium

Our goal in this section is to characterize in some sense the energy transfer between the two functions $\boldsymbol{y}$ and $\boldsymbol{z}$ of the expansion and to determine –if possible– the sign of $\mathrm{e}_{m(\boldsymbol{u})}$, at least in an average sense. We will consider both the POD case and the spectral one.

4.3.1 The POD case

The point concerning the exchange of energy between low and high modes is in the same spirit as the results recalled in Foias, Manley, Rosa, and Temam [12, Chap. 5] and follows from results obtained in a more heuristic way by Kolmogorov [19].

We first observe that the L^2-orthogonality of the POD decomposition implies that

$$\|\boldsymbol{u}\|^2 = \|\boldsymbol{y}+\boldsymbol{z}\|^2 = \|\boldsymbol{y}\|^2 + \|\boldsymbol{z}\|^2.$$

Hence, from the uniform L^2-bound on $\boldsymbol{u}$ it follows that both $\boldsymbol{y}$ and $\boldsymbol{z}$ are uniformly bounded in time. From this observation we can deduce the following result, reminding that e_m and $\mathcal{E}_m$ are defined by equations (4.6) and (4.7), and M_t is defined by equation (4.11).

Theorem 4.4. *Let $\boldsymbol{z}$ be the projection onto the less energetic POD modes of a weak solution $\boldsymbol{u}$ to the Navier-Stokes equations. Then, there exists a sequence $\{t_n\}$ such that $t_n \to +\infty$ and a field $\overline{\boldsymbol{z}} \in H$ such that*

$$\mathbf{Z}_{t_n} = M_{t_n}(\boldsymbol{z}) \rightharpoonup \overline{\boldsymbol{z}} \qquad \text{weakly in } H, \tag{4.15}$$

and

$$\liminf_{n\to+\infty} M_{t_n}(\mathrm{e}_m(\boldsymbol{u}) + \mathcal{E}_m(\boldsymbol{u})) \geq 0. \tag{4.16}$$

Proof. Let us observe first that by the energy inequality (4.12), we easily deduce that $(M_t(\boldsymbol{z}))_{t>0}$ is bounded in H, hence the first assertion of the statement and (4.15). We next prove (4.16). To do so, we average with respect to time with the operator M_t the balance equation (4.8) for $\mathrm{E}(\boldsymbol{z})$, which yields

$$\frac{1}{2t}\|\boldsymbol{z}(t)\|^2 - \frac{1}{2t}\|\boldsymbol{z}(0)\|^2 + \nu M_t(\|\nabla\boldsymbol{z}\|^2) \leq M_t(\mathrm{e}_m(\boldsymbol{u}) + \mathcal{E}_m(\boldsymbol{u})). \tag{4.17}$$

By using the energy inequality (4.12) once again, we see that the first two terms vanish as $t \to +\infty$ and also that $M_t(\|\nabla\boldsymbol{z}\|^2)$ is bounded. Therefore, (4.17) yields

$$0 \leq \nu \liminf_{n\to+\infty} M_{t_n}(\|\nabla\boldsymbol{z}\|^2) \leq \liminf_{n\to+\infty} M_{t_n}(\mathrm{e}_m(\boldsymbol{u}) + \mathcal{E}_m(\boldsymbol{u})),$$

hence (4.16). We observe that in this case we do not have any direct estimation on the behavior of the H^1-norm. □

In the case we can assume that the limit exists, we also have the following result.

Corollary 4.5. *Let us assume the limit of $M_{t_n}(\mathrm{e}_m(\boldsymbol{u}))$ for $n \to +\infty$ exists, and that*

$$\liminf_{T\to+\infty} \frac{\nu}{T}\int_0^T (\nabla\boldsymbol{z}(s), \nabla\boldsymbol{y}(s))\,ds = \liminf_{t\to\infty} \mathcal{E}_m(\boldsymbol{u}) \geq 0.$$

Then, it follows

$$\overline{\mathrm{e}_{m(\boldsymbol{u})}} = \lim_{n\to+\infty} \frac{1}{t_n}\int_0^{t_n} \mathrm{e}_m(\boldsymbol{u}(s))\,ds \geq 0.$$

This result can be interpreted as that, beyond the range of injection of energy, the average net transfer of energy occurs only into the small scales. This occurs if the term of interaction between gradients of large and small scales is negligible, in the limit of long time. This latter assumption is not proved rigorously, but we will see it is satisfied in the numerical tests, with a good degree of approximation (see Section 4.4). However, when one uses the eigenfunctions of the Stokes operator as POD basis, this is automatically satisfied since this basis is also orthogonal for the H^1-scalar product, so that in this case $\mathcal{E}_m(\boldsymbol{u}) = 0$.

4.3.2 The spectral case

The results of the previous section can be made much more precise in the case of decomposition made by a spectral basis of eigenfunctions of the Stokes operator. We present the results, which are in some sense new and not fully completely included in [12], in the sense of time-averaging. This procedure is applied to $\boldsymbol{u}$, which is a weak solution of the Navier-Stokes equations, satisfying the uniform estimates (4.12). In this way, the orthogonality (in both H and V) of the basis implies that

$$\|\boldsymbol{u}\|^2 = \|\boldsymbol{y}\|^2 + \|\boldsymbol{z}\|^2 \qquad \text{and} \qquad \|\nabla\boldsymbol{u}\|^2 = \|\nabla\boldsymbol{y}\|^2 + \|\nabla\boldsymbol{z}\|^2.$$

The results in this case are more precise than those from Theorem 4.4, since we have at disposal a larger set of a priori estimates and also the set of equations (4.9) has a better structure than (4.4).

We now prove the following results in the case of a decomposition of the velocity into small and large frequencies. The first one aims at taking the time average and then let t go to infinity in the equations (4.4) satisfied by $\boldsymbol{y}$ and $\boldsymbol{z}$. The second one aims at comparing the amount of turbulent dissipation of small and large frequencies with respect to the total work of the corresponding Reynolds stresses $\boldsymbol{\sigma}_{\boldsymbol{y}}^{(\mathrm{R})}$ and $\boldsymbol{\sigma}_{\boldsymbol{z}}^{(\mathrm{R})}$.

Theorem 4.6. *Let $\boldsymbol{u}(0) \in H$, $\boldsymbol{f} \in \mathbf{P}_m H$, and let $\boldsymbol{u}$ be a global-in-time weak solution to the NSE (4.1). Then, there exist*

1. *a sequence $\{t_n\}_{n\in\mathbb{N}}$ such that $\lim_{n\to\infty} t_n = +\infty$;*
2. *vector fields $\overline{\boldsymbol{y}}, \overline{\boldsymbol{z}} \in V$;*
3. *vector fields $\boldsymbol{B}_1, \boldsymbol{B}_2 \in V'$;*

such that it holds:

i) When $n \to \infty$,

$$M_{t_n}(\boldsymbol{y}) \rightharpoonup \overline{\boldsymbol{y}} \quad \text{weakly in } V, \qquad M_{t_n}(\boldsymbol{z}) \rightharpoonup \overline{\boldsymbol{z}} \quad \text{weakly in } V,$$
$$M_{t_n}\big((\boldsymbol{y}\cdot\nabla)\,\boldsymbol{y}\big) \rightharpoonup \boldsymbol{B}_1 \quad \text{weakly in } V', \qquad M_{t_n}\big((\boldsymbol{z}\cdot\nabla)\,\boldsymbol{z}\big) \rightharpoonup \boldsymbol{B}_2 \quad \text{weakly in } V',$$

ii) The Reynolds averaged equations:

$$\begin{cases} -\nu\Delta\overline{\boldsymbol{y}} + \nabla\overline{p_{\boldsymbol{y}}} + \boldsymbol{B}_1 = \mathbf{P}_m\boldsymbol{f} & \text{in } \Omega, \\ \nabla\cdot\overline{\boldsymbol{y}} = 0 & \text{in } \Omega, \\ \overline{\boldsymbol{y}} = \mathbf{0} & \text{on } \Gamma, \end{cases} \tag{4.18}$$

and

$$\begin{cases} -\nu\Delta\overline{\boldsymbol{z}} + \nabla\overline{p_{\boldsymbol{z}}} + \boldsymbol{B}_2 = \mathbf{0} & \text{in } \Omega, \\ \nabla\cdot\overline{\boldsymbol{z}} = 0 & \text{in } \Omega, \\ \overline{\boldsymbol{z}} = \mathbf{0} & \text{on } \Gamma, \end{cases} \tag{4.19}$$

holds true in V'.

Arguing as in [3, 28], using the relations $(\overline{\boldsymbol{z}}\cdot\nabla)\,\overline{\boldsymbol{z}} = \nabla\cdot(\overline{\boldsymbol{z}}\otimes\overline{\boldsymbol{z}})$ and $(\overline{\boldsymbol{y}}\cdot\nabla)\,\overline{\boldsymbol{y}} = \nabla\cdot(\overline{\boldsymbol{y}}\otimes\overline{\boldsymbol{y}})$, we get the existence of "small frequencies" and "large frequencies" Reynolds stresses $\boldsymbol{\sigma}_{\boldsymbol{y}}^{(\mathrm{R})}$ and $\boldsymbol{\sigma}_{\boldsymbol{z}}^{(\mathrm{R})}$ in V', such that

$$B_1 = \nabla\cdot\boldsymbol{\sigma}_{\boldsymbol{y}}^{(\mathrm{R})} + (\overline{\boldsymbol{y}}\cdot\nabla)\,\overline{\boldsymbol{y}} \qquad \text{and} \qquad B_2 = \nabla\cdot\boldsymbol{\sigma}_{\boldsymbol{z}}^{(\mathrm{R})} + (\overline{\boldsymbol{z}}\cdot\nabla)\,\overline{\boldsymbol{z}},$$

or equivalently, if we write the Reynolds decomposition as

$$\boldsymbol{y} = \overline{\boldsymbol{y}} + \boldsymbol{y}' \quad \text{and} \quad \boldsymbol{z} = \overline{\boldsymbol{z}} + \boldsymbol{z}',$$

then

$$\boldsymbol{\sigma}_{\boldsymbol{y}}^{(\mathrm{R})} = \overline{\boldsymbol{y}'\otimes\boldsymbol{y}'} \qquad \text{and} \qquad \boldsymbol{\sigma}_{\boldsymbol{z}}^{(\mathrm{R})} = \overline{\boldsymbol{z}'\otimes\boldsymbol{z}'},$$

where the bar operator denotes the limit of the M_{t_n}'s in V' as $n\to\infty$ (eventually after having extracted another sub-sequence).

According to the budget (4.14), we aim to compare the turbulent dissipation of small and large scales, denoted as $\varepsilon^{\downarrow}$ and $\varepsilon^{\uparrow}$ respectively, to the total work of the Reynolds stresses, namely $(\nabla\cdot\boldsymbol{\sigma}_{\boldsymbol{y}}^{(\mathrm{R})}, \overline{\boldsymbol{y}})$ and $(\nabla\cdot\boldsymbol{\sigma}_{\boldsymbol{z}}^{(\mathrm{R})}, \overline{\boldsymbol{z}})$, where

$$\epsilon^{\downarrow} := \nu\,\overline{\|\nabla\boldsymbol{y}'\|^2} \quad \text{and} \quad \epsilon^{\uparrow} := \nu\,\overline{\|\nabla\boldsymbol{z}'\|^2}.$$

When we compare to (4.14), we observe that triad nonlinear effect between small and large frequencies will be felt, that means the nonlinear interactions due to the convection will be provided by the term

$$\Phi_{\boldsymbol{z}}(\boldsymbol{y}) := (\mathbf{Q}_m[(\boldsymbol{y}+\boldsymbol{z})\cdot\nabla(\boldsymbol{y}+\boldsymbol{z})], \boldsymbol{z}) = -(\mathbf{P}_m[(\boldsymbol{y}+\boldsymbol{z})\cdot\nabla(\boldsymbol{y}+\boldsymbol{z})], \boldsymbol{y}) = -\Phi_{\boldsymbol{y}}(\boldsymbol{z}). \tag{4.20}$$

Notice that due to the regularity of $\boldsymbol{y}$, it is easily checked that the following energy balance holds true (this property will be shown with more details in the proof of Theorem 4.6 below)

$$\nu\,\|\nabla\overline{\boldsymbol{y}}\|^2 + (\nabla\cdot\boldsymbol{\sigma}_{\boldsymbol{y}}^{(\mathrm{R})}, \overline{\boldsymbol{y}}) = <\boldsymbol{f}, \overline{\boldsymbol{y}}>.$$

Finally, to prove Theorem 4.7 we will use the following orthogonality relation (see e.g., [3, Lemma 4.4]), formally written as follows

$$\overline{\|\nabla\psi\|^2} = \|\nabla\overline{\psi}\|^2 + \overline{\|\nabla\psi'\|^2}. \tag{4.21}$$

which is valid for any field $\psi : \mathbb{R}^+ \to V$, such that $\overline{\psi}$ is well-defined and the fluctuations are defined as $\psi' = \psi - \overline{\psi}$.

Theorem 4.7. *The families* $(M_t(\Phi_{\boldsymbol{z}}(\boldsymbol{y})))_{t>0}$ *and* $(M_t(\Phi_{\boldsymbol{y}}(\boldsymbol{z})))_{t>0}$ *converge (along certain subsequences) as* $t \to \infty$. *Let* $\overline{\Phi_{\boldsymbol{z}}(\boldsymbol{y})}$ *and* $\overline{\Phi_{\boldsymbol{y}}(\boldsymbol{z})}$ *denote the corresponding limits. One has*

$$\overline{\Phi_{\boldsymbol{z}}(\boldsymbol{y})} = -\overline{\Phi_{\boldsymbol{y}}(\boldsymbol{z})} \leq 0, \tag{4.22}$$

and the following dissipation balances hold true

$$\epsilon^{\downarrow} + \overline{\Phi_{\boldsymbol{y}}(\boldsymbol{z})} = (\nabla \cdot \boldsymbol{\sigma}_{\boldsymbol{y}}^{(\mathrm{R})}, \overline{\boldsymbol{y}}), \tag{4.23}$$

$$\epsilon^{\uparrow} + \overline{\Phi_{\boldsymbol{z}}(\boldsymbol{y})} \leq (\nabla \cdot \boldsymbol{\sigma}_{\boldsymbol{z}}^{(\mathrm{R})}, \overline{\boldsymbol{z}}). \tag{4.24}$$

Remark 4.8. Notice that by equations (4.22) and (4.23) we see that $\boldsymbol{\sigma}_{\boldsymbol{y}}^{(\mathrm{R})}$ is dissipative in mean, and follows the same law (4.14) as the complete Reynolds stress, namely

$$\epsilon^{\downarrow} \leq (\nabla \cdot \boldsymbol{\sigma}_{\boldsymbol{y}}^{(\mathrm{R})}, \overline{\boldsymbol{y}}).$$

However, nothing similar can be concluded from (4.23) about $\boldsymbol{\sigma}_{\boldsymbol{z}}^{(\mathrm{R})}$, that might be at this stage non dissipative in mean, which permits an inverse energy cascade to occur.

The results of Theorems 4.6 and 4.7 are original, even if similar results have been already obtained in [11] and reported also in [12]. In that case, the results are based on the notion of deterministic statistical solutions and on a sort of ergodic hypothesis. Even if statements could look very similar to ours, the main difference is that we do not average over the set H of initial data, and we do not introduce probability measures on H, as suggested by the work by Prodi [33, 34]. Our approach is based on a more elementary functional setting and also amenable to include treatment of sets of external forces, as those in several numerical or practical experiments. The main point is an extension of the results in [3].

Proof of Theorem 4.6. We know, from the results in [3, 28] that $\mathbf{U}_t = M_t(\boldsymbol{u})$ is such that

$$\begin{aligned} &\mathbf{U}_t \rightharpoonup \overline{\boldsymbol{u}} \qquad \text{weakly in } V, \\ &M_t((\boldsymbol{u}\cdot\nabla)\,\boldsymbol{u}) \rightharpoonup \mathbf{B} \qquad \text{in } L^{3/2}(\Omega) \subset V', \end{aligned}$$

hence, if we define $\mathbf{F} := \mathbf{B} - (\overline{\boldsymbol{u}}\cdot\nabla)\,\overline{\boldsymbol{u}}$, we get

$$\nu\,(\nabla\overline{\boldsymbol{u}}, \nabla\boldsymbol{\phi}) + \big((\overline{\boldsymbol{u}}\cdot\nabla)\,\overline{\boldsymbol{u}}, \boldsymbol{\phi}\big) + < \mathbf{F}, \boldsymbol{\phi} > = < \boldsymbol{f}, \boldsymbol{\phi} >,$$

and using $\overline{\boldsymbol{u}} \in V$ as test function we obtain

$$\nu \|\nabla \overline{\boldsymbol{u}}\|^2 + < \mathbf{F}, \overline{\boldsymbol{u}} > = < \boldsymbol{f}, \overline{\boldsymbol{u}} > .$$

We assume now that $\mathbf{P}_m \boldsymbol{f} = \boldsymbol{f}$, and we consider the equations satisfied by $\boldsymbol{y} = \mathbf{P}_m \boldsymbol{u}$ and $\boldsymbol{z} = \mathbf{Q_m} \boldsymbol{u}$. In particular, the equation for $\boldsymbol{y}$ reads, as an abstract equation in $V_m = \mathbf{P}_m V$, as follows:

$$\frac{d\boldsymbol{y}}{dt} + \nu A \boldsymbol{y} + \mathbf{P}_m[(\boldsymbol{y} + \boldsymbol{z}) \cdot \nabla(\boldsymbol{y} + \boldsymbol{z})] = \mathbf{P}_m \boldsymbol{f}.$$

