Data-Driven Variational Multiscale Reduced Order Models of Turbulent Flows

Changhong Mou

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 Borggaard
John A. Burns
Honghu Liu

May 7, 2021
Blacksburg, Virginia

Keywords: Reduced Order Models, Closure Modeling, Variational Multiscale, Geophysical Fluids, Turbulence, Eddy Viscosity, Finite Time Lyapunov Exponent, Proper Orthogonal Decomposition, Data-Driven Modeling

Copyright 2021, Changhong Mou
Data-Driven Variational Multiscale Reduced Order Models of Turbulent Flows

Changhong Mou

ABSTRACT

In this dissertation, we consider two different strategies for improving the projection-based reduced order model (ROM) accuracy: (I) adding closure terms to the standard ROM; and (II) using Lagrangian data to improve the ROM basis.

Following strategy (I), 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 quasigeostrophic 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.

Furthermore, we propose a new hybrid ROM framework for the numerical simulation of fluid flows. This hybrid framework incorporates two closure modeling strategies: (i) A structural closure modeling component that involves the recently proposed data-driven variational multiscale ROM approach, and (ii) A functional closure modeling component that introduces an
artificial viscosity term. We also utilize physical constraints for the structural ROM operators in order to add robustness to the hybrid ROM. We perform a numerical investigation of the hybrid ROM for the three-dimensional turbulent channel flow at a Reynolds number $Re = 13,750$.

In addition, we focus on the mathematical foundations of 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.

For strategy (II), we propose new Lagrangian inner products that we use together with Eulerian and Lagrangian data to construct new Lagrangian ROMs. We show that the new Lagrangian ROMs are orders of magnitude more accurate than the standard Eulerian ROMs, i.e., ROMs that use standard Eulerian inner product and data to construct the ROM basis. Specifically, for the quasi-geostrophic equations, we show that the new Lagrangian ROMs are more accurate than the standard Eulerian ROMs in approximating not only Lagrangian fields (e.g., the finite time Lyapunov exponent (FTLE)), but also Eulerian fields (e.g., the streamfunction). We emphasize that the new Lagrangian ROMs do not employ any closure modeling to model the effect of discarded modes (which is standard procedure for low-dimensional ROMs of complex nonlinear systems). Thus, the dramatic increase in the new Lagrangian ROMs’ accuracy is entirely due to the novel Lagrangian inner products used to build the Lagrangian ROM basis.
Data-Driven Variational Multiscale Reduced Order Models of Turbulent Flows

Changhong Mou

GENERAL AUDIENCE ABSTRACT

Reduced order models (ROMs) are popular in physical and engineering applications: for example, ROMs are widely used in aircraft designing as it can greatly reduce computational cost for the aircraft’s aeroelastic predictions while retaining good accuracy. However, for high Reynolds number turbulent flows, such as blood flows in arteries, oil transport in pipelines, and ocean currents, the standard ROMs may yield inaccurate results. In this dissertation, to improve ROM’s accuracy for turbulent flows, we investigate three different types of ROMs. In this dissertation, both numerical and theoretical results show that the proposed new ROMs yield more accurate results than the standard ROM and thus can be more useful.
To my parents.
Acknowledgments

First and foremost, I would like to express my sincere gratitude to my advisor Dr. Traian Iliescu for the inspiring guidance of my Ph.D. study and research. He leads me into this fantastic world of reduced order modeling, and I have always benefited from his knowledge, enthusiasm, patience, and constant support. I would not have gotten this far without the help and support from my advisor and I could not have imagined having a better advisor and mentor for my Ph.D. study.

I am also deeply grateful to Dr. Honghu Liu for many valuable comments and suggestions in my study and research. Special thanks to Dr. John A. Burns for offering me a wonderful summer internship experience at UTC CCS and for introducing me to diverse projects in real-world applications. I would also like to thank Dr. Jeffrey Borggaard for his comments and encouragement.

In addition, I wish to express my heartful thanks to Dr. Leo G. Rebohlz, Dr. Shane D. Ross and Dr. Adrian Sandu; I have learned a lot from each of them which widen my research from various perspectives.

I would also like to thank my academic family: Dr. Zhu Wang, Dr. David R. Wells, Dr. Xuping Xie, and Birgul Koc as well as Andrey A. Popov from department of computer science for their help and suggestions in my graduate study.
# Contents

List of Algorithms xii  
List of Figures xiii  
List of Tables xvii  

1 Introduction 1  
1.1 Motivation ................................................. 1  
1.2 Turbulent Flows ............................................ 2  
1.2.1 Navier-Stokes Equations ............................... 3  
1.2.2 Quasi-Geostrophic Equations ......................... 4  
1.3 Proper Orthogonal Decomposition ....................... 4  
1.4 Galerkin ROM .............................................. 6  
1.4.1 G-ROM for NSE ......................................... 7  
1.4.2 G-ROM for QGE ......................................... 8  
1.5 Variational Multiscale Closure Modeling ............... 10  
1.6 Outline of Dissertation .................................. 11  
Bibliography .................................................. 14  