The uniform estimates on $\boldsymbol{u}$ from Theorem 4.3 combined with Theorem 4.2, about the properties of the projection operator $\mathbf{P}_m$ as a continuous operator over V', yield

$$\begin{aligned} &M_t(\boldsymbol{y}) = \mathbf{Y}_t \rightharpoonup \overline{\boldsymbol{y}} \qquad \text{weakly in } V, \\ &\mathbf{P}_m M_t((\boldsymbol{u} \cdot \nabla)\, \boldsymbol{u}) = \mathbf{P}_m M_t[(\boldsymbol{y} + \boldsymbol{z}) \cdot \nabla(\boldsymbol{y} + \boldsymbol{z})] \rightharpoonup \mathbf{P}_m \mathbf{B} = \mathbf{B}_1 \qquad \text{weakly in } V', \end{aligned}$$

in such a way that $\overline{\boldsymbol{y}}$ satisfies, for the spectral basis $\mathcal{W}_k$ introduced in Section 4.1.1,

$$\nu\, (\nabla \overline{\boldsymbol{y}}, \nabla \mathcal{W}_k) + < (\overline{\boldsymbol{y}} \cdot \nabla)\, \overline{\boldsymbol{y}}, \mathcal{W}_k > + < \mathbf{F}_{\boldsymbol{y}}, \mathcal{W}_k > = < \boldsymbol{f}, \mathcal{W}_k > \qquad \text{for all } 1 \le k \le m,$$

where $\mathbf{F}_{\boldsymbol{y}} := \mathbf{B}_1 - (\overline{\boldsymbol{y}} \cdot \nabla)\, \overline{\boldsymbol{y}}$, which leads to (4.18) by De Rham Theorem, if written in a strong formulation. Arguing as in [3] (which was already mentioned above), it is easily checked that there exists $\boldsymbol{\sigma}_{\boldsymbol{y}}^{(\mathrm{R})}$ such that $\mathbf{F}_{\boldsymbol{y}} = \nabla \cdot \boldsymbol{\sigma}_{\boldsymbol{y}}^{(\mathrm{R})}$. Hence, being $\overline{\boldsymbol{y}} \in V_m \subset V$ a legitimate test function, we get

$$\nu \|\nabla \overline{\boldsymbol{y}}\|^2 + < \mathbf{F}_{\boldsymbol{y}}, \overline{\boldsymbol{y}} > = \nu \|\nabla \overline{\boldsymbol{y}}\|^2 + (\nabla \cdot \boldsymbol{\sigma}_{\boldsymbol{y}}^{(\mathrm{R})}, \overline{\boldsymbol{y}}) = < \boldsymbol{f}, \overline{\boldsymbol{y}} > . \tag{4.25}$$

The other term $\boldsymbol{z}$ of the decomposition satisfies

$$\frac{d}{dt}\boldsymbol{z} + \nu A \boldsymbol{z} + \mathbf{Q}_m[(\boldsymbol{y} + \boldsymbol{z}) \cdot \nabla(\boldsymbol{y} + \boldsymbol{z})] = 0.$$

The uniform estimates on $\boldsymbol{u}$ and the boundedness of the linear operator $\mathbf{P}_m$ imply the following convergence (up to a sub-sequence), as already shown in Theorem 4.4

$$\begin{aligned} &M_t(\boldsymbol{z}) = \mathbf{Z}_t \rightharpoonup \overline{\boldsymbol{z}} \qquad \text{weakly in } V, \\ &\mathbf{Q}_m M_t((\boldsymbol{u} \cdot \nabla)\, \boldsymbol{u}) = \mathbf{Q}_m M_t[(\boldsymbol{y} + \boldsymbol{z}) \cdot \nabla(\boldsymbol{y} + \boldsymbol{z})] \rightharpoonup \mathbf{Q}_m \mathbf{B} = \mathbf{B}_2 \qquad \text{weakly in } V'. \end{aligned}$$

By using that $\mathbf{B} = \mathbf{P}_m \mathbf{B} + \mathbf{Q}_m\, \mathbf{B}$, we get

$$\nu\, (\nabla \overline{\boldsymbol{z}}, \nabla \mathcal{W}_j) + < (\overline{\boldsymbol{z}} \cdot \nabla)\, \overline{\boldsymbol{z}}, \mathcal{W}_j > + < \mathbf{F}_{\boldsymbol{z}}, \mathcal{W}_j > = 0 \qquad \text{for all } j \ge m+1, \tag{4.26}$$

for $\mathbf{F}_{\boldsymbol{z}} = \mathbf{B}_2 - (\overline{\boldsymbol{z}} \cdot \nabla)\, \overline{\boldsymbol{z}} = \nabla \cdot \boldsymbol{\sigma}_{\boldsymbol{z}}^{(\mathrm{R})}$. Hence, (4.19) follows again by De Rham Theorem. Notice that, $\overline{\boldsymbol{z}} \in V_m^{\perp} \subset V$ and in particular $\overline{\boldsymbol{z}} \in \mathrm{span}\{\mathcal{W}_j\}_{j \ge m+1}$, so by linearity it can be used as test function in (4.26). Next, since $\nabla \cdot \overline{\boldsymbol{z}} = 0$, it follows that $< (\overline{\boldsymbol{z}} \cdot \nabla)\, \overline{\boldsymbol{z}}, \overline{\boldsymbol{z}} > = 0$ and we have the following energy equality:

$$\nu \|\nabla \overline{\boldsymbol{z}}\|^2 + (\nabla \cdot \boldsymbol{\sigma}_{\boldsymbol{z}}^{(\mathrm{R})}, \overline{\boldsymbol{z}}) = 0, \tag{4.27}$$

which concludes the proof. □

Proof of Theorem 4.7. We now write the energy inequality for $\boldsymbol{z}$, obtaining

$$\frac{1}{2}\frac{d}{dt}\|\boldsymbol{z}\|^2 + \nu\,\|\nabla\boldsymbol{z}\|^2 + (\mathbf{Q}_m[(\boldsymbol{y}+\boldsymbol{z})\cdot\nabla(\boldsymbol{y}+\boldsymbol{z})],\boldsymbol{z}) \le 0,$$

and hence, by using the orthogonality of the basis, we have that $\mathbf{Q}_m\boldsymbol{z} = \boldsymbol{z}$ and

$$\frac{1}{2}\frac{d}{dt}\|\boldsymbol{z}\|^2 + \nu\,\|\nabla\boldsymbol{z}\|^2 + \big([(\boldsymbol{y}+\boldsymbol{z})\cdot\nabla(\boldsymbol{y}+\boldsymbol{z})],\boldsymbol{z}\big) = \frac{1}{2}\frac{d}{dt}\|\boldsymbol{z}\|^2 + \|\nabla\boldsymbol{z}\|^2 + \Phi_{\boldsymbol{z}}(\boldsymbol{y}) \le 0,$$

recalling the definition of $\Phi_{\boldsymbol{z}}(\boldsymbol{y})$ in (4.20).

Averaging the above equation over a fixed time interval $(0,t)$, we get

$$\frac{1}{2t}\|\boldsymbol{z}(t)\|^2 - \frac{1}{2t}\|\boldsymbol{z}(0)\|^2 + \nu M_t(\|\nabla\boldsymbol{z}\|^2) + M_t(\Phi_{\boldsymbol{z}}(\boldsymbol{y})) \le 0.$$

The L^2-uniform bounds on $\boldsymbol{z}$ imply that $\frac{1}{2t}\|\boldsymbol{z}(t)\|^2 \to 0$, hence, possibly after having extracted another sub-sequence to ensure the convergence of the term $M_t(\|\nabla\boldsymbol{z}\|^2)$ (that is known to be bounded by the energy inequality (4.12)) we get

$$\limsup_{n\to\infty} M_{t_n}(\Phi_{\boldsymbol{z}}(\boldsymbol{y})) \le -\nu\,\overline{\|\nabla\boldsymbol{z}\|^2} \le 0. \tag{4.28}$$

We now combine the orthogonality relation (4.21) with the energy balance (4.27), so that (4.28) yields

$$\epsilon^{\uparrow} + \limsup_{n\to\infty} M_{t_n}(\Phi_{\boldsymbol{z}}(\boldsymbol{y})) \le (\nabla\cdot\boldsymbol{\sigma}_{\boldsymbol{z}}^{(\mathrm{R})},\overline{\boldsymbol{z}}),$$

which is almost inequality (4.24), up to the convergence of $(M_t(\Phi_{\boldsymbol{z}}(\boldsymbol{y})))_{t>0}$ that remains to be proved. To prove this, we deal with the budget for $\boldsymbol{y}$, recall the definition of $\Phi_{\boldsymbol{z}}(\boldsymbol{y})$ and $\Phi_{\boldsymbol{y}}(\boldsymbol{z})$ from (4.20).

Then, averaging the energy equality (in this case we have equality since $\boldsymbol{y}$ solves a finite dimensional system of ordinary differential equations) which is satisfied for $\boldsymbol{y}$, we get

$$\frac{1}{2t}\|\boldsymbol{y}(t)\|^2 - \frac{1}{2t}\|\boldsymbol{y}(0)\|^2 + \nu M_t(\|\nabla\boldsymbol{y}\|^2) + M_t(\Phi_{\boldsymbol{y}}(\boldsymbol{z})) = <\boldsymbol{f}, M_t(\boldsymbol{y})>. \tag{4.29}$$

By the same argument, eventually after having extracted a further sub-sequence, as $t_n\to\infty$ $(M_t(\|\nabla\boldsymbol{y}\|^2))_{t>0}$ is convergent, as well as $(<\boldsymbol{f}, M_t(\boldsymbol{y})>)_{t>0}$. Therefore, $\{M_{t_n}(\Phi_{\boldsymbol{y}}(\boldsymbol{z}))\}$ is also convergent by (4.29). Let $\overline{\Phi_{\boldsymbol{y}}(\boldsymbol{z})}$ denote its limit. In particular, by (4.20), $\{M_{t_n}(\Phi_{\boldsymbol{z}}(\boldsymbol{y}))\}$ is also convergent, with limit $\overline{\Phi_{\boldsymbol{z}}(\boldsymbol{y})} = -\overline{\Phi_{\boldsymbol{y}}(\boldsymbol{z})}$. We are done with (4.24).

It remains to check (4.23). Taking the limit as $n\to\infty$ in (4.29) gives the equality

$$\nu\,\overline{\|\nabla\boldsymbol{y}\|^2} + \overline{\Phi_{\boldsymbol{z}}(\boldsymbol{y})} = <\boldsymbol{f},\overline{\boldsymbol{y}}>,$$

which, combined with the energy balance (4.25) and the orthogonality relation (4.21), yields (4.23), ending the proof. □

4.4 Numerical results

In Theorem 4.4, we showed that

$$\liminf_{T\to+\infty} \frac{1}{T}\int_0^T \mathrm{e}_m(\boldsymbol{u}(s)) + \mathcal{E}_m(\boldsymbol{u}(s))\, ds \geq 0. \tag{4.30}$$

In this section, we investigate numerically whether the inequality (4.30) holds. To this end, we consider the one-dimensional Burgers equation with homogeneous Dirichlet boundary conditions as a simplified, yet relevant test case [16, 17, 20, 21, 22, 23, 24, 31, 40, 43]:

$$\begin{cases} u_t - \nu u_{xx} + u u_x = f & (x,t) \in \Omega \times [0,1], \\ u = 0 & (x,t) \in \partial\Omega \times [0,1]. \end{cases} \tag{4.31}$$

To calculate the long-time average of $\mathrm{e}_m(u)$ in (4.30), we use the composite trapezoidal rule:

$$\frac{1}{T}\int_0^T \mathrm{e}_m(u(s))\, ds \approx \frac{1}{2n}\sum_{i=1}^{n} \big(\mathrm{e}_m(u(t_i)) + \mathrm{e}_m(u(t_{i+1}))\big), \tag{4.32}$$

where $t_i = (i-1)*\frac{T}{n}$, $i = 1, \ldots, n+1$. We also use the composite trapezoidal rule to calculate the long-time average of $\mathcal{E}_m(u)$.

4.4.1 Numerical results with step function initial condition

Our numerical results are obtained by using the one-dimensional Burgers equation (4.31) with a step function initial condition [16, 43]:

$$u_0(x) = \begin{cases} 1, & x \in (0, 1/2], \\ 0, & x \in (1/2, 1]. \end{cases} \tag{4.33}$$

We use the following parameters in the finite element discretization of the Burgers equation (4.31): $\Omega = [0,1]$, $\nu = 10^{-2}$, $f = 0$, mesh size $h = 1/128$, piecewise linear finite element spatial discretization, and backward Euler time discretization.

Case 1:

For this test case, we consider the time interval $[0,T] = [0,1]$ and the time step $\Delta t = 10^{-2}$. We utilize all the snapshots to build the POD basis, whose dimension is $d = 37$. In the composite trapezoidal rule, we use $n = 100$. In Figure 4.1, we plot the DNS results (which are used to generate the snapshots). In Table 4.1, we list the time-averages of $\mathrm{e}_m(u)$ and

$\mathcal{E}_m(u)$ for different m values. We note that the time-average of $\mathrm{e}_m(u)$ is positive for all m values. The time-average of $\mathcal{E}_m(u)$ is positive for the low m values and negative for the largest m values. Furthermore, the magnitude of the time-average of $\mathcal{E}_m(u)$ is generally lower than the magnitude of the time-average of $\mathrm{e}_m(u)$. Thus, we conclude that the time average $\frac{1}{T}\int_0^T \mathrm{e}_m(u(s)) + \mathcal{E}_m(u(s))\,ds$ in (4.30) is positive for all m values.

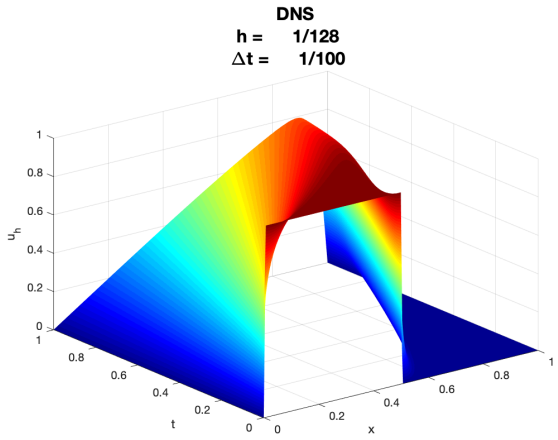

Figure 4.1: DNS solution obtained by using a piecewise linear finite element spatial discretization and the backward Euler time discretization.

m	$\int_0^1 \mathrm{e}_m(u(s))\,ds$	$\int_0^1 \mathcal{E}_m(u(s))\,ds$
3	2.6170e-02	1.0451e-03
6	7.3208e-03	2.2524e-03
9	1.4934e-03	1.5977e-03
15	3.6181e-05	3.6287e-05
20	1.2776e-06	7.0356e-07
25	4.6207e-08	9.5638e-09
30	2.0257e-09	-1.1127e-10
35	8.2806e-11	-2.2595e-11

Table 4.1: Case 1: Time-averages of $\mathrm{e}_m(u)$ and $\mathcal{E}_m(u)$ for $d = 37$ and different m values.

In Case 1, we showed that the time average $\frac{1}{T}\int_0^T \mathrm{e}_m(u(s)) + \mathcal{E}_m(u(s))\,ds$ in (4.30) is positive. In the remainder of this section, we investigate whether this time average remains positive if we make the following changes in our computational setting: (i) we increase/decrease the time-interval; and (ii) we use more quadrature points (i.e., subintervals) in the composite trapezoidal rule (4.32).

Case 2:

In this case, we use a longer time interval, i.e., $[0,T] = [0,10]$ (instead of $[0,T] = [0,1]$, as we used in Case 1). We also use different time step (Δt) values to generate the snapshots and a different number of quadrature points to evaluate the time average $\frac{1}{T}\int_0^T \mathrm{e}_m(u(s)) + \mathcal{E}_m(u(s))\,ds$.

In Tables 4.2–4.5, we list the time-averages of $\mathrm{e}_m(u)$ and $\mathcal{E}_m(u)$ for different time steps (Δt) values, different number of equally spaced quadrature points (n), and different m values. We note that the time-averages of $\mathrm{e}_m(u)$ and $\mathcal{E}_m(u)$ are positive for all Δt, n, and m values. Moreover, the magnitude of the time-average of $\mathcal{E}_m(u)$ is generally lower than the magnitude of the time-average of $\mathrm{e}_m(u)$. Thus, we conclude that the time average $\frac{1}{T}\int_0^T \mathrm{e}_m(u(s)) + \mathcal{E}_m(u(s))\,ds$ in (4.30) is positive for all Δt, n, and m values. Furthermore, we note that decreasing the time step while keeping the same number of snapshots (i.e., $n = 10000$) does not change the time average $\frac{1}{T}\int_0^T \mathrm{e}_m(u(s)) + \mathcal{E}_m(u(s))\,ds$ significantly (see Tables 4.3–4.5).

m	$\frac{1}{10}\int_0^{10} \mathrm{e}_m(u(s))\,ds$	$\frac{1}{10}\int_0^{10} \mathcal{E}_m(u(s))\,ds$
3	3.3652e-03	8.6523e-05
6	1.0001e-03	2.1570e-04
9	2.1132e-04	1.8614e-04
15	5.4224e-06	5.5724e-06
20	4.3119e-07	2.5667e-07
25	5.6884e-08	1.0451e-08
30	1.1327e-09	3.2736e-10
35	2.0469e-11	2.6396e-12

Table 4.2: Case 2: Time-averages of $\mathrm{e}_m(u)$ and $\mathcal{E}_m(u)$ for $d = 38$, $\Delta t = 10^{-2}$, 1000 equally spaced quadrature points, and different m values.

m	$\frac{1}{10}\int_0^{10} \mathrm{e}_m(u(s))ds$	$\frac{1}{10}\int_0^{10} \mathcal{E}_m(u(s))\,ds$
3	3.5296e-03	3.0515e-06
6	1.1402e-03	8.8746e-06
9	2.6359e-04	1.5090e-05
15	1.1126e-05	1.1156e-05
20	9.5913e-07	1.5183e-06
25	1.8140e-07	1.8704e-07
30	4.8765e-08	1.3491e-08
35	1.0291e-09	9.7629e-10
40	2.4680e-11	2.0416e-11

Table 4.3: Case 2: Time-averages of $\mathrm{e}_m(u)$ and $\mathcal{E}_m(u)$ for $d = 41$, $\Delta t = 10^{-3}$, 10000 equally spaced quadrature points, and different m values.

m	$\frac{1}{10}\int_0^{10} \mathrm{e}_m(u(s))ds$	$\frac{1}{10}\int_0^{10} \mathcal{E}_m(u(s))\,ds$
3	3.5637e-03	2.7841e-06
6	1.1664e-03	8.1109e-06
9	2.7346e-04	1.3956e-05
15	1.2084e-05	1.1937e-05
20	1.1785e-06	1.6370e-06
25	2.2727e-07	1.2913e-07
30	6.2606e-08	9.5191e-09
35	2.1106e-09	6.1538e-10
40	1.0330e-10	2.0241e-11

Table 4.4: Case 2: Time-averages of $\mathrm{e}_m(u)$ and $\mathcal{E}_m(u)$ for $d = 43$, $\Delta t = 10^{-4}$, 10000 equally spaced quadrature points, and different m values.

m	$\frac{1}{10}\int_0^{10} \mathrm{e}_m(u(s))ds$	$\frac{1}{10}\int_0^{10} \mathcal{E}_m(u(s))\,ds$
3	3.5668e-03	2.7594e-06
6	1.1688e-03	8.0390e-06
9	2.7436e-04	1.3843e-05
15	1.2172e-05	1.2002e-05
20	1.2030e-06	1.6405e-06
25	2.3339e-07	1.2463e-07
30	6.4120e-08	8.8971e-09
35	2.2654e-09	5.6187e-10
40	1.1211e-10	1.8216e-11

Table 4.5: Case 2: Time-averages of $\mathrm{e}_m(u)$ and $\mathcal{E}_m(u)$ for $d = 43$, $\Delta t = 2 * 10^{-5}$, 10000 equally spaced quadrature points, and different m values.

Case 3:

In this case, we use an even longer time interval, i.e., $[0, T] = [0, 100]$, and compare the time-averages for this time interval to those for the time intervals $[0, T] = [0, 1]$ (Case 1) and $[0, T] = [0, 10]$ (Case 2). For each time interval, we use the same time step values ($\Delta t = 10^{-2}$) to generate the snapshots and all the subintervals in the composite trapezoidal rule utilized in the evaluation of the time average $\frac{1}{T}\int_0^T \mathrm{e}_m(u(s)) + \mathcal{E}_m(u(s))\,ds$. In Table 4.6, we list the time-averages of $\mathrm{e}_m(u)$ and $\mathcal{E}_m(u)$ for all three time intervals and different m values. We note that the time-averages of $\mathrm{e}_m(u)$ and $\mathcal{E}_m(u)$ are positive for all time intervals and m values. Furthermore, the magnitude of the time-average of $\mathcal{E}_m(u)$ is generally lower than the magnitude of the time-average of $\mathrm{e}_m(u)$. Thus, we conclude that the time-average $\frac{1}{T}\int_0^T \mathrm{e}_m(u(s)) + \mathcal{E}_m(u(s))\,ds$ in (4.30) is positive for all time intervals and m values. Furthermore, we note that the time-averages of $\mathrm{e}_m(u)$ and $\mathcal{E}_m(u)$ for the time intervals $[0, T] = [0, 100]$ and $[0, T] = [0, 10]$ are close, whereas those for the time interval $[0, T] = [0, 1]$ are slightly different. Thus, we conclude that the time interval $[0, T] = [0, 10]$ is adequate for the approximation of the long time-average $\frac{1}{T}\int_0^T \mathrm{e}_m(u(s)) + \mathcal{E}_m(u(s))\,ds$.

m	$\frac{1}{100}\int_0^{100} \mathrm{e}_m(u(s))\,ds$	$\frac{1}{10}\int_0^{10} \mathrm{e}_m(u(s))\,ds$	$\int_0^{1} \mathrm{e}_m(u(s))\,ds$
3	3.3687e-04	3.3683e-04	1.7634e-04
6	1.0001e-04	1.0001e-04	7.3142e-05
9	2.1146e-05	2.1147e-05	1.6701e-05
15	5.5621e-07	5.5742e-07	4.7129e-07
20	7.9605e-08	7.9605e-08	6.3799e-08
25	8.3483e-09	8.3489e-09	5.7426e-09
30	1.0839e-10	1.0840e-10	9.2430e-11
35	2.7710e-12	2.7718e-12	2.9575e-12

Table 4.6: Case 3: Time-averages of $\mathrm{e}_m(u)$ for $d = 36$, $\Delta t = 10^{-2}$, different m values, and all subintervals used in the composite trapezoidal rule.

m	$\frac{1}{100}\int_0^{100} \mathcal{E}_m(u(s))\,ds$	$\frac{1}{10}\int_0^{10} \mathcal{E}_m(u(s))\,ds$	$\int_0^{1} \mathcal{E}_m(u(s))\,ds$
3	8.6270e-06	7.0320e-06	8.1057e-05
6	2.1568e-05	2.1520e-05	3.2405e-05
9	1.8614e-05	1.8599e-05	2.0255e-05
15	5.5718e-07	5.4232e-07	5.8207e-07
20	5.6044e-08	5.5874e-08	6.1423e-08
25	1.0936e-09	9.9763e-10	5.9386e-10
30	3.3046e-11	3.2478e-11	4.0949e-11
35	6.4170e-13	6.3739e-13	1.2125e-12

Table 4.7: Case 3: Time-averages of $\mathcal{E}_m(u)$ for $d = 36$, $\Delta t = 10^{-2}$, different m values, and all subintervals used in the composite trapezoidal rule.

Case 4:

In this case, we use a much shorter time interval, i.e., $[0, T] = [0, 0.1]$, and compare the time-averages for this time interval to those for the time intervals $[0, T] = [0, 1]$, $[0, T] = [0, 10]$, and $[0, T] = [0, 100]$ (Case 3). We use two different time step values to generate the snapshots, but the same (i.e., $n = 5000$) equally spaced subintervals in the composite trapezoidal rule utilized in the evaluation of the time average $\frac{1}{T}\int_0^T \mathrm{e}_m(u(s)) + \mathcal{E}_m(u(s))\,ds$. In Tables 4.8–4.9, we list the time-averages of $\mathrm{e}_m(u)$ and $\mathcal{E}_m(u)$ for two different time step values and different m values. We emphasize that, this time, the time-average of $\mathrm{e}_m(u)$ is negative for some m values. Furthermore, the magnitude of the time-average of $\mathcal{E}_m(u)$ is larger than the magnitude of the time-average of $\mathrm{e}_m(u)$. This is in stark contrast with the previous cases.

m	$\frac{1}{0.1}\int_0^{0.1} \mathrm{e}_m(u(s))ds$	$\frac{1}{0.1}\int_0^{0.1} \mathcal{E}_m(u(s))ds$
3	-6.8687e-04	2.7378e-05
5	-2.6333e-05	1.7795e-05
7	-9.1458e-07	4.2432e-06
9	-9.8188e-09	4.1220e-07
13	2.3800e-10	1.5749e-09
15	8.5423e-12	4.8043e-11

Table 4.8: Case 4: Time-averages of $\mathrm{e}_m(u)$ and $\mathcal{E}_m(u)$ for $\Delta t = 2 * 10^{-5}$, different m values, $d = 18$, and 5000 equally spaced subintervals used in the composite trapezoidal rule.

m	$\frac{1}{0.1}\int_0^{0.1} \mathrm{e}_m(u(s))ds$	$\frac{1}{0.1}\int_0^{0.1} \mathcal{E}_m(u(s))ds$
3	-6.8807e-04	2.7338e-05
5	-2.6447e-05	1.7812e-05
7	-9.1691e-07	4.2755e-06
9	-9.5609e-09	4.1694e-07
13	2.5287e-10	1.5941e-09
15	1.0725e-11	4.7030e-11

Table 4.9: Case 4: Time-averages of $\mathrm{e}_m(u)$ and $\mathcal{E}_m(u)$ for $\Delta t = 10^{-5}$, different m values, $d = 18$, and 5000 equally spaced subintervals used in the composite trapezoidal rule.

4.5 Conclusions

In this preliminary study, we investigated theoretically and numerically the time-average of the exchange of energy among modes of reduced order models (ROMs) of fluid flows. In particular, we were interested in the statistical equilibrium problem, and especially in the long-time averaging of the ROM solutions. The main goal of the paper was to deduce the possible forward and backward average transfer of the energy among ROM basis functions (modes). We considered two types of ROM modes: Eigenfunctions of the Stokes operator and proper orthogonal decomposition (POD) modes. In Theorem 4.4 and Theorem 4.6, we proved analytical results for both types of ROM modes and we highlighted the differences between them, especially those stemming from the lack of orthogonality of the gradients of the POD basis functions.

In Section 4.4, we performed a preliminary numerical investigation aiming at determining whether the time-average energy exchange between POD modes (i.e., $\frac{1}{T}\int_0^T \mathrm{e}_m(u(s)) +$

$\mathcal{E}_m(u(s))\,ds)$ in Theorem 4.4 is positive. To this end, we used the one-dimensional Burgers equation as a mathematical model. We utilized a piecewise linear FE spatial discretization and a backward Euler temporal discretization. To compute the time-averages, we used the composite trapezoidal rule. We tested different time steps, different number of subintervals in the composite trapezoidal rule, and, most importantly, different time intervals, to ensure that the computed quantities are indeed approximations of the time-averages and not numerical artifacts. The main conclusion of our numerical study is that, for long enough time intervals (i.e., time intervals longer than $[0, T] = [0, 10]$), the time-average $\frac{1}{T}\int_0^T \mathrm{e}_m(u(s)) + \mathcal{E}_m(u(s))\,ds$ in (4.30) is positive. Furthermore, the magnitude of the time-average of $\mathcal{E}_m(u)$ is much lower than the magnitude of the time-average of $\mathrm{e}_m(u)$.

There are several research directions that we plan to pursue. Probably the most important one is the numerical investigation of the theoretical results in three-dimensional, high Reynolds number flows, which could shed new light on the energy transfer among ROM modes. A related, but different numerical investigation was performed in [7].

Bibliography

[1] Batchelor GK (1953) The theory of homogeneous turbulence. *Cambridge Monographs on Mechanics and Applied Mathematics* Cambridge University Press.

[2] Berselli LC, Iliescu T, Layton WJ (2006) *Mathematics of Large Eddy Simulation of Turbulent Flows*, Berlin: Springer-Verlag.

[3] Berselli LC, Lewandowski R (2019) On the Reynolds time-averaged equations and the long-time behavior of Leray-Hopf weak solutions, with applications to ensemble averages. arXiv preprint arXiv:1801.08721.

[4] Berselli LC, Fagioli S, Spirito S (2019) Suitable weak solutions of the Navier-Stokes equations constructed by a space-time numerical discretization. *J Math Pures Appl* 125: 189–208.

[5] Rebollo TC, Lewandowski R (2014) *Mathematical and Numerical Foundations of Turbulence Models and Applications*, New York: Springer.

[6] Constantin P, Foias C (1988) *Navier-Stokes Equations*, Chicago: University of Chicago Press.

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

[8] DeCaria V, Layton WJ, McLaughlin M (2017) A conservative, second order, unconditionally stable artificial compression method. *Comput Methods Appl Mech Engrg* 325: 733–747.

[9] DeCaria V, Iliescu T, Layton W, et al. (2019) An artificial compression reduced order model. arXiv preprint arXiv:1902.09061.

[10] Girault V, Raviart PA (1986) *Finite Element Methods for Navier-Stokes Equations*, Berlin: Springer-Verlag.

[11] Foiaş C (1972/73) Statistical study of Navier-Stokes equations. I, II. *Rend Sem Mat Univ Padova* 48: 219–348; ibid. 49: 9–123.

[12] Foias C, Manley O, Rosa R, et al. (2001) *Navier-Stokes Equations and Turbulence*, Cambridge: Cambridge University Press.

[13] Guermond JL, Minev P, Shen J (2006) An overview of projection methods for incompressible flows. *Comput Methods Appl Mech Engrg* 195: 6011–6045.

[14] Guermond JL, Oden JT, Prudhomme S (2004) Mathematical perspectives on large eddy simulation models for turbulent flows. *J Math Fluid Mech* 6: 194–248.

[15] Hesthaven JS, Rozza G, Stamm B (2016) *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*, Berlin: Springer.

[16] Iliescu T, Wang Z (2014) Are the snapshot difference quotients needed in the proper orthogonal decomposition? *SIAM J Sci Comput* 36: A1221–A1250.

[17] Iliescu T, Liu H, Xie X (2018) Regularized reduced order models for a stochastic Burgers equation *Int J Numer Anal Mod* 15: 594–607.

[18] Jiang N, Layton WJ (2016) Algorithms and models for turbulence not at statistical equilibrium. *Comput Math Appl* 71: 2352–2372.

[19] Kolmogorov AN (1941) The local structure of turbulence in incompressible viscous fluids for very large Reynolds number. *Dokl Akad Nauk SSR* 30: 9–13.

[20] Kunisch K, Volkwein S (1999) Control of the Burgers equation by a reduced-order approach using proper orthogonal decomposition. *J Optim Theory Appl* 102: 345–371.

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

[22] Kunisch K, Volkwein S, Xie L (2004) HJB-POD-based feedback design for the optimal control of evolution problems. *SIAM J Appl Dyn Syst* 3: 701–722.

[23] Kunisch K, Xie L (2005) POD-based feedback control of the Burgers equation by solving the evolutionary HJB equation. *Comput Math Appl* 49: 1113–1126.

[24] Kunisch K, Volkwein S (2008) Proper orthogonal decomposition for optimality systems. *ESAIM: Math Model Numer Anal* 42: 1–23.

[25] Lassila T, Manzoni A, Quarteroni A, et al. (2014) Model order reduction in fluid dynamics: challenges and perspectives. In: *Reduced order methods for modeling and computational reduction*, Springer, 9: 235–273. .

[26] Layton WJ (2014) The 1877 Boussinesq conjecture: Turbulent fluctuations are dissipative on the mean flow. Technical Report TR-MATH 14-07, Pittsburgh Univ.

[27] Layton WJ, Rebholz L (2012) *Approximate Deconvolution Models of Turbulence Approximate Deconvolution Models of Turbulence*, Heidelberg: Springer.

[28] Lewandowski R (2015) Long-time turbulence model deduced from the Navier-Stokes equations. *Chin Ann Math Ser B* 36: 883–894.

[29] Lions JL, (1969) *Quelques Méthodes de Résolution des Problèmes aux Limites Non Linéaires*, Paris: Dunod.

[30] Málek J, Nečas J, Rokyta M, et al. (1996) *Weak and Measure-valued Solutions to Evolutionary PDEs*, London: Chapman & Hall.

[31] Park HM, Jang YD (2000) Control of Burgers equation by means of mode reduction. *Int J of Eng Sci* 38: 785–805.

[32] Prandtl L (1925) Bericht über Untersuchungen zur ausgebildeten Turbulenz. *Z Angew Math Mech* 5: 136–139.

[33] Prodi G (1960) Teoremi ergodici per le equazioni della idrodinamica. In:*Sistemi Dinamici e Teoremi Ergodici*, Berlin: Springer, 159-177.

[34] Prodi G (1961) On probability measures related to the Navier-Stokes equations in the 3-dimensional case. Technical Report AF61(052)-414, Trieste Univ.

[35] Quarteroni A, Manzoni A, Negri F (2016) *Reduced Basis Methods for Partial Differential Equations*, Berlin: Springer.

[36] Quarteroni A, Rozza G, Manzoni A (2011) Certified reduced basis approximation for parametrized partial differential equations and applications. *J Math Ind* 1: 3.

[37] Reynolds O (1895) On the dynamic theory of the incompressible viscous fluids and the determination of the criterion. *Philos Trans Roy Soc London Ser A* 186: 123–164.

[38] Rozza G (2014) Fundamentals of reduced basis method for problems governed by parametrized PDEs and applications, In: *Separated Representations and PGD-based Model Reduction*, Vienna: Springer, 153–227.

[39] Sagaut P (2001) *Large Eddy Simulation for Incompressible Flows*. Berlin: Springer-Verlag.

[40] San O, Maulik R (2018) Neural network closures for nonlinear model order reduction. *Adv Comput Math* 44: 1717–1750.

[41] Wells D, Wang Z, Xie X, et al. (2017) An evolve-then-filter regularized reduced order model for convection-dominated flows. *Internat J. Numer Methods Fluids* 84: 598–615.

[42] Xie X, Wells D, Wang Z, et al. (2017) Approximate deconvolution reduced order modeling. *Comput Methods Appl Mech Engrg* 313: 512–534.

[43] Xie X, Mohebujjaman M, Rebholz LG, et al. (2018) Data-driven filtered reduced order modeling of fluid flows. *SIAM J Sci Comput* 40: B834–B857.

[44] Xie X, Mohebujjaman M, Rebholz LG, et al. (2018) Lagrangian data-driven reduced order modeling of finite time Lyapunov exponents. arXiv preprint arXiv:1808.05635.

Chapter 5

Verifiability of the Data-Driven Variational Multiscale Reduced Order Model

This is joint work with Changhong Mou, Honghu Liu, Zhu Wang, Gianluigi Rozza, and Traian Iliescu. My contribution in this work was being part of the mathematical analysis of the model and presenting numerical experiments for the Burgers equation in Section 5.7.3.

5.1 Abstract

In this paper, we focus on the mathematical foundations of reduced order model (ROM) closures. First, we extend the verifiability concept from large eddy simulation to the ROM setting. Specifically, we call a ROM closure model verifiable if a small ROM closure model error (i.e., a small difference between the true ROM closure and the modeled ROM closure) implies a small ROM error. Second, we prove that a data-driven ROM closure (i.e., the data-driven variational multiscale ROM) is verifiable. Finally, we investigate the verifiability of the data-driven variational multiscale ROM in the numerical simulation of the Burgers equation and a two-dimensional flow past a circular cylinder at Reynolds numbers $Re = 100$ and $Re = 1000$.

5.2 Introduction

Full order models (FOMs) are computational models obtained with classical numerical methods (e.g., finite element or finite difference methods). In the numerical simulation of fluid flows, FOMs often yield high-dimensional (e.g., $\mathcal{O}(10^6)$) systems of equations. Thus, the computational cost of using FOMs in important many-query fluid flow applications (e.g., uncertainty quantification, optimal control, and shape optimization) can be prohibitively high.

Reduced order models (ROMs) are computational models that yield systems of equations whose dimensions are dramatically lower than those corresponding to FOMs. For example, in the numerical simulation of fluid flows that are dominated by recurrent spatial structures (e.g., flow past bluff bodies), the dimensions of the resulting system of equations can be $\mathcal{O}(10)$ for ROMs and $\mathcal{O}(10^6)$ for FOMs, while the ROM and FOM accuracy is of the same order. Thus, ROMs have been used in many-query fluid flow applications to reduce the computational cost of FOMs. Probably the most popular type of ROM used in these applications is the Galerkin ROM (G-ROM), which is constructed by using the Galerkin method. The G-ROM is based on a simple yet powerful idea: Instead of using millions or even billions of general purpose basis functions (as in classical Galerkin methods, such as the tent functions in the finite element method), G-ROM uses a lower-dimensional data-driven basis. Specifically, the available numerical or experimental data is used to build a few ROM basis functions that model the spatial structures that dominate the flow dynamics.