2 Data-Driven Variational Multiscale Reduced Order Models 22  
2.1 Introduction ................................................. 23  
2.2 Data-Driven Variational Multiscale Reduced Order Models (DD-VMS-ROMs) 26  
2.2.1 Classical VMS ........................................... 27
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2.2</td>
<td>Galerkin ROM (G-ROM)</td>
<td>28</td>
</tr>
<tr>
<td>2.2.3</td>
<td>Two-Scale Data-Driven Variational Multiscale ROMs (2S-DD-VMS-ROM)</td>
<td>29</td>
</tr>
<tr>
<td>2.2.4</td>
<td>Three-Scale Data-Driven Variational Multiscale ROMs (3S-DD-VMS-ROM)</td>
<td>32</td>
</tr>
<tr>
<td>2.3</td>
<td>Numerical Results</td>
<td>34</td>
</tr>
<tr>
<td>2.3.1</td>
<td>Computational Setting</td>
<td>36</td>
</tr>
<tr>
<td>2.3.2</td>
<td>Burgers Equation</td>
<td>40</td>
</tr>
<tr>
<td>2.3.3</td>
<td>Flow Past A Cylinder</td>
<td>47</td>
</tr>
<tr>
<td>2.3.4</td>
<td>Quasi-Geostrophic Equations (QGE)</td>
<td>64</td>
</tr>
<tr>
<td>2.3.5</td>
<td>Backward Facing Step</td>
<td>69</td>
</tr>
<tr>
<td>2.3.6</td>
<td>Qualitative Comparison of 2S-DD-VMS-ROM and 3S-DD-VMS-ROM</td>
<td>75</td>
</tr>
<tr>
<td>2.4</td>
<td>Conclusions and Outlook</td>
<td>78</td>
</tr>
<tr>
<td></td>
<td>Bibliography</td>
<td>79</td>
</tr>
<tr>
<td>3</td>
<td>Physically Constrained DD-VMS-ROMs</td>
<td>88</td>
</tr>
<tr>
<td>3.1</td>
<td>Introduction</td>
<td>89</td>
</tr>
<tr>
<td>3.1.1</td>
<td>Connections to Previous Work</td>
<td>92</td>
</tr>
<tr>
<td>3.2</td>
<td>Data-Driven Correction ROM (DDC-ROM)</td>
<td>93</td>
</tr>
<tr>
<td>3.2.1</td>
<td>Quasi-geostrophic Equations (QGE)</td>
<td>93</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Standard Galerkin ROM (G-ROM)</td>
<td>94</td>
</tr>
<tr>
<td>3.2.3</td>
<td>DDC-ROM</td>
<td>96</td>
</tr>
<tr>
<td>3.3</td>
<td>Numerical Experiments</td>
<td>98</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Computational Setting and Snapshot Generation</td>
<td>98</td>
</tr>
<tr>
<td>3.3.2</td>
<td>ROM Construction</td>
<td>99</td>
</tr>
</tbody>
</table>
5.3.2 3S-DD-VMS-ROM ........................................ 149
5.3.3 Choices of generic functions ........................................ 150
5.3.4 Constrained DD-VMS-ROM ........................................ 151
5.4 Hybrid Data-Driven ROMs ........................................ 154
5.5 Numerical Results ........................................ 155
  5.5.1 Numerical Setting ........................................ 155
  5.5.2 Numerical Criteria ........................................ 157
  5.5.3 Qualitative Summary of Numerical Results ............ 159
  5.5.4 Numerical Results for $r = 10$ ........................................ 161
Bibliography ........................................ 164

6 Verifiability of Hybrid Data-Driven ROMs 167
  6.1 Introduction ........................................ 167
    6.1.1 Artificial Viscosity Model ........................................ 167
  6.2 Verifiability Theorem ........................................ 168
    6.2.1 Hybrid Data-Driven ROM Closure Terms ............ 168
    6.2.2 Verifiability ........................................ 170
Bibliography ........................................ 170

7 Lagrangian ROM Using Finite Time Lyapunov Exponents 172
  7.1 Introduction ........................................ 173
  7.2 Lagrangian Reduced Order Models ........................................ 175
    7.2.1 Finite Time Lyapunov Exponents (FTLE) Computation ............ 175
    7.2.2 Eulerian Reduced Order Model (E-ROM) ........................................ 176
    7.2.3 Lagrangian ROMs ........................................ 179
# List of Algorithms

<table>
<thead>
<tr>
<th></th>
<th>Algorithm</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Proper Orthogonal Decomposition</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>Galerkin ROM</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>2S-DD-VMS-ROM</td>
<td>37</td>
</tr>
<tr>
<td>4</td>
<td>3S-DD-VMS-ROM</td>
<td>39</td>
</tr>
<tr>
<td>5</td>
<td>Truncated SVD in Solving Least Square Problem</td>
<td>131</td>
</tr>
<tr>
<td>6</td>
<td>Full Order Model (FOM)</td>
<td>183</td>
</tr>
<tr>
<td>7</td>
<td>Reduced Order Model (ROM)</td>
<td>184</td>
</tr>
</tbody>
</table>
## List of Figures

1.1 Illustrative outline of this dissertation. ........................................ 12

2.1 Burgers equation, $\nu = 10^{-3}$, reconstructive regime. FOM projection, G-ROM, 2S-DD-VMS-DDC-ROM, and 3S-DD-VMS-DDC-ROM plots for $r = 7$. .... 43

2.2 Geometry of the flow past a circular cylinder numerical experiment. .... 48

2.3 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. ........................................ 51

2.4 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. ........................................ 52

2.5 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. ........................................ 53

2.6 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. ........................................ 56

2.7 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. ........................................ 57

2.8 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. ........................................ 58

2.9 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. ........................................ 61
2.10 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. .................................................. 62

2.11 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. ................................. 63

2.12 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. .................................................. 67

2.13 QGE, $Re = 450$, $Ro = 0.0036$, reconstructive regime. Time-averaged streamfunction $\bar{\psi}$ over the interval $[10, 80]$ for FOM, G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different $r$ values. .................................................. 68

2.14 Backward facing step, $Re = 1000$. Geometry and finite element mesh (top). Magnitude of FOM velocity field at $t = 125$ (bottom). ................................. 69

2.15 Backward facing step, $Re = 1000$. Time evolution of the FOM kinetic energy. .................................................. 70

2.16 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 $(19, 1)$. ................................. 72

2.17 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. .................................................. 73

2.18 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 $(19, 1)$. ................................. 74

2.19 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 $(0.43, 0.2)$ .................................................. 76

2.20 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 $r = 5$. .................................................. 77
3.1 Basis functions for the streamfunction (first row) and vorticity (second row). The vorticity basis functions \( \varphi_i \)'s are the POD modes computed based on the DNS snapshots for the vorticity, while each streamfunction basis function \( \phi_i \) is related to \( \varphi_i \) via \( \phi_i = -\Delta^{-1} \varphi_i \); see Section 3.2.2.

3.2 Kinetic energy of DNS, G-ROM, DDC-ROM and CDDC-ROM with different \( r \) values. All the ROMs are initialized at \( t = 10 \) using the projected DNS data.

3.3 Time-averaged streamfunction \( \psi \) over the interval \([10, 80]\) for DNS, 10-dim G-ROM, 10-dim DDC-ROM, and 10-dim CDDC-ROM.

4.1 Burgers equation (4.55), reconstructive regime: linear regression for \( E(L^2) \) and \( \eta(L^2) \) for fixed \( r \) values and different tolerance values in the truncated SVD.

4.2 Geometry of the flow past a circular cylinder numerical experiment.

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

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

5.1 Framework of the hybrid ROM

5.2 Geometry of the three-dimensional channel flow

5.3 Scaled eigenvalues \( \lambda_k / \lambda_1 \) for (i) the 3D channel flow with \( Re = 13, 750 \), (ii) the 2D flow past a circular cylinder with \( Re = 1000 \) and (iii) the quasi-geostrophic equation (QGE) with \( Re = 450 \) and \( Ro = 0.0036 \).

5.4 Centering trajectory \( U_0 \) and POD bases, \( \varphi_1, \varphi_{25}, \) and \( \varphi_{50} \) for the three-dimensional channel flow.

5.5 Kinetic energy over \( t \in [0, 10] \) for LES data and projection

5.6 Second-order statistics for different sampling time intervals. Left: RMS turbulent intensities; Right: The \( \mathbb{R}_{12} \) Reynolds stress components.