The G-ROM has been successful in the efficient numerical simulation of relatively simple laminar flows, e.g., flow past a circular cylinder at low Reynolds numbers. However, the standard G-ROM generally fails in the numerical simulation of turbulent flows. The main reason is that, in order to ensure a relatively low computational cost, only a few ROM basis functions are used to build the standard G-ROM. These few ROM basis functions can represent the simple dynamics of laminar flows, but not the complex dynamics of turbulent

flows. Thus, in the numerical simulation of turbulent flows, the standard G-ROM is equipped with a ROM closure model, i.e., a correction term that models the effect of the discarded ROM basis functions on the ROM dynamics.

Over the last two decades, ROM closure modeling has witnessed a dynamic development. Three main types of ROM closure models have been proposed: (i) Functional ROM closures are constructed by using physical insight. Classical examples of functional ROM closures include eddy viscosity models [35], in which the main role of the ROM closure model is to dissipate energy, as predicted by Kolmogorov's statistical theory of turbulence and confirmed in a ROM setting both numerically [7] and theoretically [3]. (ii) Structural ROM closures are a different class of models that are developed by using mathematical arguments. Examples of structural ROM closures include the approximate deconvolution ROM [38], the Mori-Zwanzig formalism [20, 26], and the parameterized manifolds [6]. (iii) The most active research area in ROM closure modeling is in the development of data-driven ROM closures in which available data is utilized to build the ROM closure model. An example of data-driven ROM closure is the data-driven variational multiscale ROM (DD-VMS-ROM) that was proposed in [23, 36]. The DD-VMS-ROM has been investigated numerically in [17, 21, 23, 24, 36, 37]. However, providing mathematical support for the DD-VMS-ROM is an open problem.

In classical CFD, there exists extensive mathematical support for closure modeling. For example, the monographs [4, 13, 30] present the mathematical analysis for many large eddy simulation (LES) models, as well as the numerical analysis of their discretization. In contrast, despite the recent increased interest in ROM closure modeling, the mathematical foundations of ROM closures is relatively scarce. Indeed, the ROM closure models are generally assessed heuristically: The proposed ROM closure model is used in numerical simulations and is shown to improve the numerical accuracy of the standard G-ROM and/or other ROM closure models. However, fundamental questions in ROM numerical analysis are still wide open for most of these ROM closure models: Is the proposed ROM closure model stable? Does the ROM closure model converge? If so, what does it converge to?

Only the first steps in the numerical analysis of ROM closures have been taken. To our knowledge, the first numerical analysis of a ROM closure model was performed in [5], where an eddy viscosity ROM closure model (i.e., the Smagorinsky model) was analyzed in a simplified setting. Next, the numerical analysis of eddy viscosity variational multiscale ROMs was carried out in [10, 12]. Finally, the numerical analysis of the Samagorinsky model in a reduced basis method setting was performed in [2, 29]. (We also note that numerical analysis for regularized ROMs, which are related to but different from ROM closures, was performed in [8, 39].)

In this paper, we take a next step in the development of numerical analysis for ROM closures and prove verifiability for a data-driven ROM closure model, i.e., the DD-VMS-ROM proposed in [23, 36]. Specifically, we show that the ROM closure model in the DD-VMS-ROM is accurate in a precise sense. More importantly, we prove that the DD-VMS-ROM is verifiable, i.e., we prove that since the DD-VMS-ROM closure model is accurate, the DD-VMS-ROM

solution is accurate. We note that this is not a trivial task: The Navier-Stokes equations (and their filtered counterparts), which are the mathematical models that use in this paper, are nonlinear and sensitive to perturbations, so adding to them a relatively small term (i.e., the ROM closure term) does not automatically imply that the resulting solution will be close to the original one. To prove that the DD-VMS-ROM closure model is verifiable, we use the following ingredients: (i) We use ROM spatial filtering to determine an explicit formula for the exact ROM closure term, which needs to be modeled. (ii) We use data-driven modeling to construct the DD-VMS-ROM closure model and show that this closure model is accurate, i.e., it is close to the exact ROM closure model. (iii) We use physical constraints to increase the accuracy of our data-driven ROM closure model. We note that the verifiability concept was defined in an LES context (see, e.g., [16] as well as [4] for a survey). However, to our knowledge, this is the first time the verifiability concept is defined and investigated in a ROM context.

The rest of the paper is organized as follows: In Section 5.3, we outline the construction of the standard G-ROM. In Sections 5.4 and 5.5, we use ROM spatial filtering to build LES-ROMs and utilize data-driven modeling to build the closure model in the DD-VMS-ROM, respectively. In Section 5.6, we prove the main theoretical result in this paper, i.e., we prove that the DD-VMS-ROM is verifiable. In Section 5.7, we illustrate the theoretical developments. Specifically, for the Burgers equation and the two-dimensional flow past a circular cylinder, we show the following: (i) the ROM closure error (i.e., the difference between the true ROM closure term and the DD-VMS-ROM closure term) is small and it becomes smaller and smaller as we increase the ROM dimension; and (ii) as the ROM closure error decreases, so does the ROM error (i.e., the DD-VMS-ROM is verifiable). Finally, in Section 5.8, we present the conclusions of our theoretical and numerical investigations and outline several directions for future research.

5.3 Galerkin ROM (G-ROM)

In this section, we outline the construction of the Galerkin ROM (G-ROM) for the Navier-Stokes equations (NSE):

$$\frac{\partial \boldsymbol{u}}{\partial t} - Re^{-1}\Delta \boldsymbol{u} + \boldsymbol{u}\cdot\nabla\boldsymbol{u} + \nabla p = \mathbf{f}\,, \tag{5.1}$$

$$\nabla\cdot\boldsymbol{u} = 0\,, \tag{5.2}$$

where $\boldsymbol{u}$ is the velocity, p the pressure, and Re the Reynolds number. The NSE (5.1)–(5.2) are equipped with an initial condition and, for simplicity, homogeneous Dirichlet boundary conditions. To build the ROM basis, we assume that we have access to the snapshots $\{\boldsymbol{u}_h^0, ..., \boldsymbol{u}_h^M\}$, which are the coefficient vectors of the FEM approximations of the NSE (5.1)–(5.2) at the time instances $t_0, t_1, \ldots, t_M$, respectively. The number of snapshots, M, is an arbitrary positive integer. In what follows, we assume that M is fixed. Next, we use

these snapshots and the proper orthogonal decomposition (POD) [9, 34] to construct an orthonormal ROM basis $\{\boldsymbol{\varphi}_1, ..., \boldsymbol{\varphi}_d\}$, which generates the ROM space $\boldsymbol{X}^d$ defined as follows:

$$\boldsymbol{X}^d := \text{span}\{\boldsymbol{\varphi}_1, ..., \boldsymbol{\varphi}_d\}, \tag{5.3}$$

where d is the number of linearly independent snapshots $\{\boldsymbol{u}_h^0, ..., \boldsymbol{u}_h^M\}$. Thus, d is the maximal dimension of a basis that spans the same space as the space spanned by the given snapshots. By using the ROM basis functions in (5.3), we construct $\boldsymbol{u}_d$, which is the d-dimensional ROM approximation of NSE velocity, $\boldsymbol{u}$:

$$\boldsymbol{u}_d(\boldsymbol{x}, t) = \sum_{i=1}^{d} (\boldsymbol{a}_d)_i(t)\, \boldsymbol{\varphi}_i(\boldsymbol{x}). \tag{5.4}$$

To find the vector of ROM coefficients $\boldsymbol{a}_d$ in (5.4), we use the Galerkin projection, i.e., we replace $\boldsymbol{u}$ with $\boldsymbol{u}_d$ in the NSE (5.1)–(5.2), and then project the resulting equations onto the ROM space, $\boldsymbol{X}^d$. This yields the d-dimensional Galerkin ROM (G-ROM):

$$((\boldsymbol{u}_d)_t, \boldsymbol{v}_d) + Re^{-1}(\nabla \boldsymbol{u}_d, \nabla \boldsymbol{v}_d) + (\boldsymbol{u}_d \cdot \nabla \boldsymbol{u}_d, \boldsymbol{v}_d) = (\boldsymbol{f}, \boldsymbol{v}_d) \qquad \forall\, \boldsymbol{v}_d \in \boldsymbol{X}^d. \tag{5.5}$$

We note that the G-ROM (5.5) does not include a pressure term, since the ROM basis functions are assumed to be discretely divergence-free (which is the case if, e.g., the snapshots are discretely divergence-free). We also note that, for simplicity, in the G-ROM (5.5) we used a nonlinearity formulation that is equivalent with the nonlinearity formulation in the NSE (5.1) when the velocity field is incompressible (i.e., it satisfies equation (5.2)).

By using the backward Euler time discretization, we get the full discretization of the d-dimensional G-ROM (5.5) as follows: $\forall\, n = 1, ..., M$

$$(\frac{\boldsymbol{u}_d^n - \boldsymbol{u}_d^{n-1}}{\Delta t}, \boldsymbol{v}_d) + Re^{-1}(\nabla \boldsymbol{u}_d^n, \nabla \boldsymbol{v}_d) + (\boldsymbol{u}_d^n \cdot \nabla \boldsymbol{u}_d^n, \boldsymbol{v}_d) = (\boldsymbol{f}^n, \boldsymbol{v}_d)\ \forall\, \boldsymbol{v}_d \in \boldsymbol{X}^d, \tag{5.6}$$

where the superscript n denotes the approximation at time step n. To obtain the finite-dimensional representation of the d-dimensional G-ROM (5.6), we choose $\boldsymbol{v}_d$ to be $\boldsymbol{\varphi}_1, \cdots, \boldsymbol{\varphi}_d$, which yields the following system of equations:

$$\frac{\boldsymbol{a}_d^n - \boldsymbol{a}_d^{n-1}}{\Delta t} = \boldsymbol{b}^n + \boldsymbol{A}\, \boldsymbol{a}_d^n + (\boldsymbol{a}_d^n)^\top\, \boldsymbol{B}\, \boldsymbol{a}_d^n, \tag{5.7}$$

where $\boldsymbol{a}_d^n$ is the vector of unknown ROM coefficients, $\boldsymbol{b}$ is a $d \times 1$ vector, $\boldsymbol{A}$ is a $d \times d$ matrix, and $\boldsymbol{B}$ is a $d \times d \times d$ tensor. The system of equations in (5.7) can be written componentwise as follows:

$$\frac{(\boldsymbol{a}_d^n)_i - (\boldsymbol{a}_d^{n-1})_i}{\Delta t} = \boldsymbol{b}^n + \sum_{m=1}^{d} \boldsymbol{A}_{im} \boldsymbol{a}_m^n + \sum_{m=1}^{d} \sum_{k=1}^{d} \boldsymbol{B}_{imk} \boldsymbol{a}_m^n \boldsymbol{a}_k^n\,, \quad 1 \le i \le d\,, \tag{5.8}$$

where, for $1 \le i, m, k \le d$,

$$\boldsymbol{b}_i^n = (\boldsymbol{f}^n, \boldsymbol{\varphi}_i), \tag{5.9}$$

$$\boldsymbol{A}_{im} = -Re^{-1}\, (\nabla \boldsymbol{\varphi}_m, \nabla \boldsymbol{\varphi}_i)\,, \tag{5.10}$$

$$\boldsymbol{B}_{imk} = -\big(\boldsymbol{\varphi}_m \cdot \nabla \boldsymbol{\varphi}_k, \boldsymbol{\varphi}_i\big)\,. \tag{5.11}$$

5.4 Large Eddy Simulation ROM (LES-ROM)

The ROM closure that we investigate in this paper (i.e., the DD-VMS-ROM presented in Section 5.5) is a large eddy simulation ROM (LES-ROM). Thus, in this section, we briefly outline the construction of LES-ROMs.

LES-ROMs are ROM closures that have been developed over the last decade (see, e.g., [35, 38]). LES-ROMs are utilizing mathematical principles used in classical LES [4, 31] to construct ROM closure models for ROMs in under-resolved regimes, i.e., when the number of ROM basis functions is insufficient to represent the complex dynamics of the underlying flows. Classical LES and LES-ROMs are similar in spirit: They both aim at approximating the large scales in the flow at the available coarse resolution (e.g., coarse mesh in classical LES and not enough ROM basis functions in LES-ROMs). Furthermore, they both use spatial filtering to define the large scales than need to be approximated. We emphasize, however, that there are also major differences between classical LES and LES-ROMs. One of the main differences is the type of spatial filtering used to define the large flow structures. In classical LES, continuous filters (e.g., the Gaussian filter) are used to define the filtered equations at a continuous level. In contrast, in LES-ROMs, due to the hierarchical structure of the ROM spaces, the ROM projection (which is a discrete spatial filter) is generally used instead. (For a notable exception, see the ROM differential filter, which is a continuous spatial ROM filter used in [38] to construct the approximate deconvolution ROM closure.) The ROM projection is used, in particular, to build variational multiscale (VMS) ROM closures, such as the closure that we investigate in this paper, which we describe next.

To construct the DD-VMS-ROM, we start by choosing the "truth" solution, i.e., the most accurate ROM solution that we can construct with the given snapshots.

Definition 5.1 (Truth Solution)**.** For fixed M and d, we define the d-dimensional G-ROM solution of (5.6) as our "truth" solution.

The goal of an LES-ROM is to construct an accurate ROM of dimension r, which is much smaller than the dimension of the "truth" solution (i.e., $r \ll d$). Since $r \ll d$, the LES-ROM development takes place in an under-resolved regime.

Thus, we use the LES-ROM framework to achieve the following: (i) use the ROM projection to define the large ROM spatial scales; (ii) Use the ROM projection to filter the d-dimensional G-ROM (5.6) to obtain the LES-ROM, i.e., the set of equations for the filtered ROM variables; and (iii) Finally, use data-driven modeling to construct a ROM closure model for the filtered ROM equations in step (ii). In this section, we discuss steps (i) and (ii); in the next section, we discuss step (iii), i.e., we construct the DD-VMS-ROM.

To define the large ROM scales and build the VMS framework, we first decompose the

d-dimensional ROM space $\boldsymbol{X}^d$ into two orthogonal subspaces

$$\boldsymbol{X}^r := \text{span}\{\boldsymbol{\varphi}_1, ..., \boldsymbol{\varphi}_r\}, \tag{5.12a}$$

$$(\boldsymbol{X}^r)^\perp := \text{span}\{\boldsymbol{\varphi}_{r+1}, ..., \boldsymbol{\varphi}_d\}, \tag{5.12b}$$

where $\boldsymbol{X}^r$ contains first r dominant ROM basis functions, and $(\boldsymbol{X}^r)^\perp$, which is orthogonal to $\boldsymbol{X}^r$, contains the less energetic ROM basis functions. We also define the following orthogonal projections:

Definition 5.2 (Orthogonal Projections)**.** Let $P_r : L^2 \to \boldsymbol{X}^r$ be the orthogonal projection onto $\boldsymbol{X}^r$, and $Q_r : L^2 \to (\boldsymbol{X}^r)^\perp$ be the orthogonal projection onto $(\boldsymbol{X}^r)^\perp$, which can be defined as

$$P_r(\boldsymbol{u}) = \sum_{i=1}^{r} (\boldsymbol{u}, \boldsymbol{\varphi}_i)_{L^2} \boldsymbol{\varphi}_i, \quad \boldsymbol{u} \in L^2, \tag{5.13a}$$

$$Q_r(\boldsymbol{u}) = \sum_{i=r+1}^{d} (\boldsymbol{u}, \boldsymbol{\varphi}_i)_{L^2} \boldsymbol{\varphi}_i, \quad \boldsymbol{u} \in L^2. \tag{5.13b}$$

Next, in the LES spirit, we decompose the most accurate ROM solution at time step n, $\boldsymbol{u}_d^n$ (i.e., the d-dimensional G-ROM solution (5.6), which is the "truth" solution that is employed as a benchmark in our investigation) as

$$\boldsymbol{u}_d^n := \underbrace{P_r(\boldsymbol{u}_d^n)}_{\text{large scales}} + \underbrace{Q_r(\boldsymbol{u}_d^n)}_{\text{small scales}}, \tag{5.14}$$

where P_r and Q_r are the two orthogonal projections in Definition 5.2. Equation (5.14) represents the LES-ROM decomposition of the "truth" solution, $\boldsymbol{u}_d^n$, into its large scale component, $P_r(\boldsymbol{u}_d^n)$, and its small scale component, $Q_r(\boldsymbol{u}_d^n)$.