7.1 Streamfunction contour plots at $t = 40$ (top, left), $t = 60$ (top, middle), and time-averaged (top, right). FTLE contour plots at $t = 40$ (bottom, left), $t = 60$ (bottom, middle), and time-averaged (bottom, right). An FTLE movie is available at https://youtu.be/JXqdcBVfhMw.

7.2 Time evolution of the spatially averaged kinetic energy of the FOM.

7.3 ROM basis functions $\psi_{10}$ (first column), $\psi_{20}$ (second column), and $\psi_{30}$ (third column) for the standard E-ROM (first row), new Lagrangian $\alpha$-ROM with $\alpha = 10^4$ (second row), and new Lagrangian $\lambda$-ROM (third row).

7.4 Eulerian investigation, reconstructive regime: Mean streamfunction from E-ROM (first row), $\lambda$-ROM (second row) $\alpha$-ROM with $\alpha = 1$ (third row), and $\alpha = 10^4$ (fourth row), for $r = 10$ (first column), $r = 15$ (second column), $r = 20$ (third column), and $r = 30$ (fourth column).

7.5 Eulerian investigation, predictive regime: Mean streamfunction from E-ROM (first row), $\lambda$-ROM (second row) $\alpha$-ROM with $\alpha = 1$ (third row), and $\alpha = 10^4$ (fourth row), for $r = 10$ (first column), $r = 15$ (second column), $r = 20$ (third column), and $r = 30$ (fourth column).
## List of Tables

2.1 Burgers equation, $\nu = 10^{-3}$, reconstructive regime, optimal $tol$, $tol_L$, and $tol_S$. Average $L^2$ error for G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different r values. .................................................. 42

2.2 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. .................................................. 44

2.3 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. .................................................. 44

2.4 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. .................................................. 45

2.5 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. .................................................. 45

2.6 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. .................................................. 45

2.7 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. .................................................. 46

2.8 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. .................................................. 46
2.9 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.

2.10 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.

2.11 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.

2.12 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.

2.13 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.

2.14 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.

2.15 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.

2.16 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.

2.17 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.

2.18 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.

2.19 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.

2.20 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.

2.21 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.
3.1 The relative errors for the time-averaged streamfunction defined by (3.26).

3.2 The ROM relative errors for the time-averaged streamfunction defined in (3.26) for the two predictive test cases: the POD basis functions are generated using DNS snapshots over the time interval $[10, 45]$ for Case I and over $[10, 35]$ for Case II. The ROM simulations are carried out in the time interval $[10, 80]$.  

4.1 Burgers equation (4.55), reconstructive regime: $E(L^2)$ and $\eta(L^2)$ values for fixed $r$ values and different tolerance values in the truncated SVD.  

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

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

5.1 Acronyms for different ROMs  

5.2 Qualitative description of the different ROMs  

5.3 Artificial viscosity values in AV-G-ROM and hybrid data-driven ROMs  

7.1 The new Lagrangian ROMs ($\alpha$-ROM and $\lambda$-ROM), the standard Eulerian ROM (E-ROM), and the inner products used to construct their bases.  

7.2 Eulerian investigation, reconstructive regime: $L^2$ norm of the errors in the time-averaged streamfunction (7.21) for E-ROM (second column), $\lambda$-ROM (third column), and $\alpha$-ROM for $\alpha = 1$ (fourth column), $\alpha = 10$ (fifth column), $\alpha = 10^2$ (sixth column), $\alpha = 10^3$ (seventh column), and $\alpha = 10^4$ (eighth column).  

7.3 Eulerian investigation, predictive regime: $L^2$ norm of the errors in the time-averaged streamfunction (7.21) for E-ROM (second column), $\lambda$-ROM (third column), and $\alpha$-ROM for $\alpha = 1$ (fourth column), $\alpha = 10$ (fifth column), $\alpha = 10^2$ (sixth column), $\alpha = 10^3$ (seventh column), and $\alpha = 10^4$ (eighth column).
7.4 Lagrangian investigation, reconstructive regime: $L^2$ norm of the errors in the time-averaged FTLE (7.22) for E-ROM (second column) $\lambda$-ROM (third column), and $\alpha$-ROM for $\alpha = 1$ (fourth column), $\alpha = 10$ (fifth column), $\alpha = 10^2$ (sixth column), $\alpha = 10^3$ (seventh column), and $\alpha = 10^4$ (eighth column).

7.5 Lagrangian investigation, predictive regime: $L^2$ norm of the errors in the time-averaged FTLE (7.22) for E-ROM (second column) $\lambda$-ROM (third column), and $\alpha$-ROM for $\alpha = 1$ (fourth column), $\alpha = 10$ (fifth column), $\alpha = 10^2$ (sixth column), $\alpha = 10^3$ (seventh column), and $\alpha = 10^4$ (eighth column).

7.6 Speed-up factors (7.23) for velocity field computation: E-ROM (second column), $\lambda$-ROM (third column), and $\alpha$-ROM for $\alpha = 1$ (fourth column), $\alpha = 10$ (fifth column), $\alpha = 10^2$ (sixth column), $\alpha = 10^3$ (seventh column), and $\alpha = 10^4$ (eighth column).
Chapter 1

Introduction

1.1 Motivation

Computer simulations are now essential for many physical and engineering applications. For flow simulation and control problems, computational fluid dynamics (CFD), which is one class of numerical simulations that are derived from different disciplines of fluid dynamics, is usually used in practical applications. CFD is heavily used by engineers from power generation industry for it can reduce the development and retrofitting costs [77]. Over the last two decades, the available computational power has increased rapidly. However, it is still challenging to use the direct numerical simulation (DNS) for fluids, especially for turbulent flows which possibly require millions or even billions of degrees of freedom and days of CPU time on state-of-art parallel computer hardware architectures [64].

In some CFD applications, e.g., flow control or optimal design problems, repeated solution of the fluid flows governed by partial differential equations (PDEs) with different values of parameters are required, which hence places a much heavier burden on the available computational power. Therefore, reduced order models (ROMs) (or in some literature, model order reduction (MOR) [74]), which are surrogate modeling strategies, are used to significantly reduce the computational cost and storage capacity. From this point of view, ROMs can be defined as a class of numerical approaches which aim to replace high fidelity models (or full order models (FOMs)) with models with a much lower numerical complexity and cost [7, 25, 64, 65].