The ROM spatial filter that we use to construct the LES-ROM is the ROM projection filter [25, 35], i.e., the orthogonal projection P_r defined in Definition 5.2, which satisfies the following equation: For given $\boldsymbol{u} \in L^2$,

$$\big(P_r(\boldsymbol{u}), \boldsymbol{\varphi}_i\big) = \big(\boldsymbol{u}, \boldsymbol{\varphi}_i\big), \qquad \forall\, i = 1, ..., r. \tag{5.15}$$

To construct the LES-ROM, we need to construct the equation satisfied by the large scales, $P_r(\boldsymbol{u}_d^n)$, defined in (5.14). We note that, by using Definition 5.2 and the ROM orthogonality property, we obtain the following formula for the large scale component $P_r(\boldsymbol{u}_d^n)$:

$$P_r(\boldsymbol{u}_d^n) = \sum_{i=1}^{r} (\boldsymbol{a}_d^n)_i \, \boldsymbol{\varphi}_i. \tag{5.16}$$

To construct the LES-ROM satisfied by $P_r(\boldsymbol{u}_d^n)$, we apply the ROM spatial filter, P_r, to the equation satisfied by the "truth" solution, $\boldsymbol{u}_d^n$ (i.e., to the full discretization of the

d-dimensional G-ROM (5.6)), we restrict the test functions in (5.6) to the r-dimensional ROM subspace $\boldsymbol{X}^r$ defined in (5.12a), and we use the decomposition (5.14). This yields the equations satisfied by the large scales, $P_r(\boldsymbol{u}_d^n)$, i.e., the LES-ROM equations:

$$\begin{aligned}\left(\frac{P_r(\boldsymbol{u}_d^n)-P_r(\boldsymbol{u}_d^{n-1})}{\Delta t},\boldsymbol{v}_r\right)+Re^{-1}(\nabla P_r(\boldsymbol{u}_d^n),\nabla\boldsymbol{v}_r)+(P_r(\boldsymbol{u}_d^n)\cdot\nabla P_r(\boldsymbol{u}_d^n),\boldsymbol{v}_r)\\ +\mathcal{E}^n+(\boldsymbol{\tau}^{FOM}(\boldsymbol{u}_d^n),\boldsymbol{v}_r)=(\boldsymbol{f}^n,\boldsymbol{v}_r)\,,\qquad\forall\,\boldsymbol{v}_r\in\boldsymbol{X}^r,\end{aligned}\tag{5.17}$$

where we used that, by (5.15), $(P_r(\boldsymbol{f}^n),\boldsymbol{v}_r)=(\boldsymbol{f}^n,\boldsymbol{v}_r)$. In the LES-ROM equations (5.17), the Reynolds stress tensor $\boldsymbol{\tau}^{FOM}(\boldsymbol{u}_d^n)$ and commutation error $\mathcal{E}$ are defined as follows:

$$\boldsymbol{\tau}^{FOM}(\boldsymbol{u}_d^n):=\boldsymbol{u}_d^n\cdot\nabla\boldsymbol{u}_d^n-P_r(\boldsymbol{u}_d^n)\cdot\nabla P_r(\boldsymbol{u}_d^n),\tag{5.18}$$

$$\mathcal{E}^n:=Re^{-1}(\nabla Q_r(\boldsymbol{u}_d^n),\nabla\boldsymbol{v}_r),\tag{5.19}$$

respectively. We note that, to obtain the LES-ROM equations (5.17), we used the fact that the term $(Q_r(\boldsymbol{u}_d^n),\boldsymbol{v}_r)$ vanishes since $Q_r(\boldsymbol{u}_d^n)$ is orthogonal to any vector in $\boldsymbol{X}^r$. We also note that the term $(\nabla Q_r(\boldsymbol{u}_d^n),\nabla\boldsymbol{v}_r)$ in the commutation error term (5.19) does not vanish since the ROM basis functions are only L^2-orthogonal, not H_0^1-orthogonal.

Remark 5.3 (Commutation Error)**.** In [17], we investigated the effect of the commutation error (5.19) on ROMs. We showed that the commutation error is genrally nonzero, but becomes negligible for large Re. Since our current investigation centers around LES-ROMs for turbulent flows, for simplicity, we do not consider the commutation error.

Definition 5.4 (Closure Model)**.** A closure model consists of replacing in (5.17) the Reynolds stress tensor $\boldsymbol{\tau}^{FOM}(\boldsymbol{u}_d^n)$ by another tensor $\boldsymbol{\tau}^{ROM}(P_r(\boldsymbol{u}_d^n))$ depending only on $P_r(\boldsymbol{u}_d^n)$.

Thus, the role of the closure model $\boldsymbol{\tau}^{ROM}$ is to replace the true closure model $\boldsymbol{\tau}^{FOM}(\boldsymbol{u}_d^n)$ (which cannot be computed in $\boldsymbol{X}^r$) with a term that can actually be computed in $\boldsymbol{X}^r$. Since a closure model cannot in general be exact (i.e., $\boldsymbol{\tau}^{FOM}(\boldsymbol{u}_d^n)\neq\boldsymbol{\tau}^{ROM}(P_r(\boldsymbol{u}_d^n))$), when $\boldsymbol{\tau}^{ROM}(P_r(\boldsymbol{u}_d^n))$ is inserted for $\boldsymbol{\tau}^{FOM}(\boldsymbol{u}_d^n)$ in (5.17) the solution of the resulting system is just an approximation to $P_r(\boldsymbol{u}_d^n)$. We denote this LES-ROM approximation to $P_r(\boldsymbol{u}_d^n)$ as $\boldsymbol{u}_r^n$, which can be written as

$$\boldsymbol{u}_r^n=\sum_{i=1}^{r}(\boldsymbol{a}_r^n)_i\,\boldsymbol{\varphi}_i.\tag{5.20}$$

Thus, the LES-ROM equations for $\boldsymbol{u}_r^n$ are

$$\begin{aligned}(\frac{\boldsymbol{u}_r^n-\boldsymbol{u}_r^{n-1}}{\Delta t},\boldsymbol{v}_r)+Re^{-1}(\nabla\boldsymbol{u}_r^n,\nabla\boldsymbol{v}_r)+(\boldsymbol{u}_r^n\cdot\nabla\boldsymbol{u}_r^n,\boldsymbol{v}_r)\\ +(\boldsymbol{\tau}^{ROM}(\boldsymbol{u}_r^n),\boldsymbol{v}_r)=(\boldsymbol{f}^n,\boldsymbol{v}_r),\quad\forall\,\boldsymbol{v}_r\in\boldsymbol{X}^r.\end{aligned}\tag{5.21}$$

Inserting (5.20) into (5.21) yields the following matrix form of the LES-ROM:

$$\frac{\boldsymbol{a}_r^n-\boldsymbol{a}_r^{n-1}}{\Delta t}=\boldsymbol{b}^n+\boldsymbol{A}\boldsymbol{a}_r^n+(\boldsymbol{a}_r^n)^T\boldsymbol{B}\boldsymbol{a}_r^n+[-(\boldsymbol{\tau}^{ROM}(\boldsymbol{u}_r^n),\boldsymbol{\varphi}_i)_{i=1,\ldots,r}],\tag{5.22}$$

where the vector $\boldsymbol{b}^n$, the matrix $\boldsymbol{A}$, and the tensor $\boldsymbol{B}$ are defined in (5.9)-(5.11).

5.5 Data Driven Variational Multiscale ROM (DD-VMS-ROM)

In this section, we outline the construction of the data-driven variational multiscale ROM (DD-VMS-ROM) closure model proposed in [23, 36]. We also describe the physical constraints that we add to the DD-VMS-ROM in order to increase its stability and accuracy. The construction of the DD-VMS-ROM is carried out within the LES-ROM framework described in Section 5.4.

To construct the DD-VMS-ROM, we start from the LES-ROM equations (5.22). First, we notice that since we used the ROM projection as a spatial filter, the LES-ROM (5.22) is in fact a variational multiscale ROM (VMS-ROM). However, the VMS-ROM (5.22) is not closed since the closure term $\boldsymbol{\tau}^{ROM}(\boldsymbol{u}_r^n)$ still needs to be determined. To construct a VMS-ROM closure model, we use data-driven modeling. Specifically, we first postulate a linear ansatz for the VMS-ROM closure term, and then we determine the parameters in the linear ansatz that best match the FOM data. The linear ansatz for the VMS-ROM closure term can be written as follows:

$$-(\boldsymbol{\tau}^{ROM}(\boldsymbol{u}_r^n), \boldsymbol{\varphi}_i)_{i=1,\dots,r} \approx \widetilde{\boldsymbol{A}}\, \boldsymbol{a}_r^n, \tag{5.23}$$

where $\boldsymbol{a}_r^n$ is vector of ROM coefficients of the solution $\boldsymbol{u}_r^n$. To determine the $r \times r$ matrix $\widetilde{\boldsymbol{A}}$ in (5.23), in the offline stage, we solve the following low-dimensional *least squares problem*:

$$\min_{\widetilde{\boldsymbol{A}}} \sum_{n=1}^{M} \left\| -\left[\left(\boldsymbol{u}_d^n \cdot \nabla \boldsymbol{u}_d^n - P_r(\boldsymbol{u}_d^n) \cdot \nabla P_r(\boldsymbol{u}_d^n)\,,\, \boldsymbol{\varphi}_i \right)_{i=1,\dots,r} \right] - \underbrace{\left(\widetilde{\boldsymbol{A}}\, \boldsymbol{a}_d^n\right)_{i=1,\dots,r}}_{:=(\boldsymbol{\tau}^{ROM}(P_r(\boldsymbol{u}_d)), \boldsymbol{\varphi}_i)_{i=1,\dots,r}} \right\|^2, \tag{5.24}$$

where $\boldsymbol{u}_d^n$ and $P_r(\boldsymbol{u}_d^n)$ are obtained from the available FOM data and are defined in (5.4) and (5.16), respectively.

Physical Constraint In the numerical investigation in [7], it was shown that, in the mean, the LES-ROM closure model dissipates energy. Thus, to mimic this behavior, in [21] we equipped the DD-VMS-ROM with a similar physical constraint. Specifically, in the least squares problem (5.24), we added the constraint that $\tilde{\boldsymbol{A}}$ be negative semidefinite:

$$(\boldsymbol{a}_r^n)^T \tilde{\boldsymbol{A}} \boldsymbol{a}_r^n \leq 0 \qquad \forall\, \boldsymbol{a}_r^n \in \mathbb{R}^r. \tag{5.25}$$

Solving the least squares problem (5.24) with the physical constraint (5.25), using the resulting matrix $\widetilde{\boldsymbol{A}}$ in the linear ansatz (5.23), and plugging this in the VMS-ROM (5.22) yields the data-driven variational multiscale ROM (DD-VMS-ROM):

$$\frac{\boldsymbol{a}_r^n - \boldsymbol{a}_r^{n-1}}{\Delta t} = \boldsymbol{b}^n + (\boldsymbol{A} + \widetilde{\boldsymbol{A}})\boldsymbol{a}_r^n + (\boldsymbol{a}_r^n)^T \boldsymbol{B} \boldsymbol{a}_r^n. \tag{5.26}$$

5.6 Verifiability of the DD-VMS-ROM

In this section, we prove the verifiability of the DD-VMS-ROM described in Section 5.5. In Section 5.6.1, we introduce the verifiability and mean dissipativity concepts in the ROM setting. In Section 5.6.2, we prove that the DD-VMS-ROM is verifiable.

5.6.1 Definition of Verifiability and Mean Dissipativity

The goal of this subsection is to define the verfiability of ROM closure models. Verifiability of closure models has been investigated for decades in classical CFD (see, e.g., [16] as well as [4] for a survey of verifiability methods in LES). We emphasize, however, that, to our knowledge, the verifiability concept has not been defined in a ROM context. In this section, we take a first step in this direction and define verifiability of ROM closure models. We also define the mean dissipativity of ROM closures, which will be used in Section 5.6.2 to prove the verifiability of the DD-VMS-ROM.

Definition 5.5 (Verifiability)**.** Let the number of snapshots, M, (and, thus, the number of linearly independent snapshots, d) be fixed. A ROM closure model is verifiable in the L^2 norm if there is a constant C such that, for all $r \leq d$ and for all $n = 1, \ldots, M$, the following *a priori* error bound holds:

$$\boxed{|| P_r(\boldsymbol{u}_d^n) - \boldsymbol{u}_r^n||_{L^2}^2 \leq C \frac{1}{n} \sum_{j=1}^{n} ||P_r(\, \boldsymbol{\tau}^{FOM}(\boldsymbol{u}_d^j) - \boldsymbol{\tau}^{ROM}(P_r(\boldsymbol{u}_d^j)\,)||_{L^2}^2,} \tag{5.27}$$

where $\boldsymbol{u}_d^j$ represents the "truth" solution (i.e., the d-dimensional G-ROM solution of (5.6)) at $t = t_j,\ j = 1, \ldots, M$, and $\boldsymbol{u}_r^n$ solves the ROM equipped with the given ROM closure model at $t = t_n,\ n = 1, \ldots, M$.

Definition 5.5 says that a ROM closure model is verifiable if a small average error in the ROM closure term implies a small error in the LES-ROM approximation.

Definition 5.6 (Mean Dissipativity)**.** A ROM closure model satisfies the mean dissipativity condition if $P_r(\boldsymbol{u}_d^n), \boldsymbol{u}_r^n \in \boldsymbol{X}^r$ satisfy the following inequalities:

$$\boxed{0 \leq (\boldsymbol{\tau}^{ROM}(P_r(\boldsymbol{u}_d^n)) - \boldsymbol{\tau}^{ROM}(\boldsymbol{u}_r^n)\,,\, P_r(\boldsymbol{u}_d^n) - \boldsymbol{u}_r^n) < \infty.} \tag{5.28}$$

5.6.2 Proof of DD-VMS-ROM's Verifiability

In this section, we first prove that the DD-VMS-ROM is mean dissipative. Then, we use this result to prove that the DD-VMS-ROM is verifiable.

Theorem 5.7. *The DD-VMS-ROM with linear ansatz (5.26) and physical constraint (5.25) satisfies mean dissipativity according to Definition 5.6.*

Proof. The least squares problem (5.24) yields the ROM operator $\tilde{\boldsymbol{A}}$ for $-(\boldsymbol{\tau}^{ROM}(P_r(\boldsymbol{u}_d^n), \boldsymbol{\varphi}_i)$, which is the VMS-ROM closure term. We emphasize that the same ROM operator $\tilde{\boldsymbol{A}}$ is used to construct the VMS-ROM closure term $-(\tau^{ROM}(\boldsymbol{u}_r^n), \boldsymbol{\varphi}_i)$. Specifically, the ROM operator $\tilde{\boldsymbol{A}}$ that is created by solving the least squares problem (5.24) for the VMS-ROM closure term $-(\boldsymbol{\tau}^{ROM}(P_r(\boldsymbol{u}_d^n), \boldsymbol{\varphi}_i)$ is used in the linear ansatz $-(\boldsymbol{\tau}^{ROM}(P_r(\boldsymbol{u}_d^n), \boldsymbol{\varphi}_i)_{i=1,\dots,r} \approx \tilde{\boldsymbol{A}}\, \boldsymbol{b}_r$, where $\boldsymbol{b}_r^n$ is an r-dimensional vector that contains the first r entries of the vector $\boldsymbol{a}_d^n$. The same ROM operator $\tilde{\boldsymbol{A}}$ is also used in the linear ansatz (5.23) for the VMS-ROM closure term $-(\boldsymbol{\tau}^{ROM}(\boldsymbol{u}_r^n), \boldsymbol{\varphi}_i)$: $-(\boldsymbol{\tau}^{ROM}(\boldsymbol{u}_r^n), \boldsymbol{\varphi}_i)_{i=1,\dots,r} \approx \tilde{\boldsymbol{A}}\, \boldsymbol{a}_r$. We approximate the VMS-ROM closure terms with these ansatzes and we obtain the following equalities:

$$\begin{aligned}(\boldsymbol{\tau}^{ROM}(P_r(\boldsymbol{u}_d^n)) - \boldsymbol{\tau}^{ROM}(\boldsymbol{u}_r^n)\,,\, \boldsymbol{\varphi}_i) &= \Big(\boldsymbol{\tau}^{ROM}(P_r(\boldsymbol{u}_d^n)\,,\, \boldsymbol{\varphi}_i\Big) - \Big(\boldsymbol{\tau}^{ROM}(\boldsymbol{u}_r^n)\,,\, \boldsymbol{\varphi}_i\Big) \\ &= (-\tilde{\boldsymbol{A}}\, \boldsymbol{b}_r^n)_i - (-\tilde{\boldsymbol{A}}\, \boldsymbol{a}_r^n)_i \\ &= \big(-\tilde{\boldsymbol{A}}\, (\boldsymbol{b}_r^n - \boldsymbol{a}_r^n)\big)_i \qquad \forall i = 1, .., r.\end{aligned} \tag{5.29}$$

To prove that the inner product $(\boldsymbol{\tau}^{ROM}(P_r(\boldsymbol{u}_d^n)) - \boldsymbol{\tau}^{ROM}(\boldsymbol{u}_r^n)\,,\, P_r(\boldsymbol{u}_d^n) - \boldsymbol{u}_r^n)$ is non-negative, we use the definitions of $P_r(\boldsymbol{u}_d^n)$ in (5.16) and $\boldsymbol{u}_r^n$ in (5.20) and rewrite it as follows:

$$\begin{aligned}&\Big(\boldsymbol{\tau}^{ROM}(P_r(\boldsymbol{u}_d^n)) - \boldsymbol{\tau}^{ROM}(\boldsymbol{u}_r^n)\,,\, P_r(\boldsymbol{u}_d^n) - \boldsymbol{u}_r^n\Big) \\ &= \Big(\boldsymbol{\tau}^{ROM}(P_r(\boldsymbol{u}_d^n)) - \boldsymbol{\tau}^{ROM}(\boldsymbol{u}_r^n)\,,\, \sum_{i=1}^{r} (\boldsymbol{a}_d^n - \boldsymbol{a}_r^n)_i\, \boldsymbol{\varphi}_i\Big) \\ &= \sum_{i=1}^{r} (\boldsymbol{a}_d^n - \boldsymbol{a}_r^n)_i \Big(\boldsymbol{\tau}^{ROM}(P_r(\boldsymbol{u}_d^n)) - \boldsymbol{\tau}^{ROM}(\boldsymbol{u}_r^n)\,,\, \boldsymbol{\varphi}_i\Big).\end{aligned} \tag{5.30}$$

By applying (5.29) to (5.30) and using the physical constraint (5.25), we get

$$\begin{aligned}(\boldsymbol{\tau}^{ROM}(P_r(\boldsymbol{u}_d^n)) - \boldsymbol{\tau}^{ROM}(\boldsymbol{u}_r^n)\,,\, P_r(\boldsymbol{u}_d^n) - \boldsymbol{u}_r^n) &= \sum_{i=1}^{r} (\boldsymbol{a}_d^n - \boldsymbol{a}_r^n)_i \big(-\tilde{\boldsymbol{A}}\, (\boldsymbol{b}_r^n - \boldsymbol{a}_r^n)\big)_i \\ &= -(\boldsymbol{b}_r^n - \boldsymbol{a}_r^n)^T\, \tilde{\boldsymbol{A}}\, (\boldsymbol{b}_r^n - \boldsymbol{a}_r^n) \geq 0,\end{aligned} \tag{5.31}$$

since $\tilde{\boldsymbol{A}}$ is negative semi-definite. In (5.31), we have used that $\boldsymbol{b}_r^n$ is an r-dimensional vector that contains the first r entries of the $\boldsymbol{a}_d^n$. The inequality in (5.31) concludes the proof. □

Remark 5.8. We note that in Theorem 5.7 we prove the ROM mean dissipativity property only for $P_r(\boldsymbol{u}_d^n)$ and $\boldsymbol{u}_r^n$. This is contrast with the FEM context, where mean dissipativity is proven for general FEM functions (see, e.g., [16]).