The ROMs can be placed into three different categories: (1). data-fit models, (2). projection-based order reduced models (Proj-ROMs), and (3). hierarchical models [6, 20, 60, 61]. Both the data-fit models and Proj-ROMs are derived from high fidelity models, while the hierarchical models are physics-based models with simplifying physics assumptions or numerical settings. In particular, the data-fit models involve the non-intrusive offline stage that is
based solely on access to snapshots and does not use governing equations, while the Proj-ROMs are intrusive in nature, which mandates projecting system operators onto a ROM subspace [14, 59, 88]. Although the non-intrusive process with a ‘black-box’ input and output mode can be stable and efficient in modeling nonlinear PDEs, it loses the underlying structure from FOM and is very difficult to derive rigorous system-theoretic error bounds and estimates [6].

For fluid problems, the Proj-ROMs are used in approximating the flow equations, e.g., the Navier-Stokes equations (NSE) and the quasi-geostrophic equations (QGE). The Proj-ROM framework usually comprises online and offline stages: an offline stage is performed using the FOM data in order to obtain the ROM basis functions and pre-compute the ROM operators. Then, in an online stage, these ROM operators are repeatedly used to perform online computation with different parameter settings [34, 37, 38].

### 1.2 Turbulent Flows

Turbulent flows are ubiquitous in our everyday surroundings and consist of a wide range of scales [5, 62]: from the volutes of smoke in a cigarette to atmospheric eddies of a mountain stream. Turbulent flows feature transport and mixing, which play an important role in applications. For fully homogeneous isotropic turbulent flows, the continuous energy spectrum can be divided into three categories of ranges: large production scales, inertial energy transfer scales, and small dissipative scales [2, 22, 40, 69]. Scale interactions, e.g., from a spectral point of view, stemming from the nonlinear interactions of fluid flows, contribute to the cascade of kinetic energy from large scales to small scales, which is predicted by the well-known Kolmogorov law about the kinetic energy spectrum in an inertia-dominated range of intermediate scales [9, 17, 92].

The Reynolds number, \( Re \), a non-dimensional parameter [5, 15, 19, 43, 62], can used to quantify the degree of turbulence and can be defined as:

\[
Re = \frac{UL}{\nu},
\]

where \( U \) is the characteristic velocity scale, \( L \) the characteristic length scale, and \( \nu \) the kinematic viscosity of the fluids. If the Reynolds number is small or within a moderate range,
the flow will be laminar, and displays regular and trackable spatiotemporal features. On the other hand, if the Reynolds number increases, flows will induce a sequence of instabilities and at some large enough value of $Re$, fully turbulent flows occur [62].

In turbulence theory, the Richardson’s energy cascade [62, 69] provides a qualitative picture of scale interactions: at the largest scales of turbulent flows, kinetic energy is produced and then transmitted to smaller and smaller scales, which is independent of the viscosity parameters, i.e., $\nu$ (or Reynolds number); at the smallest scales, the viscosity effects enforce the dissipation of energy. Within this cascade, most of the energy is exchanged across a certain scale-size level, and the largest and the smallest scales, on the other hand, do not have any effect on the energy transfer. This cascade theory encourages one to check or even build a model which captures the ‘correct’ energy content of turbulent flows.

In engineering, turbulent flows are widespread. For example, flows in compressors, combustion engines or pipes are all turbulent. Pope summarized three different research categories of turbulence [62]: (1). Discovery, i.e., using experimental studies for qualitative and quantitative properties, (2). Modeling, i.e., developing tractable mathematical tools for predicting, and (3). Control, i.e., manipulating the flow in a beneficial way. In this dissertation, we mainly focus on the modeling, and in particular, reduced order modeling of the turbulent flow: In particular, we mainly consider two types of simplified mathematical models for turbulent flows: (i). the incompressible Navier-Stokes equations, and (ii). the quasi-geostrophic equations.

### 1.2.1 Navier-Stokes Equations

The fundamental dynamics of incompressible flows are described by the incompressible Navier-Stokes equations (NSE):

\[
\frac{\partial \mathbf{u}}{\partial t} - Re^{-1} \Delta \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p = \mathbf{f} , \tag{1.2}
\]
\[
\nabla \cdot \mathbf{u} = 0 , \tag{1.3}
\]

where $\mathbf{u}$ is the velocity, $p$ the pressure, $\mathbf{f}$ the forcing, and $Re$ the Reynolds number. Boundary conditions and initial conditions need to be specified. In our ROM investigation, we often choose appropriate boundary conditions such that the boundary terms vanishes in the weak form.
1.2.2 Quasi-Geostrophic Equations Equations

From the incompressible NSE (1.2) and (1.3), one can derive the primitive equations under the Boussinesq approximation with a rotation framework. Then, using the asymptotic analysis under certain vertical and horizontal scale assumptions, the single-layer quasi-geostrophic equations (QGE) can be derived [58, 78]:

\[
\frac{\partial \omega}{\partial t} + J(\omega, \psi) - Ro^{-1} \frac{\partial \psi}{\partial x} = Re^{-1} \Delta \omega + Ro^{-1} F, \tag{1.4}
\]

\[
\omega = -\Delta \psi, \tag{1.5}
\]

where \( J(\omega, \psi) = \omega_x \psi_y - \omega_y \psi_x \) is the Jacobian, \( \omega \) is the vorticity, \( \psi \) is the streamfunction, \( Re \) is the Reynolds number, and \( Ro \) is the Rossby number. The QGE (1.4) and (1.5) (or the barotropic vorticity equation (BVE)), are a mathematical model for simplified forced-dissipative large scale ocean circulation [12, 48, 58, 78]. The QGE retain a semi-geostrophic balance between the pressure gradient and the Coriolis force due to rotation.

1.3 Proper Orthogonal Decomposition

The proper orthogonal decomposition (POD, also known as the Karhunen–Loéve decomposition or principal component analysis (PCA)) is a very useful tool in both understanding and low-rank surrogate modeling of non-homogeneous turbulent flows [11, 27]. POD modes in non-homogeneous turbulent flows, similar to Fourier modes in homogeneous context, are inherently hierarchical in the energy sense and can provide deep insights into the dynamics of turbulence, i.e., understanding energy transfers and quantifying forward and backward energy cascades among different scales [9, 17, 92].