Next, we prove that the DD-VMS-ROM is verifiable. We note that, as explained in Section 5.4, the goal for the DD-VMS-ROM solution is to approximate as accurately as possible

$P_r(\boldsymbol{u}_d^n)$, which is the large scale component of the d-dimensional G-ROM solution (5.6), which is the "truth" solution that is employed as a benchmark in our investigation. We also note that $P_r(\boldsymbol{u}_d^n)$ satisfies the LES-ROM equations (5.17), which, for clarity, we rewrite below:

$$(\frac{P_r(\boldsymbol{u}_d^n) - P_r(\boldsymbol{u}_d^{n-1})}{\Delta t}, \boldsymbol{v}_r) + Re^{-1}(\nabla P_r(\boldsymbol{u}_d^n), \nabla \boldsymbol{v}_r) + (P_r(\boldsymbol{u}_d^n) \cdot \nabla P_r(\boldsymbol{u}_d^n), \boldsymbol{v}_r) + (\boldsymbol{\tau}^{FOM}(\boldsymbol{u}_d^n), \boldsymbol{v}_r) = (\boldsymbol{f}^n, \boldsymbol{v}_r), \tag{5.32}$$

where we used the fact that $(\boldsymbol{\tau}^{FOM}(\boldsymbol{u}_d^n), \boldsymbol{v}_r)$ is equal to $(P_r(\boldsymbol{\tau}^{FOM}(\boldsymbol{u}_d^n)), \boldsymbol{v}_r)$. We also rewrite the full discretization of the DD-VMS-ROM (5.21):

$$(\frac{\boldsymbol{u}_r^n - \boldsymbol{u}_r^{n-1}}{\Delta t}, \boldsymbol{v}_r) + Re^{-1}(\nabla \boldsymbol{u}_r^n, \nabla \boldsymbol{v}_r) + (\boldsymbol{u}_r^n \cdot \nabla \boldsymbol{u}_r^n, \boldsymbol{v}_r) + (\boldsymbol{\tau}^{ROM}(\boldsymbol{u}_r^n), \boldsymbol{v}_r) = (\boldsymbol{f}^n, \boldsymbol{v}_r). \tag{5.33}$$

Furthermore, we use the linear ansatz (5.23) and the physical constraints (5.25) for the ROM closure model in the DD-VMS-ROM (5.33). We also choose the initial condition $\boldsymbol{u}_r^0 = P_r(\boldsymbol{u}_d^0)$.

Thus, the DD-VMS-ROM error at time step n, which we denote with $\boldsymbol{e}^n$, is defined as the difference between the large scale component of the "truth" solution, $P_r(\boldsymbol{u}_d^n)$ (which is the solution of (5.32)), and the DD-VMS-ROM solution of (5.33), $\boldsymbol{u}_r^n$: $\boldsymbol{e}^n = P_r(\boldsymbol{u}_d^n) - \boldsymbol{u}_r^n$.

To prove the DD-VMS-ROM's verifiability, we use the following sharper bound on the nonlinear term, which is given in Lemma 22 in [19] (see also Lemma 61.1 in [32]):

Lemma 5.9. *Let $\Omega \subset \mathbb{R}^q$ be an open, bounded set of class C^2, with $q = 2$ or 3. For all $\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w} \in [\boldsymbol{H}_0^1(\Omega)]^q$,*

$$b(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}) \leq C(\Omega)\sqrt{||\boldsymbol{u}||\,||\nabla \boldsymbol{u}||}\,||\nabla \boldsymbol{v}||\,||\nabla \boldsymbol{w}||, \tag{5.34}$$

where the trilinear form $b(\cdot,\cdot,\cdot)$ [19, 33] is defined as

$$b(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}) = (\boldsymbol{u} \cdot \nabla \boldsymbol{v}, \boldsymbol{w}). \tag{5.35}$$

Theorem 5.10. *The DD-VMS-ROM (5.33) with linear ansatz (5.23), physical constraint (5.25), and the initial condition $\boldsymbol{u}_r^0 = P_r(\boldsymbol{u}_d^0)$ is verifiable: For a small enough time step, $\Delta t\, d_j < 1,\ \forall\, j = 1, ..., M$, where $d_j = \left(\frac{3ReC(\Omega)^2}{4}||\nabla P_r(\boldsymbol{u}_d^j)||^4 + Re\right)$ and $C(\Omega)$ is the constant in Lemma 5.9, the following inequality holds for all $n = 1, \ldots, M$:*

$$||\boldsymbol{e}^n||^2 + \Delta t \sum_{j=1}^{n} Re^{-1}||\nabla \boldsymbol{e}^j||^2 \leq \exp\left(\Delta t \sum_{j=1}^{n} \frac{d_j}{1 - \Delta t d_j}\right)\left(\Delta t \sum_{j=1}^{n} Re^{-1}||P_r(\boldsymbol{\tau}^{FOM}(\boldsymbol{u}_d^j) - \boldsymbol{\tau}^{ROM}(P_r(\boldsymbol{u}_d^j)))||^2\right). \tag{5.36}$$

Proof. We subtract (5.33) from (5.32), and replace n with j to get the error equation:

$$\begin{aligned}(\frac{\boldsymbol{e}^j-\boldsymbol{e}^{j-1}}{\Delta t},\boldsymbol{v}_r)+Re^{-1}(\nabla\boldsymbol{e}^j,\nabla\boldsymbol{v}_r)+b(P_r(\boldsymbol{u}_d^j),P_r(\boldsymbol{u}_d^j),\boldsymbol{v}_r)-b(\boldsymbol{u}_r^j,\boldsymbol{u}_r^j,\boldsymbol{v}_r)\\+\big(\boldsymbol{\tau}^{ROM}(P_r(\boldsymbol{u}_d^j))-\boldsymbol{\tau}^{ROM}(\boldsymbol{u}_r^j),\boldsymbol{v}_r\big)\\=-\big(\boldsymbol{\tau}^{FOM}(\boldsymbol{u}_d^j)-\boldsymbol{\tau}^{ROM}(P_r(\boldsymbol{u}_d^j)),\boldsymbol{v}_r\big).\end{aligned}\tag{5.37}$$

We set $\boldsymbol{v}_r=\boldsymbol{e}^j$ in (5.37), add and subtract $b(\boldsymbol{u}_r^j,P_r(\boldsymbol{u}_d^j),\boldsymbol{e}^j)$, and use the fact that $b(\boldsymbol{u}_r^j,\boldsymbol{e}^j,\boldsymbol{e}^j)=0$ to get the following equation:

$$\begin{aligned}\Delta t^{-1}(\boldsymbol{e}^j-\boldsymbol{e}^{j-1},\boldsymbol{e}^j)+Re^{-1}||\nabla\boldsymbol{e}^j||^2+b(\boldsymbol{e}^j,P_r(\boldsymbol{u}_d^j),\boldsymbol{e}^j)\\+(\boldsymbol{\tau}^{ROM}(P_r(\boldsymbol{u}_d^j))-\boldsymbol{\tau}^{ROM}(\boldsymbol{u}_r^j),\boldsymbol{e}^j)\\=-(\boldsymbol{\tau}^{FOM}(\boldsymbol{u}_d^j)-\boldsymbol{\tau}^{ROM}(P_r(\boldsymbol{u}_d^j)),\boldsymbol{e}^j).\end{aligned}\tag{5.38}$$

From Theorem 5.7, we have the following inequality:

$$(\boldsymbol{\tau}^{ROM}(P_r(\boldsymbol{u}_d^j))-\boldsymbol{\tau}^{ROM}(\boldsymbol{u}_r^j),\boldsymbol{e}^j)\geq 0.\tag{5.39}$$

Then by applying (5.39) to (5.38), we get the following inequality:

$$\begin{aligned}\Delta t^{-1}\big(\boldsymbol{e}^j-\boldsymbol{e}^j,\boldsymbol{e}^j\big)+Re^{-1}||\nabla\boldsymbol{e}^j||^2\leq -b(\boldsymbol{e}^j,P_r(\boldsymbol{u}_d^j),\boldsymbol{e}^j)\\-\big(\boldsymbol{\tau}^{FOM}(\boldsymbol{u}_d^j)-\boldsymbol{\tau}^{ROM}(P_r(\boldsymbol{u}_d^j)),\boldsymbol{e}^j\big).\end{aligned}\tag{5.40}$$

Applying Hölder's and Young's inequalities to the terms $(\boldsymbol{e}^j-\boldsymbol{e}^{j-1},\boldsymbol{e}^j)$ and $-(\boldsymbol{\tau}^{FOM}(\boldsymbol{u}_d^j)-\boldsymbol{\tau}^{ROM}(P_r(\boldsymbol{u}_d^j)),\boldsymbol{e}^j)$ in (5.40) we obtain that, for any $C_1,C_2>0$, the following inequalities hold:

$$\begin{aligned}(\boldsymbol{e}^j-\boldsymbol{e}^{j-1},\boldsymbol{e}^j)&=||\boldsymbol{e}^j||^2-(\boldsymbol{e}^j,\boldsymbol{e}^{j-1})\\&\geq||\boldsymbol{e}^j||^2-||\boldsymbol{e}^j||\,||\boldsymbol{e}^{j-1}||\\&\geq||\boldsymbol{e}^j||^2-\frac{C_1}{2}||\boldsymbol{e}^j||^2-\frac{1}{2C_1}||\boldsymbol{e}^{j-1}||^2\end{aligned}\tag{5.41}$$

and

$$\begin{aligned}&|-(\boldsymbol{\tau}^{FOM}(\boldsymbol{u}_d^j)-\boldsymbol{\tau}^{ROM}(P_r(\boldsymbol{u}_d^j)),\boldsymbol{e}^j)|\\&=|-(P_r(\boldsymbol{\tau}^{FOM}(\boldsymbol{u}_d^j)-\boldsymbol{\tau}^{ROM}(P_r(\boldsymbol{u}_d^j))),\boldsymbol{e}^j)|\\&\leq\frac{1}{2C_2}||P_r(\boldsymbol{\tau}^{FOM}(\boldsymbol{u}_d^j)-\boldsymbol{\tau}^{ROM}(P_r(\boldsymbol{u}_d^j)))\,||^2+\frac{C_2}{2}||\boldsymbol{e}^j||^2.\end{aligned}\tag{5.42}$$

Applying Lemma 5.9 to the term $-b(\boldsymbol{e}^j,P_r(\boldsymbol{u}_d^j),\boldsymbol{e}^j)$ we obtain the following inequality for any $C_3>0$:

$$\begin{aligned}|-b(\boldsymbol{e}^j,P_r(\boldsymbol{u}_d^j),\boldsymbol{e}^j)|&\leq C(\Omega)\,||\nabla\boldsymbol{e}^j||^{3/2}\,||\nabla P_r(\boldsymbol{u}_d^j)||\,||\boldsymbol{e}^j||^{1/2}\\&\leq\frac{3C_3C(\Omega)}{4}||\nabla\boldsymbol{e}^j||^2+\frac{C(\Omega)}{4C_3}||\nabla P_r(\boldsymbol{u}_d^j)||^4||\boldsymbol{e}j||^2,\end{aligned}\tag{5.43}$$

where $C(\Omega)$ is the constant in Lemma 5.9.

By choosing $C_1 = 1$, $C_2 = Re$, and $C_3 = 2Re^{-1}/3C(\Omega)$, we get the following inequality:

$$\begin{aligned}&\frac{1}{2\Delta t}(||\boldsymbol{e}^j||^2 - ||\boldsymbol{e}^{j-1}||^2) + \frac{Re^{-1}}{2}||\nabla \boldsymbol{e}^j||^2 \\ &\leq \Big(\frac{3ReC(\Omega)^2}{8}||\nabla P_r(\boldsymbol{u}_d^j)||^4 + \frac{Re}{2}\Big)||\boldsymbol{e}^j||^2 + \frac{Re^{-1}}{2}||P_r(\boldsymbol{\tau}^{FOM}(\boldsymbol{u}_d^j) - \boldsymbol{\tau}^{ROM}(P_r(\boldsymbol{u}_d^j)))\,||^2.\end{aligned} \tag{5.44}$$

By multiplying (5.44) by $2\Delta t$ and summing the resulting inequalities from $j = 1$ to n, we obtain the following inequality:

$$\begin{aligned}||\boldsymbol{e}^n||^2 + \Delta t\sum_{j=1}^{n} Re^{-1}||\nabla \boldsymbol{e}^j||^2 \leq ||\boldsymbol{e}^0||^2 + \Delta t\sum_{j=1}^{n}\Big(\frac{3ReC(\Omega)^2}{4}||\nabla P_r(\boldsymbol{u}_d^j)||^4 + Re\Big)||\boldsymbol{e}^j||^2 \\ +\Delta t\sum_{j=1}^{n} Re^{-1}||P_r(\boldsymbol{\tau}^{FOM}(\boldsymbol{u}_d^j) - \boldsymbol{\tau}^{ROM}(P_r(\boldsymbol{u}_d^j)))\,||^2.\end{aligned} \tag{5.45}$$

To apply the discrete Gronwall's lemma, we first make the following notation:

$$\begin{aligned}a_j &:= ||\boldsymbol{e}^j||^2 \geq 0,\\ b_j &:= Re^{-1}||\nabla \boldsymbol{e}^j||^2 \geq 0,\\ d_j &:= \Big(\frac{3ReC(\Omega)^2}{4}||\nabla P_r(\boldsymbol{u}_d^j)||^4 + Re\Big) \geq 0,\\ c_j &:= Re^{-1}||P_r(\boldsymbol{\tau}^{FOM}(\boldsymbol{u}_d^j) - \boldsymbol{\tau}^{ROM}(P_r(\boldsymbol{u}_d^j)))\,||^2 \geq 0,\\ H &:= ||\boldsymbol{e}^0||^2 \geq 0.\end{aligned} \tag{5.46}$$

We also recall that, by the small time step assumption, the following inequality holds: $\Delta t\, d_j < 1$, $\forall j$. By using the notation in (5.46), we rewrite (5.45) as follows:

$$a_n + \Delta t\sum_{j=1}^{n} b_j \leq \Delta t\sum_{j=1}^{n} d_j\, a_j + \Delta t\sum_{j=1}^{n} c_j + H. \tag{5.47}$$

By using the discrete Gronwall's lemma (see Lemma 27 in [19]) in (5.47), we obtain the following inequality:

$$a_n + \Delta t\sum_{j=1}^{n} b_j \leq \exp\Big(\Delta t\sum_{j=1}^{n}\frac{d_j}{1-\Delta t d_j}\Big)\Big(\Delta t\sum_{j=1}^{n} c_j + H\Big). \tag{5.48}$$

(We note that choosing the initial condition $\boldsymbol{u}_r^0 = P_r(\boldsymbol{u}_d^0)$, implies that $\boldsymbol{e}^0 = \boldsymbol{u}_r^0 - P_r(\boldsymbol{u}_d^0)$ and $H = 0$.) The inequality (5.48) proves (5.36). □

Remark 5.11. We note that the small time step assumption that we made in the theorem, i.e., that $\Delta t\, d_j < 1\ \forall j = 1, ..., M$, is also made in a FE context (see Lemma 27 and the proof of Theorem 24 in [19]).

Remark 5.12. In this paper, we used backward Euler time discretization to obtain the full discretizations of the ROMs. However, other time discretization schemes could be applied as well.

5.7 Numerical Results

In Theorem 5.10, we proved that the DD-VMS-ROM presented in Section 5.5 is verifiable. In this section, we present numerical support for the theoretical results in Theorem 5.10. In Section 5.7.1, we provide details on the numerical implementation of the DD-VMS-ROM. We numerically show that the DD-VMS-ROM is verifiable for the Burgers equation in Section 5.7.3 and for the flow past a cylinder in Section 5.7.4.

5.7.1 Numerical Implementation

"Truth" Solution For computational efficiency, instead of solving the G-ROM (5.5), which is a very large-dimensional system, to get the "truth" solution, $\boldsymbol{u}_d$, we simply project the FOM data on the ROM space, i.e., $\boldsymbol{u}_d = P_r(\boldsymbol{u}_h)$, $r = d$. In our numerical investigation, the two approaches yield similar results (i.e., the difference between the two approaches is on the order of the time discretization error). Thus, using the projection of the FOM data as "truth" solution does not affect our numerical investigation of the DD-VMS-ROM's verifiability.

Truncated SVD As is often the case in data-driven modeling [27], the least squares problem (5.24) that we need to solve in order to determine the entries in the ROM closure operator $\tilde{A}$ used to construct the DD-VMS-ROM (5.26) is ill conditioned. To alleviate the ill conditioning of the least squares problem, we proposed the use of the truncated SVD [23, 36] (see also [40] for a related approach). For completeness, in Algorithm 4, we outline the construction of the DD-VMS-ROM with the truncated SVD procedure.

The tolerance *tol* specified in step 3 of Algorithm 4 plays an important role in the numerical implementation of the DD-VMS-ROM. Specifying a large *tol* value yields a well conditioned least squares problem in step 1 and, as a result, minimizes the numerical errors in the least squares problem. However, a large *tol* value also decreases the accuracy of the least squares problem, i.e., yields a DD-VMS-ROM closure operator $\tilde{\boldsymbol{A}}$ that does not accurately match the FOM data. On the other hand, choosing a small *tol* value does not significantly decrease the accuracy of the DD-VMS-ROM closure operator $\tilde{\boldsymbol{A}}$, but does not significantly alleviate

the ill conditioning of the least sqaures problem either. In our numerical investigation, a careful choice of the tolerance *tol* yields optimal DD-VMS-ROM results.

Algorithm 4 Truncated SVD in Solving Least Square Problem

1: Formulate the standard linear least square problem for the unknown vector $\boldsymbol{x}_u$:

$$\min_{\boldsymbol{x}_u} \left\| E\boldsymbol{x}_u - \boldsymbol{f} \right\|^2, \tag{5.49}$$

where $E \in \mathbb{R}^{Mr \times r^2}$ is a matrix whose entries are determined by $\boldsymbol{a}_d(t_j), j = 1, \cdots, M$, $\boldsymbol{f} \in \mathbb{R}^{Mr \times 1}$ is a vector whose entries are determined by $P_r(\boldsymbol{\tau}^{FOM}(t_j))$, and $\boldsymbol{x}_u \in \mathbb{R}^{r^2 \times 1}, j = 1, \cdots, M$ is a vector whose entries are determined by $\tilde{\boldsymbol{A}}$.

2: Calculate the SVD of E:

$$E = U\Sigma V^\top. \tag{5.50}$$

3: Specify a tolerence *tol*.

4: Keep the entries in Σ that are larger than *tol*; the result matrix is $\widetilde{\Sigma}$ ($\widetilde{\sigma} = \sigma$ if $\sigma > tol$; also the singular values of E can be chosen as a *tol*).

5: Construct the truncated SVD of E, $\widetilde{E}$:

$$\widetilde{E} = \widetilde{U}\widetilde{\Sigma}\widetilde{V}^\top, \tag{5.51}$$

where $\widetilde{U}$ and $\widetilde{V}$ are the entries of U, V that correspond to $\widetilde{\Sigma}$.

6: The solution is given by

$$\boldsymbol{x}_u = \left(\widetilde{V}\widetilde{\Sigma}^{-1}\widetilde{U}^\top\right)\boldsymbol{f}. \tag{5.52}$$

Time Discretization Although the DD-VMS-ROM's verifiabiliy was proven in Theorem 5.10 for the backward Euler time discretization, in the numerical investigation in this section we are using two different time discretizations: Crank-Nicolson for the Burgers equation (Section 5.7.3) and the linearized BDF2 for the flow past a cylinder (Section 5.7.4). We use this higher-order time discretization in order to decrease the impact of the time discretization error onto the LES-ROM error, which is the main focus of the numerical investigation in this section. Furthermore, we believe that the mathematical arguments used to prove the DD-VMS-ROM's verifiabiliy in Theorem 5.10 can be extended to higher-order time discretizations such as those considered in this section.

Criteria To illustrate numerically the DD-VMS-ROM verifiability proven in Theorem 5.10, we use the following approach: First, we fix the number of snapshots, M. Therefore, the

maximal dimension of the ROM space, d, is also fixed. Furthermore, the "truth" solution $\boldsymbol{u}_d$ (i.e., the solution of the d-dimensional G-ROM (5.5)) is also fixed. The goal of our numerical investigation is to show that, for fixed M, d, and $\boldsymbol{u}_d$, there exists a constant C such that for varying r values and for varying tol values, the inequality (5.36) is satisfied.

To this end, we use the following metrics: To quantify the LES-ROM error, i.e., the term on the LHS of inequality (5.36), we use the following average L^2 norm:

$$\mathcal{E}(L^2) = \frac{1}{M}\sum_{n=1}^{M} \|P_r(\boldsymbol{u}_d^n) - \boldsymbol{u}_r^n\|^2 = \frac{1}{M}\sum_{n=1}^{M} \|\boldsymbol{e}^n\|^2 \,. \tag{5.53}$$

To quantify the LES-ROM closure error, i.e., the term on the RHS of inequality (5.36), we use the following metric:

$$\eta(L^2) = \frac{1}{M}\sum_{n=1}^{M} \left\|P_r(\boldsymbol{\tau}^{FOM}(\boldsymbol{u}_d^n) - \boldsymbol{\tau}^{ROM}(P_r(\boldsymbol{u}_d^n)\,)\,)\right\|_{L^2}^2 . \tag{5.54}$$

5.7.2 Assessment of Results

To illustrate numerically the DD-VMS-ROM verifiability proven in Theorem 5.10, we need to show that, for varying r values, as $\eta(L^2)$ in (5.54) decreases, so does $\mathcal{E}(L^2)$ in (5.53). To this end, for different r values, we decrease the tolerance in the truncated SVD algorithm to increase the accuracy of our LES-ROM closure term approximation and, therefore, to decrease $\eta(L^2)$.

We note that our numerical investigation is somewhat different from the standard investigations used in the numerical analysis literature. In our numerical investigation, we first consider several r values, and for each of these r values we decrease the tolerance used in the truncated SVD algorithm in order to decrease the LES-ROM closure term error, which is quantified by $\eta(L^2)$ in (5.54). Our hope is that, as $\eta(L^2)$ decreases, so does the corresponding LES-ROM error, which is quantified by $\mathcal{E}(L^2)$ in (5.53). Thus, our results do not illustrate the error convergence with respect to r (as is the case in standard numerical analysis papers). Instead, our numerical results aim at showing that, as $\eta(L^2)$ decreases, so does $\mathcal{E}(L^2)$.

5.7.3 Burgers Equation

In this section, we investigate the DD-VMS-ROM verifiability in the numerical simulation of the one-dimensional viscous Burgers equation:

$$\begin{cases} u_t - \nu u_{xx} + u u_x = 0 \,, \quad x \in [0,1],\ t \in [0,1], \\ u(0,t) = u(1,t) = 0 \,, \quad t \in (0,1], \\ u(x,0) = u_0(x) \,, \quad x \in [0,1], \end{cases} \tag{5.55}$$

with non-smooth initial condition (5.56):

$$u_0(x) = \begin{cases} 1, & x \in (0, 1/2], \\ 0, & x \in (1/2, 1]. \end{cases} \tag{5.56}$$

This test problem has been used in [1, 11, 18, 36].

Snapshot Generation We generate the FOM results by using a linear finite element (FE) spatial discretization with mesh size $h = 1/2048$, a Crank-Nicolson time discretization with timestep size $\Delta t = 10^{-3}$, and a viscosity coefficient $\nu = 10^{-3}$.

ROM Construction We run the FOM from $t = 0$ to $t = 1$. To generate the ROM basis functions, we collect a total of 1000 equally spaced snapshots. These snapshots are the FOM solutions from $t = 0$ to $t = 1$. To train the DD-VMS-ROM closure operator $\tilde{A}$, we use FOM data on the time interval $[0, 1]$. We test the DD-VMS-ROM on the time interval $[0, 1]$. Thus, we consider the reconstructive regime.

Numerical Results In Table 5.1, for three different r values, we list $\mathcal{E}(L^2)$ in (5.53), which measures the DD-VMS-ROM error, and $\eta(L^2)$ in (5.54), which measures the DD-VMS-ROM closure error. To compute $\mathcal{E}(L^2)$ and $\eta(L^2)$, we fix the r value and decrease the tolerance in the truncated SVD, which is used in the data-driven modeling part. As the tolerance decreases, we monitor the decaying rate of $\mathcal{E}(L^2)$ with respect to $\eta(L^2)$. The results in Table 5.1, for $r = 3, 7$, and 11, generally show that, as $\eta(L^2)$ decreases, so does $\mathcal{E}(L^2)$. In Figure 5.1, we plot the linear regression (LR) slope to understand the relation between $\mathcal{E}(L^2)$ and $\eta(L^2)$. For $r = 3, 7, 11$, the LR slope is around 3.

Overall, the results in Table 5.1 and Figure 5.1 support the theoretical results in Theorem 5.10.

$r = 3$		$r = 7$		$r = 11$	
$\eta(L^2)$	$\mathcal{E}(L^2)$	$\eta(L^2)$	$\mathcal{E}(L^2)$	$\eta(L^2)$	$\mathcal{E}(L^2)$
2.047e-01	1.131e-02	5.597e-01	3.230e-02	8.021e-01	1.516e-02
2.040e-01	1.121e-02	5.600e-01	3.187e-02	8.040e-01	1.490e-02
2.032e-01	1.109e-02	5.606e-01	3.135e-02	8.051e-01	1.466e-02
1.976e-01	1.048e-02	5.543e-01	2.906e-02	7.969e-01	1.396e-02
1.912e-01	9.150e-03	5.452e-01	2.678e-02	7.900e-01	1.329e-02
1.596e-01	4.203e-03	4.933e-01	1.706e-02	7.433e-01	9.463e-03
1.354e-01	3.070e-03	4.453e-01	1.101e-02	6.932e-01	6.822e-03
1.158e-01	2.123e-03	2.628e-01	1.667e-03	4.441e-01	1.705e-03

Table 5.1: Burgers equation (5.55), reconstructive regime: $\mathcal{E}(L^2)$ and $\eta(L^2)$ values for fixed r values and different tolerance values in the truncated SVD.

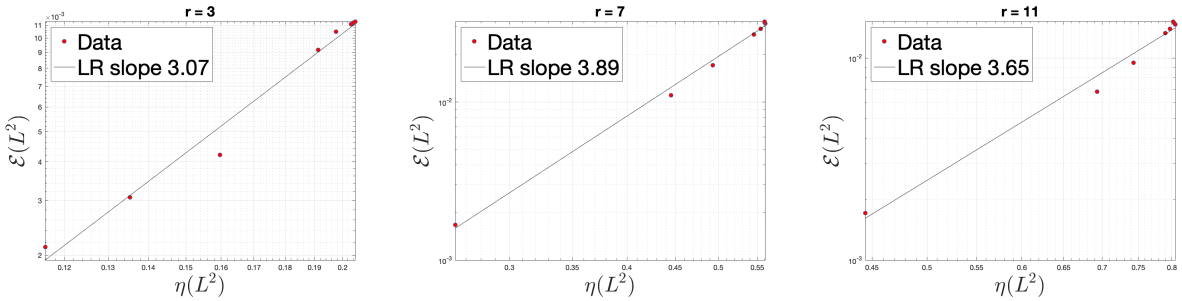

Figure 5.1: Burgers equation (5.55), reconstructive regime: linear regression for $\mathcal{E}(L^2)$ and $\eta(L^2)$ for fixed r values and different tolerance values in the truncated SVD.

5.7.4 Flow Past A Cylinder

In this section, we investigate the DD-VMS-ROM verifiability in the numerical simulation of a 2D channel flow past a circular cylinder at Reynolds numbers $Re = 100$ and $Re = 1000$. This test problem has been used in, e.g., [21, 23, 36].

Computational Setting As a mathematical model, we use the NSE (5.1)–(5.2). The computational domain is a 2.2×0.41 rectangular channel with a radius $= 0.05$ cylinder, centered at $(0.2, 0.2)$, see Figure 5.2.

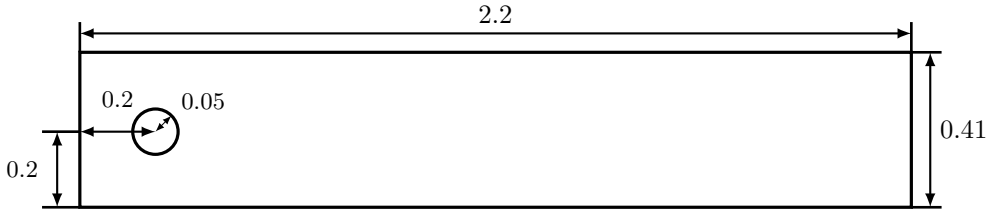

Figure 5.2: Geometry of the flow past a circular cylinder numerical experiment.

We prescribe no-slip boundary conditions on the walls and cylinder, and the following inflow and outflow profiles [14, 21, 22, 28]:

$$u_1(0, y, t) = u_1(2.2, y, t) = \frac{6}{0.41^2} y(0.41 - y), \tag{5.57}$$

$$u_2(0, y, t) = u_2(2.2, y, t) = 0, \tag{5.58}$$

where $\boldsymbol{u} = \langle u_1, u_2 \rangle$. There is no forcing and the flow starts from rest.

Snapshot Generation For the spatial discretization, we use the pointwise divergence-free, LBB stable (P_2, P_1^{disc}) Scott-Vogelius finite element pair on a barycenter refined regular triangular mesh [15]. The mesh yields $103K$ (102962) velocity and $76K$ (76725) pressure degrees of freedom. We use the linearized BDF2 temporal discretization and a time step size $\Delta t = 0.002$ for both FOM and ROM time discretizations. On the first time step, we use a backward Euler scheme so that we have two initial time step solutions required for the BDF2 scheme.

ROM Construction The FOM simulations achieve the statistically steady state at different time instances for the two Reynolds numbers used in the numerical investigation: For $Re = 100$, after $t = 5s$ and for $Re = 1000$, after $t = 13s$. To construct the ROM basis functions, we use $10s$ of FOM data. Thus, to ensure a fair comparison of the numerical results at different Reynolds numbers, we collect FOM snapshots on the following time intervals: For $Re = 100$, from $t = 7$ to $t = 17$ and for $Re = 1000$, from $t = 13$ to $t = 23$.

To train the DD-VMS-ROM closure operator $\tilde{\boldsymbol{A}}$, we use FOM data for one period. The period length of the statistically steady state is different for the two different Reynolds numbers: From $t = 7$ to $t = 7.332$ for $Re = 100$ and from $t = 13$ to $t = 13.268$ for $Re = 1000$. Thus, we collect 167 snapshots for $Re = 100$ and 135 snapshots for $Re = 1000$.

Numerical Results for $Re = 100$

In Table 5.2, for three different r values, we list $\mathcal{E}(L^2)$ in (5.53), which measures the DD-VMS-ROM error, and $\eta(L^2)$ in (5.54), which measures the DD-VMS-ROM closure error. To compute $\mathcal{E}(L^2)$ and $\eta(L^2)$, we fix the r value and decrease the tolerance in the truncated SVD, which is used in the data-driven modeling part. As the tolerance decreases, we monitor the decaying rate of $\mathcal{E}(L^2)$ with respect to $\eta(L^2)$. The results in Table 5.2, for $r = 4, 6$, and 8, generally show that, as $\eta(L^2)$ decreases, so does $\mathcal{E}(L^2)$.

$r = 4$		$r = 6$		$r = 8$	
$\eta(L^2)$	$\mathcal{E}(L^2)$	$\eta(L^2)$	$\mathcal{E}(L^2)$	$\eta(L^2)$	$\mathcal{E}(L^2)$
3.545e+03	1.231e-01	5.571e-01	4.186e-03	8.361e-01	4.334e-03
9.360e+02	8.742e-02	3.261e-01	3.223e-03	6.380e-01	4.062e-03
1.126e+01	1.536e-02	3.642e-02	1.323e-03	2.393e-02	2.368e-03
3.094e-02	4.021e-03	1.507e-03	1.837e-04	5.515e-03	2.593e-04
1.378e-02	9.042e-04	1.503e-03	1.802e-04	4.972e-03	1.225e-04
1.379e-02	9.043e-04	5.862e-05	5.407e-06	2.586e-03	3.106e-05
2.549e-04	2.313e-04	5.615e-05	5.382e-06	2.144e-04	4.229e-06

Table 5.2: Flow past a cylinder, $Re = 100$, reconstructive regime: $\mathcal{E}(L^2)$ and $\eta(L^2)$ values for fixed r values and different tolerance values in the truncated SVD.

In Figure 5.3, for $r = 4, 6$, and 8, we plot the LR slope for $\mathcal{E}(L^2)$ with respect to $\eta(L^2)$. For $r = 4$, the LR slope is 0.54, for $r = 6$ the LR slope is 0.94, and for $r = 8$ the LR slope is 1.18. These results indicate an almost linear correlation between $\mathcal{E}(L^2)$ and $\eta(L^2)$.

Overall, the results in Table 5.2 and Figure 5.3 support the theoretical results in Theorem 5.10.

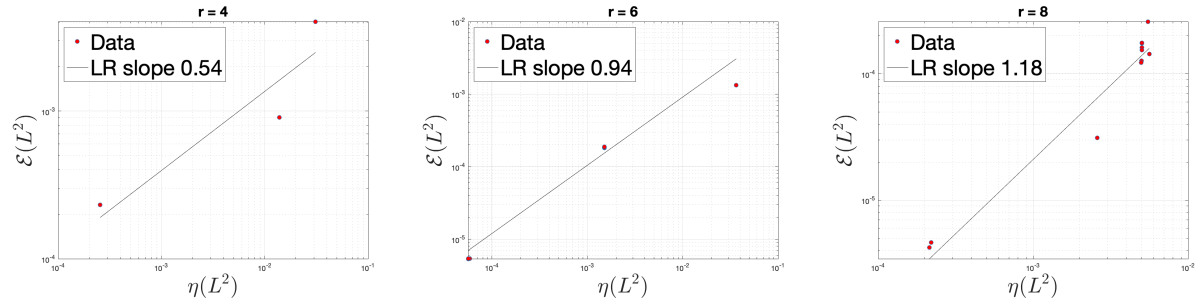

Figure 5.3: Flow past a cylinder, $Re = 100$, reconstructive regime: linear regression for $\mathcal{E}(L^2)$ and $\eta(L^2)$ for fixed r values and different tolerance values in the truncated SVD.

Numerical Results for $Re = 1000$

In Table 5.3, for three different r values, we list $\mathcal{E}(L^2)$ in (5.53), which measures the DD-VMS-ROM error, and $\eta(L^2)$ in (5.54), which measures the DD-VMS-ROM closure error. To compute $\mathcal{E}(L^2)$ and $\eta(L^2)$, we fix the r value and decrease the tolerance in the truncated SVD, which is used in the data-driven modeling part. As the tolerance decreases, we monitor the decaying rate of $\mathcal{E}(L^2)$ with respect to $\eta(L^2)$. The results in Table 5.3, for $r = 4, 6$, and 8, generally show that, as $\eta(L^2)$ decreases, so does $\mathcal{E}(L^2)$.

$r = 4$		$r = 6$		$r = 8$	
$\eta(L^2)$	$\mathcal{E}(L^2)$	$\eta(L^2)$	$\mathcal{E}(L^2)$	$\eta(L^2)$	$\mathcal{E}(L^2)$
3.679e+02	4.427e-01	3.265e+00	2.138e-01	2.684e+01	2.585e-02
1.966e+00	3.315e-01	1.927e+00	1.840e-01	7.423e+00	1.519e-02
1.757e+00	3.743e-02	1.017e+00	1.095e-01	1.550e+00	9.833e-03
7.410e-01	2.729e-01	7.261e-01	6.325e-02	6.149e-01	6.654e-03
7.400e-01	2.636e-01	9.313e-02	2.291e-02	2.586e-01	4.936e-03
5.783e-01	2.041e-02	5.425e-02	2.451e-03	9.122e-02	2.085e-03
4.991e-02	1.448e-03	2.899e-02	1.889e-04	3.382e-02	1.760e-04

Table 5.3: Flow past a cylinder, $Re = 1000$, reconstructive regime: $\mathcal{E}(L^2)$ and $\eta(L^2)$ values for fixed r values and different tolerance values in the truncated SVD.

In Figure 5.4, for $r = 4, 6$, and 8, we plot the LR slope for $\mathcal{E}(L^2)$ with respect to $\eta(L^2)$. For

$r = 4$, the LR slope is 1.10, for $r = 6$ the LR slope is 1.00, and for $r = 8$ the LR slope is 1.29. These results indicate an almost linear correlation between $\mathcal{E}(L^2)$ and $\eta(L^2)$.