In what follows, we briefly describe the POD procedure. We consider a finite number of time instances,

\[
t_1, t_2, \cdots, t_M, \tag{1.6}
\]

with \( t_i = i\Delta t, i = 1, \cdots, M \). Suppose we have the ensemble set of snapshots,

\[
Y = \{ u(\cdot, t_1), \cdots, u(\cdot, t_M) \}. \tag{1.7}
\]
The proper orthogonal decomposition (POD) method [11, 26, 27, 41, 42, 81, 82, 83], seeks a low dimensional basis \( \varphi_1, \cdots, \varphi_r \), by solving the following minimization problem in \( X \):

\[
\min \frac{1}{M} \sum_{l=1}^{M} \left\| u(\cdot, t_l) - \sum_{i=1}^{r} \left( u(\cdot, t_i), \varphi_i(\cdot) \right) X \varphi_i(\cdot) \right\|^2_X ,
\]

subject to the orthonormality of the basis, i.e.,

\[
(\varphi_i, \varphi_j) = \delta_{ij}, \quad \text{for } 1 \leq i, j \leq r,
\]

where \((\cdot, \cdot)\) denotes the inner product on \( X \), and \( \delta_{ij} \) is the Kronecker delta. To solve the minimization problem (1.8), we consider the eigenvalue problem:

\[
K \xi_j = \lambda_j \xi_j, \quad j = 1, \cdots, d,
\]

where \( K \in \mathbb{R}^{M \times M} \) with \( K_{l_1 l_2} = \frac{1}{M}(u(\cdot, t_{l_1}), u(\cdot, t_{l_2})) \), is the snapshot correlation matrix, and \((\lambda_j, \xi_j)\) is the \( j \)th pair of eigenvalue and eigenvector. We note that the eigenvalues are positive and sorted in a descending order, i.e,

\[
\lambda_1 \geq \lambda_2 \geq \cdots, \lambda_d > 0.
\]

Then, the solution to (1.8) yields

\[
\varphi_j(\cdot) = \frac{1}{\sqrt{\lambda_j}} \sum_{l=1}^{M} (\xi_j)_l u(\cdot, t_l), \quad 1 \leq j \leq d.
\]

The POD method can be summarized in Algorithm 1 [33, 34, 35, 80].

\begin{algorithm}
\caption{Proper Orthogonal Decomposition}
1: Given an ensemble set of snapshots: \( Y = \{ u(\cdot, t_1), \cdots, u(\cdot, t_M) \} \) at \( M \) different time instances.
2: Calculate the snapshot correlation matrix \( K \), where
\[
K_{l_1 l_2} = \frac{1}{M}(u(\cdot, t_{l_1}), u(\cdot, t_{l_2})), 1 \leq l_1, l_2 \leq M.
\]
3: Solve the eigenvalue problem: \( K \xi_j = \lambda_j \xi_j, 1 \leq j \leq d \), where \( d \) is rank of \( Y \).
4: Calculate the first POD \( r \) basis (\( r \leq d \)) by using the following formula
\[
\varphi_j(\cdot) = \frac{1}{\sqrt{\lambda_j}} \sum_{l=1}^{M} (\xi_j)_l u(\cdot, t_l), \quad 1 \leq j \leq r.
\]
\end{algorithm}
Remark 1.1. In the finite element method (FEM) setting, the snapshot correlation matrix can be obtained via the following relation:

$$ K = \frac{1}{M} Y^T M_h Y, $$

where $M_h \in \mathbb{R}^{N_h \times N_h}$ is FEM mass matrix, $Y \in \mathbb{R}^{N_h \times M}$. Then the eigenvalue problem (1.10) becomes,

$$ \frac{1}{M} Y^T M_h Y \xi_j = \lambda \xi_j. $$

Provided that $\varphi_j = \frac{1}{\sqrt{\lambda_j}} Y \xi_j$,

$$ \lambda_j \varphi_j = \sqrt{\lambda_j} Y \xi_j = \frac{1}{\sqrt{\lambda_j}} Y (\lambda_j \xi_j) $$

$$ = \frac{1}{\sqrt{\lambda_j}} Y \left( \frac{1}{M} Y^T M_h Y \xi_j \right) $$

$$ = \frac{1}{\sqrt{\lambda_j}} Y \left( \frac{1}{M} Y^T M_h (\sqrt{\lambda_j} \varphi_j) \right) $$

$$ = \frac{1}{M} Y Y^T M_h \varphi_j. $$

Hence, the solution of the minimization problem (1.8) is equivalent to solving the following eigenvalue problem:

$$ \frac{1}{M} Y Y^T M_h \varphi_j = \lambda_j \varphi_j, \quad j = 1, \ldots, d. $$

### 1.4 Galerkin ROM

The standard Galerkin ROM (G-ROM, or in some references [17], POD-Galerkin ROM), which in most cases uses the POD as the basis set of the reduced space (as described in Section 1.3; there are, however, many other choices of ROM basis, e.g., Fourier modes, dynamic mode decomposition (DMD)), builds a low-dimensional dynamical system in order to obtain an efficient and relatively accurate description of the flows. We sketch the standard
1.4. Galerkin ROM

G-ROM derivation in Algorithm 2 [9, 10, 34, 73, 82, 83, 87].

Algorithm 2: Galerkin ROM

1: Suppose the physical system is governed by a general nonlinear system of PDEs:
   \[ \dot{\mathbf{u}} = \mathcal{F}(\mathbf{u}), \quad (1.17) \]

2: Use available data (snapshots) for few parameter values to construct orthonormal modes \( \{\varphi_1, \ldots, \varphi_R\} \), \( R = O(10^3) \), which represent the recurrent spatial structures.

3: Choose the dominant modes \( \{\varphi_1, \ldots, \varphi_r\} \), \( r = O(10) \), as basis functions for the ROM.

4: Use a Galerkin truncation \( \mathbf{u}_r(\mathbf{x}, t) = \sum_{j=1}^r a_j(t) \varphi_j(\mathbf{x}) \).

5: Replace \( \mathbf{u} \) with \( \mathbf{u}_r \) in (1.17).

6: Use a Galerkin projection of the PDE obtained in step (4) onto the ROM space \( \mathbf{X}_r := \text{span}\{\varphi_1, \ldots, \varphi_r\} \) to obtain an \( r \)-dimensional system, which is the Galerkin ROM (G-ROM).
   \[ (\mathbf{u}_r^*, \varphi_i) = (\mathcal{F}(\mathbf{u}_r), \varphi_i), \quad i = 1, \ldots, r. \quad (1.18) \]

7: In an offline stage, compute the ROM operators.

8: In an online stage, repeatedly use the G-ROM (1.18) (for parameters different from the training parameters and/or longer time intervals).

With an intrusive nature, G-ROM approximates the coefficients of modes from the projected ordinary differential equations with the help of a Galerkin projection method and thus relies on the governing equations. In what follows, we present the forms of G-ROM for the two different fluid models we are interested in.

1.4.1 G-ROM for NSE

The incompressible NSE (1.2)–(1.3) can be cast in the general form (1.17) by choosing \( \mathcal{F} = Re^{-1}\Delta \mathbf{u} - \mathbf{u} \cdot \nabla \mathbf{u} + \mathbf{f} \) and \( \mathbf{X} \) the space of weakly divergence-free functions in \( H_0^1 \):

\[ \left( \frac{\partial \mathbf{u}}{\partial t}, \varphi_i \right) + Re^{-1}(\nabla \mathbf{u}, \nabla \varphi_i) + (\mathbf{u} \cdot \nabla \mathbf{u}, \varphi_i) = (\mathbf{f}, \varphi_i), \quad i = 1, \ldots, d, \quad (1.19) \]
Then, the G-ROM reads

\[
\dot{a} = b + A a + a^\top B a,
\]

where \( a(t) \) is the vector of unknown coefficients \( a_j(t), 1 \leq j \leq r \), \( b \) is a constant vector with entries

\[
b_i = (f, \varphi_i),
\]

\( A \) is an \( r \times r \) matrix with entries

\[
A_{im} = -Re^{-1} (\nabla \varphi_m, \nabla \varphi_i),
\]

and \( B \) is an \( r \times r \times r \) tensor with entries

\[
B_{imn} = -(\varphi_m \cdot \nabla \varphi_n, \varphi_i), \quad 1 \leq i, m, n \leq r.
\]

The G-ROM (1.20) does not include a pressure approximation \([44, 55, 76, 83]\), 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., \([18, 25, 63]\). Once the matrix \( A \) and tensor \( B \) are assembled in the offline stage, the G-ROM (1.20) 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 \([27, 32, 54, 56, 83, 89, 91]\).

### 1.4.2 G-ROM for QGE

To determine the POD streamfunction basis functions, we use the POD vorticity basis functions and follow the approach in \([52, 53, 72]\). Specifically, we define the POD streamfunction basis functions as the normalized functions \( \{ \phi_j \}_{j=1}^r \), which are chosen such that the satisfy the following Poisson problem with homogeneous Dirichlet boundary conditions:

\[
-\Delta \phi_j = \varphi_j, \quad j = 1, \ldots, r.
\]
Next, we define the ROM approximations of the vorticity and streamfunction as follows:

\[
\omega_r(x,t) = \sum_{j=1}^{r} a_j(t) \varphi_j(x),
\]

(1.25)

\[
\psi_r(x,t) = \sum_{j=1}^{r} a_j(t) \phi_j(x),
\]

(1.26)

where \(\{a_j(t)\}_{j=1}^{r}\) are the sought time-varying ROM coefficients. We note that we made two important choices in our approach: (i) We enforced the coupling between the POD vorticity and streamfunction basis functions in (1.24); and (ii) We used the same ROM coefficients in the ROM vorticity approximation (1.25) and in the ROM streamfunction approximation (1.26). The motivation for making these two choices is efficiency. Indeed, we only need to construct a ROM for the vorticity; once the coefficients \(a_j\) are determined from (1.4), equation (1.4) is automatically satisfied. (Of course, one could use a different approach and construct two different ROM bases and two different ROM approximations for the vorticity and streamfunction, but that would increase the ROM computational cost.) To construct a ROM for the vorticity, we replace the vorticity \(\omega\) by \(\omega_r\) in the QGE (1.4), and then we use a Galerkin projection onto \(X^r\). Thus, we obtain the *Galerkin ROM (G-ROM)* for the QGE: \(\forall i = 1, \ldots, r,\)

\[
\begin{aligned}
\left( \frac{\partial \omega_r}{\partial t}, \varphi_i \right) + (J(\omega_r, \psi_r), \varphi_i) - Ro^{-1} \left( \frac{\partial \psi_r}{\partial x}, \varphi_i \right) + Re^{-1} (\nabla \omega_r, \nabla \varphi_i) &= Ro^{-1} (F_e, \varphi_i) .
\end{aligned}
\]

(1.27)

The G-ROM (1.27) yields the following autonomous dynamical system for the vector of time coefficients, \(\mathbf{a}(t) = (a_i(t))_{i=1,\ldots,r}\):

\[
\dot{\mathbf{a}} = \mathbf{b} + \mathbf{A} \mathbf{a} + \mathbf{a}^\top \mathbf{B} \mathbf{a},
\]

(1.28)

where \(\mathbf{b}, \mathbf{A},\) and \(\mathbf{B}\) are an \(r \times 1\) vector, an \(r \times r\) matrix, and an \(r \times r \times r\) tensor, which correspond to the constant, linear, and quadratic terms in the numerical discretization of the QGE (1.4), respectively. The \(r\)-dimensional system (1.28) can be written componentwise as follows: For all \(i = 1, \ldots, r,\)

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

(1.29)
where

\[
\begin{align*}
b_i &= \text{Ro}^{-1} \left( F_e, \varphi_i \right), \quad (1.30) \\
A_{im} &= \text{Ro}^{-1} \left( \frac{\partial \phi_m}{\partial x}, \varphi_i \right) - \text{Re}^{-1} \left( \nabla \varphi_m, \nabla \varphi_i \right), \quad (1.31) \\
B_{imn} &= -\left( J(\varphi_m, \phi_n), \varphi_i \right). \quad (1.32)
\end{align*}
\]

### 1.5 Variational Multiscale Closure Modeling

In the *large eddy simulation* (LES) context, the inaccuracy caused by the cutoff of subgrid scales is remedied by the so-called *closure modeling*. The closure modeling introduces one extra closure term to the original equations and aims to model the nonlinear interactions between modeled and discarded scales. The closure term which arises from the nonlinear term of the original PDEs (e.g., the convection term in the NSE) should be in compliance of both the physical and numerical constraints [70].

The standard Galerkin ROM described in Section 1.4 is relatively accurate for simple test problems. However, for realistic flows, the premise of low-dimensional G-ROM inevitably requires the drop-out of low-energetic modes in the Galerkin projection procedure and thus omits the scale interactions and energy transfers among retained and neglected modes. This neglect could result in G-ROM’s inaccuracy for turbulent flows. Thus, adding such ‘missing’ interactions, i.e., the closure term, to the G-ROM is a promising strategy to improve ROM accuracy. In recent research, this strategy has been shown to be effective [3, 4, 8, 13, 21, 23, 24, 27, 32, 33, 35, 45, 46, 47, 57, 68, 75, 79, 80, 81, 82, 83, 84, 85, 86, 87, 89, 91].

In LES, closure schemes can be categorized into two: (1). functional approach, and (2). structural approach. ROM closure modeling can also be divided into these two categories. Functional ROM closure modeling usually models the action of the discarded modes on the resolved modes, generically by dissipating energy. For example, the Smagorinsky type ROM [67, 71, 83] employs the dissipation assumption and assumes an eddy viscosity formulation. Structural ROM closure modeling, on the other hand, tends to approximate the closure term by a formal evaluation or by applying a series expansion (see, e.g., the AD-ROM that applies the approximate deconvolution (AD) method to approximate the closure term). The data-driven (DD) ROM closure modeling (see, e.g., [49, 52, 89]) can be categorized as a
third approach, which attempts to use data to construct the explicit functions or operators for the closure term.