Overall, the results in Table 5.3 and Figure 5.4 support the theoretical results in Theorem 5.10, which is indentical to the conclusion in Section 5.7.4.

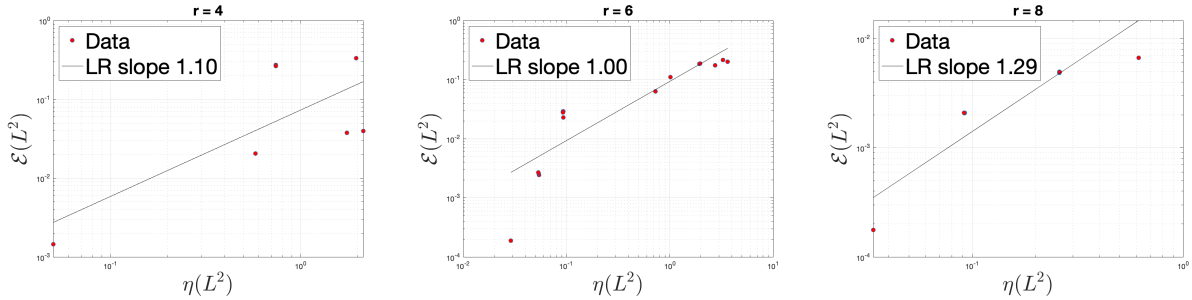

Figure 5.4: Flow past a cylinder, $Re = 1000$, reconstructive regime: linear regression for $\mathcal{E}(L^2)$ and $\eta(L^2)$ for fixed r values and different tolerance values in the truncated SVD.

5.8 Conclusions and Future Work

Over the last two decades, a plethora of ROM closure models have been developed for reduced order modeling of convection-dominated flows. Various ROM closure models have been constructed by using physical insight, mathematical arguments, or data. Although these ROM closure models are built by using different arguments, they are constructed by using the same *heuristic* algorithm: (i) In the offline stage, the ROM closure model is built so that it is as close as possible (in some norm) to the "true" ROM closure term. (ii) In the online stage, one needs to check whether the ROM closure model yields a ROM solution that is as close as possible to the filtered FOM solution. If the ROM solution is an accurate approximation of the filtered FOM solution, the ROM closure model is deemed accurate. This heuristic algorithm is the most popular approach used in assessing the success of the current ROM closure models. However, a natural question is whether one can actually *prove* anything about these ROM closure models. For example, can one prove that an accurate ROM closure model (constructed in the offline phase) yields an accurate ROM solution (in the online phase)?

In this paper, we took a step in this direction and we answered the above question by extended the verifiability concept from classical LES to a ROM setting. Specifically, we defined a ROM closure model as verifiable if the ROM error is bound (in some norm) by the ROM closure model error. Furthermore, we proved that a recently introduced data-driven ROM closure model (i.e., the DD-VMS-ROM [23, 36]) is verifiable. Finally, we showed numerically that the DD-VMS-ROM closure is verifiable. Specifically, in the numerical simulation of the one-dimensional Burgers equation and the two-dimensional flow past a circular cylinder at

Reynolds numbers $Re = 100$ and $Re = 1000$, we showed that by decreasing the error in the ROM closure term, we can achieve a decrease in the error in the ROM error, as predicted by the theoretical results.

There are several natural research directions that can be pursued in the quest to lay mathematical foundations for ROM closure models. For example, one could investigate the verifiability of (functional, structural, or data-driven) ROM closure models that are different from the DD-VMS-ROM investigated in this paper. One could also extend the verifiability concept to ROM closures that are built from experimental data. In that case, one could replace the high-dimensional "truth" solution used in this paper with the experimental solution interpolated onto a discrete mesh. Finally, one could consider other mathematical concepts that are used in classical LES (see, e.g., [4]) and extend them to a ROM setting.

Bibliography

[1] M. Ahmed and O. San. Stabilized principal interval decomposition method for model reduction of nonlinear convective systems with moving shocks. *Comp. Appl. Math.*, 37(5):6870–6902, 2018.

[2] F. Ballarin, T. C. Rebollo, E. D. Ávila, M. G. Mármol, and G. Rozza. Certified reduced basis vms-smagorinsky model for natural convection flow in a cavity with variable height. *Computers & Mathematics with Applications*, 80(5):973–989, 2020.

[3] L. C. Berselli, T. Iliescu, B. Koc, and R. Lewandowski. Long-time reynolds averaging of reduced order models for fluid flows: Preliminary results [j]. *Mathematics in Engineering*, 2(1):1–25, 2020.

[4] L. C. Berselli, T. Iliescu, and W. J. Layton. *Mathematics of Large Eddy Simulation of Turbulent Flows*. Scientific Computation. Springer-Verlag, Berlin, 2006.

[5] J. Borggaard, T. Iliescu, and Z. Wang. Artificial viscosity proper orthogonal decomposition. *Math. Comput. Modelling*, 53(1-2):269–279, 2011.

[6] M. D. Chekroun, H. Liu, and S. Wang. *Stochastic parameterizing manifolds and non-Markovian reduced equations: stochastic manifolds for nonlinear SPDEs II*. Springer, 2014.

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

[8] M. Gunzburger, T. Iliescu, and M. Schneier. A Leray regularized ensemble-proper orthogonal decomposition method for parameterized convection-dominated flows. *IMA J. Numer. Anal.*, 40(2):886–913, 2020.

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

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

[11] T. Iliescu and Z. Wang. Are the snapshot difference quotients needed in the proper orthogonal decomposition? *SIAM J. Sci. Comput.*, 36(3):A1221–A1250, 2014.

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

[13] V. John. *Large Eddy Simulation of Turbulent Incompressible Flows*, volume 34 of *Lecture Notes in Computational Science and Engineering*. Springer-Verlag, Berlin, 2004. Analytical and Numerical Results for a Class of LES Models.

[14] V. John. Reference values for drag and lift of a two dimensional time-dependent flow around a cylinder. *Int. J. Num. Meth. Fluids*, 44:777–788, 2004.

[15] V. John, A. Linke, C. Merdon, M. Neilan, and L. G. Rebholz. On the divergence constraint in mixed finite element methods for incompressible flows. *SIAM Rev.*, 2016.

[16] M. Kaya, W. Layton, et al. On" verifiability" of models of the motion of large eddies in turbulent flows. *Differential and Integral Equations*, 15(11):1395–1407, 2002.

[17] B. Koc, M. Mohebujjaman, C. Mou, and T. Iliescu. Commutation error in reduced order modeling of fluid flows. *Adv. Comput. Math.*, 45(5-6):2587–2621, 2019.

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

[19] W. Layton. *Introduction to the numerical analysis of incompressible viscous flows*. SIAM, 2008.

[20] F. Lu. Data-driven model reduction for stochastic Burgers equations. *Entropy*, 22(12):1360.

[21] M. Mohebujjaman, L. G. Rebholz, and T. Iliescu. Physically-constrained data-driven correction for reduced order modeling of fluid flows. *Int. J. Num. Meth. Fluids*, 89(3):103–122, 2019.

[22] M. Mohebujjaman, L. G. Rebholz, X. Xie, and T. Iliescu. Energy balance and mass conservation in reduced order models of fluid flows. *J. Comput. Phys.*, 346:262–277, 2017.

[23] C. Mou, B. Koc, O. San, L. G. Rebholz, and T. Iliescu. Data-driven variational multiscale reduced order models. *Computer Methods in Applied Mechanics and Engineering*, 373:113470.

[24] C. Mou, H. Liu, D. R. Wells, and T. Iliescu. Data-driven correction reduced order models for the quasi-geostrophic equations: A numerical investigation. *Int. J. Comput. Fluid Dyn.*, pages 1–13, 2020.

[25] A. A. Oberai and J. Jagalur-Mohan. Approximate optimal projection for reduced-order models. *Int. J. Num. Meth. Engng.*, 105(1):63–80, 2016.

[26] E. J. Parish and K. Duraisamy. A unified framework for multiscale modeling using the Mori-Zwanzig formalism and the variational multiscale method. *arXiv preprint, http://arxiv.org/abs/1712.09669*, 2017.

[27] B. Peherstorfer and K. Willcox. Data-driven operator inference for nonintrusive projection-based model reduction. *Comput. Methods Appl. Mech. Engrg.*, 306:196–215, 2016.

[28] L. Rebholz and M. Xiao. Improved accuracy in algebraic splitting methods for Navier-Stokes equations. *SIAM J. Sci. Comput.*, 39(4):A1489–A1513, 2017.

[29] T. C. Rebollo, E. D. Ávila, M. G. Mármol, F. Ballarin, and G. Rozza. On a certified Smagorinsky reduced basis turbulence model. *SIAM J. Numer. Anal.*, 55(6):3047–3067, 2017.

[30] T. C. Rebollo and R. Lewandowski. *Mathematical and Numerical Foundations of Turbulence Models and Applications*. Springer, 2014.

[31] P. Sagaut. *Large Eddy Simulation for Incompressible Flows*. Scientific Computation. Springer-Verlag, Berlin, third edition, 2006.

[32] G. R. Sell and Y. You. *Dynamics of evolutionary equations*, volume 143. Springer Science & Business Media, 2013.

[33] R. Temam. *Navier-Stokes equations: Theory and numerical analysis*, volume 2. American Mathematical Society, 2001.

[34] S. Volkwein. Proper orthogonal decomposition: Theory and reduced-order modelling. *Lecture Notes, University of Konstanz*, 2013. http://www.math.uni-konstanz.de/numerik/personen/volkwein/teaching/POD-Book.pdf.

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

[36] X. Xie, M. Mohebujjaman, L. G. Rebholz, and T. Iliescu. Data-driven filtered reduced order modeling of fluid flows. *SIAM J. Sci. Comput.*, 40(3):B834–B857, 2018.

[37] X. Xie, C. Webster, and T. Iliescu. Closure learning for nonlinear model reduction using deep residual neural network. *Fluids*, 5(1):39, 2020.

[38] X. Xie, D. Wells, Z. Wang, and T. Iliescu. Approximate deconvolution reduced order modeling. *Comput. Methods Appl. Mech. Engrg.*, 313:512–534, 2017.

[39] X. Xie, D. Wells, Z. Wang, and T. Iliescu. Numerical analysis of the Leray reduced order model. *J. Comput. Appl. Math.*, 328:12–29, 2018.

[40] S. Yıldız, P. Goyal, P. Benner, and B. Karasozen. Data-driven learning of reduced-order dynamics for a parametrized shallow water equation. *arXiv preprint arXiv:2007.14079*, 2020.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

In this dissertation, we addressed the pointwise optimality of the classical ROM in Chapter 2, the numerical accuracy of a data-driven ROM in Chapter 3, the time-average of the exchange of energy among ROM modes in Chapter 4, and the verifiability of a data-driven ROM closure model in Chapter 5.

In Chapter 2, we studied the effect of the DQs in the optimality of the pointwise projection and error bounds with respect to the time discretization and ROM discretization errors. For both the theoretical and numerical investigations, we used the heat equation. We theoretically and numerically proved that in the noDQ case, the pointwise projection and error bounds are suboptimal; however, when the DQs are used, the pointwise projection and error bounds are optimal with respect to both the ROM discretization and time discretization errors.

In Chapter 3, we proposed a new data-driven variational multiscale reduced order model (DD-VMS-ROM) framework. Using the VMS methodology and the ROM basis' hierarchical structure, we built the closure term, which models the interaction among the ROM spatial scales. We built two DD-VMS-ROMs, i.e., the two-scale DD-VMS-ROM (2S-DD-VMS-ROM) and the three-scale DD-VMS-ROM (3S-DD-VMS-ROM). In the 2S-DD-VMS-ROM, we decomposed the scales as resolved and unresolved. Thus, we constructed just one closure term to model the interaction between two different scales. However, in the 3S-DD-VMS-ROM, since we decomposed the scales into three parts, i.e., the resolved large, resolved small scales, and unresolved scales, we constructed two closure terms to model the interaction among the three different scales. Our numerical investigation showed that both the 2S-DD-VMS-ROM and the 3S-DD-VMS-ROM are more accurate than the standard Galerkin ROM (G-ROM), and the 3S-DD-VMS-ROM was generally more accurate than the 2S-DD-VMS-ROM.

In Chapter 4, we investigated theoretically and numerically the time-average of the exchange of energy among ROM modes of fluid flows. We considered two types of ROM modes: Eigenfunctions of the Stokes operator and proper orthogonal decomposition (POD) modes. In Theorem 4.4 and Theorem 4.6, we proved analytical results for both types of ROM modes, and we highlighted the differences between them. In Section 4.4, we used a one-dimensional Burgers equation as a mathematical model to show numerically that the time-average energy exchange between the most energetic POD modes and the least energetic POD modes is positive. Furthermore, we showed that for a longer time interval, the time-average energy exchange is still positive. However, for a short time interval, e.g., [0,0.1], the time-average energy exchange is negative.

In Chapter 5, we presented the first numerical analysis work for the DD-VMS-ROM closure. In this dissertation, we investigated under which conditions a small difference between the ROM closure term and the FOM closure term implies that the ROM solution is close to the FOM solution. In other words, we theoretically proved and numerically demonstrated

the verifiability of the data-driven ROM closure model. For both the numerical and the theoretical investigations, we used the ROM projection filter and the data-driven closure model with a linear ansatz.

6.2 Future Work

There are several research directions that we plan to pursue for each chapter.

Pointwise Optimality of the Classical ROM Several research directions need to be investigated related to the optimality of the error bounds. One future direction is to extend optimal uniform estimates to more complicated nonlinear PDEs (e.g., the Navier-Stokes equations). Another direction is to investigate the effect of the POD adaptivity, which allows choosing snapshot time instances optimally to minimize the error between the ROM and FOM trajectories. In this dissertation, we focused on the optimality of the rates of convergence of ROM error bounds; however, we did not address the size of the ROM error. Thus, the relation between the ROM error size in the noDQ and DQ cases should also be investigated. Finally, we plan to investigate the extension of these results to 2D and 3D problems as well as to nonlinear problems, e.g., the NSE.

DD-VMS ROMs Several research directions need to be investigated related to the DD-VMS-ROM framework. The first research direction is finding the optimal parameter r_1 (which determines the decomposition of the resolved scales into large resolved scales and small resolved scales) and the optimal tolerances tol_L and tol_S in the new 3S-DD-VMS-ROM. Another research direction that we plan to pursue is the development of new DD-VMS-ROM closure models by leveraging ideas from VMS methods for finite element discretizations (see, e.g., Section 8.8 in [6]), e.g., the time-dependent subscale-orthogonal methods [4, 9, 10]. We also plan to explore different topological structures for the ROM closure term. We emphasize that, without loss of generality, our DD-VMS-ROM framework can be formulated by utilizing a supervised machine learning approach [8, 11, 12, 13], a topic that we would like to explore in the future. Furthermore, we intend to explore the extension of the new DD-VMS-ROM to the Petrov-Galerkin framework [2, 3, 5, 7]. Finally, we plan to investigate the role played by residuals in the DD-VMS-ROM framework.

Long-time Averaging of ROMs The most important direction that we plan to pursue is the numerical investigation of the theoretical results in three-dimensional, high Reynolds number flows, which could shed new light on the energy transfer among ROM modes. Furthermore, we will also investigate the time-average energy exchange for general short time intervals, which are not necessarily located at the beginning of the simulation (as in the present

study). Finally, we plan to investigate the potential connection between time-averaging and Reynolds-averaged Navier-Stokes (RANS).

Verifiability of the DD-VMS ROM As future work, we plan to investigate the verifiability of the DD-VMS-ROM closure with a quadratic ansatz and the ROM differential filter. Furthermore, we will investigate whether this type of numerical analysis can be performed for different functional, structural, or data-driven ROM closure models, e.g., the Smagorinsky eddy viscosity and the approximate deconvolution closure models. Finally, one could consider other mathematical concepts that are used in classical LES (see, e.g., [1]) and extend them to a ROM setting.

Bibliography

[1] L. C. Berselli, T. Iliescu, and W. J. Layton. *Mathematics of Large Eddy Simulation of Turbulent Flows*. Scientific Computation. Springer-Verlag, Berlin, 2006.

[2] K. Carlberg, M. Barone, and H. Antil. Galerkin v. least-squares Petrov–Galerkin projection in nonlinear model reduction. *J. Comput. Phys.*, 330:693–734, 2017.

[3] K. Carlberg, C. Bou-Mosleh, and C. Farhat. Efficient non-linear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximations. *Int. J. Num. Meth. Eng.*, 86(2):155–181, 2011.

[4] R. Codina. Stabilized finite element approximation of transient incompressible flows using orthogonal subscales. *Comput. Methods Appl. Mech. Engrg.*, 191(39-40):4295–4321, 2002.

[5] S. Grimberg, C. Farhat, and N. Youkilis. On the stability of projection-based model order reduction for convection-dominated laminar and turbulent flows. *arXiv preprint,http://arxiv.org/abs/2001.10110*, 2020.

[6] V. John. *Finite element methods for incompressible flow problems*. Springer, 2016.

[7] E. J. Parish, C. Wentland, and K. Duraisamy. The adjoint Petrov-Galerkin method for non-linear model reduction. *arXiv preprint arXiv:1810.03455*, 2019.

[8] S. M. Rahman, O. San, and A. Rasheed. A hybrid approach for model order reduction of barotropic quasi-geostrophic turbulence. *Fluids*, 3(4):86, 2018.

[9] R. Reyes and R. Codina. Projection-based reduced order models for flow problems: A variational multiscale approach. *Comput. Methods Appl. Mech. Engrg.*, 363:112844, 2020.

[10] R. Reyes, R. Codina, J. Baiges, and S. Idelsohn. Reduced order models for thermally coupled low mach flows. *Adv. Model. Simul. Eng. Sci.*, 5(1):28, 2018.

[11] O. San and R. Maulik. Extreme learning machine for reduced order modeling of turbulent geophysical flows. *Phys. Rev. E*, 97(4):042322, 2018.

[12] O. San and R. Maulik. Machine learning closures for model order reduction of thermal fluids. *Appl. Math. Model.*, 60:681–710, 2018.

[13] O. San and R. Maulik. Neural network closures for nonlinear model order reduction. *Adv. Comput. Math.*, 44(6):1717–1750, 2018.