The variational multiscale (VMS) framework, which was proposed by Hughes et al [28, 29, 30, 31], has been widely applied for general problems of computational mechanics and fluid simulation (see, e.g., [1, 16, 30, 36, 66] for surveys). The VMS methodology divides the underlying problem into different groups of scales and then applies different numerical treatments for each group. This separation of scales allows one to consider different effects imposed by the unresolved scales on different resolved scales. For example, if one uses a three-scale separation for turbulent simulation (i.e., the large resolved scale, small resolved scale, and unresolved scale), then following the turbulence energy cascade concept, one may consider in the VMS scheme that the large resolved scales and unresolved scales are of minor relevance, while the small resolved scales can be assumed to be dissipative.

Using the VMS framework in data-driven modeling allows ROMs to incorporate physical laws in the ROM closure modeling, e.g., the Kolmogorov hypothesis. In this dissertation, our main goal is to extend the data-driven closure ROM proposed in [89] to a data-driven variational multiscale (DD-VMS) framework. Also, we aim to include physical modeling in the DD-VMS framework, e.g., by adding physical constraints to the data-driven approach. We will show that this is indispensable in challenging test problems, e.g., turbulent channel flows in two or three dimensions.

1.6 Outline of Dissertation

In this dissertation, we develop the data-driven variational multiscale reduced order model (DD-VMS-ROM) framework, which improves the accuracy of the standard G-ROM. Within the DD-VMS-ROM framework, we develop the hybrid data-driven ROM, which can further improve the DD-VMS-ROM’s performance in the numerical simulation of turbulent flows.

In addition, we develop a Lagrangian type of ROMs, which incorporates Lagrangian data, i.e., the finite time Lyapunov exponent (FTLE) field, when constructing the ROM basis.

In Figure 1.1, the main ROMs that are investigated in this dissertation, as well as their connections, are illustrated.
The rest of the dissertation is organized as follows:

1. In Chapter 2, the data-driven variational multiscale reduced order model (DD-VMS-ROM) is proposed and developed. In particular, two different DD-VMS-ROMs, i.e., two scale and three scale DD-VMS-ROMs, are investigated. To show the accuracy of the proposed ROMs, four tests are performed: (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 material in this chapter is based on the following report:
1.6. OUTLINE OF DISSERTATION


2. In Chapter 3, the physically constrained DD-VMS-ROM is investigated for the test problem of QGE (1.4) and (1.5).

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


3. In Chapter 4, the verifiability of the DD-VMS-ROM is investigated.

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


4. In Chapter 5, a hybrid data-driven ROM is proposed and investigated for the test problem of *three dimensional* turbulent channel flow with Reynolds number $Re = 13,750$.

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


5. In Chapter 7, the Lagrangian ROM, which uses FTLE data in the offline stage, is proposed and investigated. Moreover, the accuracy and efficiency are investigated in generating FTLE using ROM data. The numerical test is done for the QGE test problem (1.4) and (1.5).

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

Bibliography


Chapter 2

Data-Driven Variational Multiscale Reduced Order Models


The author of this dissertation contributes to the model’s conceptual development and numerical experiments for 2D flow past a cylinder in Section 2.3.3; the quasi-geostrophic equation in Section 2.3.4, and 2D flow over a backward facing step in Section 2.3.5.

**ABSTRACT OF CHAPTER 2**

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...
2.1. Introduction

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.

2.1 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:

$$\dot{a} = F(a),$$  \hspace{1cm} (2.1)

where $a$ is the vector of coefficients in the ROM approximation $\sum_{i=1}^{r} a_i(t)\varphi_i(x)$ of the variable of interest and $F$ 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 (2.1) is then used in a regime that is different from the training regime. Since the ROM (2.1) 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 \textit{ROM closure problem} [4, 6, 7, 12, 19, 26, 30, 44, 45, 46, 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\}:

\[
\dot{a} = F(a) + \text{Closure}(a),
\]

where \text{Closure}(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 \textit{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 \textit{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 \textit{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) \textit{resolved large} scales, (ii) \textit{resolved small} scales, and (iii) \textit{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, 43, 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) [48, 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 [49].)

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, 42]. 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, 44, 54, 58, 85].

The rest of the paper is organized as follows: In Section 2.2, we introduce the new DD-VMS-ROM. In Section 2.3, 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 2.4, we draw conclusions and outline future research directions.

### 2.2 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 2.2.1 we briefly sketch the VMS methodology for general numerical discretizations (see, e.g., [3, 39] for more details), and in Section 2.2.2 we outline the standard Galerkin ROMs.

We construct the new DD-VMS-ROM in two stages: In Section 2.2.3, 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 [49, 52] under the name “data-driven correction ROM.” However, we decided to outline
2.2. DATA-DRIVEN VARIATIONAL MULTISCALE REDUCED ORDER MODELS (DD-VMS-ROMs)

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 2.2.4, 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.

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

\[ \dot{u} = f(u), \quad (2.3) \]

whose weak (variational) form is

\[ (\dot{u}, v) = (f(u), v), \quad \forall v \in X, \quad (2.4) \]

where \( f \) is a general nonlinear function and \( X \) is an appropriate infinite dimensional space. To build the VMS framework, we start with a sequence of hierarchical spaces of increasing resolutions: \( X_1, X_1 \oplus X_2, X_1 \oplus X_2 \oplus X_3, \ldots \). Next, we project system (2.3) onto each of the spaces \( X_1, X_2, X_3, \ldots \), which yields a separate equation for each space. The goal is, of course, to solve for the \( u \) component that lives in the coarsest space (i.e., \( X_1 \)), since this yields the lowest-dimensional system:

\[ (\dot{u}, v_1) = (f(u), v_1) \quad \forall v_1 \in X_1, \quad (2.5) \]

System (2.5), however, is not closed, since its right-hand side

\[ (f(u), v_1) = (f(u_1 + u_2 + u_3 + \ldots), v_1) \quad \forall v_1 \in X_1, \quad (2.6) \]
involves \( u \) components that do not live in \( X_1 \) (i.e., \( u_2 \in X_2, u_3 \in X_3, \ldots \)). This coupling is mainly due to the nonlinearity of \( f \). Thus, the VMS closure problem needs to be solved, i.e., (2.6) needs to be approximated in \( X_1 \). The VMS (2.5) equipped with an appropriate closure model yields an accurate approximation of the large scale \( X_1 \) component of \( u \).

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.

\subsection*{2.2.2 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 \( \{\varphi_1, \ldots, \varphi_R\}, R = O(10^3) \), which represent the recurrent spatial structures; (ii) Choose the dominant modes \( \{\varphi_1, \ldots, \varphi_r\}, r = O(10) \), as basis functions for the ROM; (iii) Use a Galerkin truncation \( u_r(x, t) = \sum_{j=1}^r a_j(t) \varphi_j(x) \); (iv) Replace \( u \) with \( u_r \) in (2.3); (v) Use a Galerkin projection of the PDE obtained in step (iv) onto the ROM space \( X^r := \text{span}\{\varphi_1, \ldots, \varphi_r\} \) to obtain an \( r \)-dimensional system, which is the Galerkin ROM (G-ROM):

\begin{equation}
(u_r, \varphi_i) = (f(u_r), \varphi_i), \quad i = 1, \ldots, r;
\end{equation}

(vi) In an offline stage, compute the ROM operators; (vii) In an online stage, repeatedly use the G-ROM (2.7) (for parameters different from the training parameters and/or longer time intervals).
We illustrate the G-ROM for the Navier-Stokes equations (NSE):
\[
\begin{align*}
\frac{\partial \mathbf{u}}{\partial t} - \text{Re}^{-1} \Delta \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p &= 0, \\
\nabla \cdot \mathbf{u} &= 0,
\end{align*}
\]
where \( \mathbf{u} \) is the velocity, \( p \) the pressure, and \( \text{Re} \) the Reynolds number. For clarity of presentation, we use homogeneous Dirichlet boundary conditions. The NSE \((2.8)-(2.9)\) can be cast in the general form \((2.3)\) by choosing \( f = \text{Re}^{-1} \Delta \mathbf{u} - \mathbf{u} \cdot \nabla \mathbf{u} \) and \( X \) the space of weakly divergence-free functions in \( H^1_0 \). For the NSE, the G-ROM reads
\[
\dot{\mathbf{a}} = A \mathbf{a} + \mathbf{a}^\top B \mathbf{a}, \tag{2.10}
\]
where \( \mathbf{a}(t) \) is the vector of unknown coefficients \( a_j(t), 1 \leq j \leq r \), \( A \) is an \( r \times r \) matrix with entries \( A_{im} = -\text{Re}^{-1} (\nabla \varphi_m, \nabla \varphi_i) \), and \( B \) is an \( r \times r \times r \) tensor with entries \( B_{imn} = - (\varphi_m \cdot \nabla \varphi_n, \varphi_i) \), \( 1 \leq i, m, n \leq r \). The G-ROM \((2.10)\) 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 \((2.10)\) 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 \([30, 53, 56, 84]\).

### 2.2.3 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, \( X_1 \) and \( X_2 \). Since the ROM basis is orthonormal by construction, we can build the two orthogonal spaces in a natural way: \( X_1 := \text{span}\{\varphi_1, \ldots, \varphi_r\} \), which represents the resolved ROM scales, and \( X_2 := \text{span}\{\varphi_{r+1}, \ldots, \varphi_R\} \), 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 \( u \) in the space \( X_1 \oplus X_2 \), i.e., \( u_R \in X_1 \oplus X_2 \) defined as

\[
u_R = \sum_{j=1}^{R} a_j \varphi_j = \sum_{j=1}^{r} a_j \varphi_j + \sum_{j=r+1}^{R} a_j \varphi_j = u_r + u',
\]

(2.11)

where \( u_r \in X_1 \) represents the resolved ROM component of \( u \), and \( u' \in X_2 \) represents the unresolved ROM component of \( u \). Plugging \( u_R \) in (2.3), projecting the resulting equation onto \( X_1 \), and using the ROM basis orthogonality to show that \( (\dot{u}_r, \varphi_i) = (\dot{u}_r, \varphi_i) \), \( \forall i = 1, \ldots, r \), we obtain

\[
(\dot{u}_r, \varphi_i) = (f(u_r), \varphi_i) + \left[ (f(u_R), \varphi_i) - (f(u_r), \varphi_i) \right], \ \forall i = 1, \ldots, r.
\]

(2.12)

The boxed term in (2.12) is the VMS-ROM closure term, which models the interaction between the ROM modes \( \{\varphi_1, \ldots, \varphi_r\} \) and the discarded ROM modes \( \{\varphi_{r+1}, \ldots, \varphi_R\} \). The VMS-ROM closure term is essential for the accuracy of (2.12): If we drop the VMS-ROM closure term, we are left with the G-ROM (2.7), which yields inaccurate results in the under-resolved regime. The VMS-ROM closure term is a correction term that ensures an accurate approximation of \( u_r \in X_1 \) in the higher-dimensional space \( X_1 \oplus X_2 \).

Next, we approximate the VMS-ROM closure term with \( g(u_r) \), where \( g \) is a generic function whose coefficients/parameters still need to be determined:

\[
\text{VMS-ROM closure term} = \left[ (f(u_R), \varphi_i) - (f(u_r), \varphi_i) \right] \approx (g(u_r), \varphi_i).
\]

(2.13)

To determine the coefficients/parameters in \( g \) used in (2.13), in the offline stage, we solve the following low-dimensional least squares problem:

\[
\min_{g \text{ parameters}} \sum_{j=1}^{M} \left\| \left[ (f(u_R^{FOM}(t_j)), \varphi_i) - (f(u_r^{FOM}(t_j)), \varphi_i) \right] - (g(u_r^{FOM}(t_j)), \varphi_i) \right\|^2,
\]

(2.14)

where \( u_R^{FOM} \) and \( u_r^{FOM} \) are obtained from the FOM data and \( M \) is the number of snapshots. Once \( g \) is determined, the model (2.12) with the VMS-ROM closure term replaced by \( g \) yields
2.2. DATA-DRIVEN VARIATIONAL MULTISCALE REDUCED ORDER MODELS (DD-VMS-ROMs)

the two-scale data-driven VMS-ROM (2S-DD-VMS-ROM):

\[
(u_r, \phi_i) = (f(u_r), \phi_i) + (g(u_r), \phi_i), \quad i = 1, \ldots, r.
\]  

(2.15)

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 (2.13) becomes: \( \forall i = 1, \ldots, r, \)

VMS-ROM closure term \[= -[((u_R \cdot \nabla) u_R, \phi_i) - ((u_r \cdot \nabla) u_r, \phi_i)] \]

\[\approx (g(u_r), \phi_i) \]

\[= (\tilde{A} a + a^\top \tilde{B} a)_i, \quad (2.16)\]

where, for computational efficiency, we assume that the structures of \( g \) and \( f \) are similar. Thus, in the least squares problem (2.14), we solve for all the entries in the \( r \times r \) matrix \( \tilde{A} \) and the \( r \times r \times r \) tensor \( \tilde{B} \):

\[
\min_{A,B} \sum_{j=1}^{M} \left\| -\left[ ((u_R^{FOM}(t_j) \cdot \nabla) u_R^{FOM}(t_j), \phi_i) - ((u_r^{FOM}(t_j) \cdot \nabla) u_r^{FOM}(t_j), \phi_i) \right] - (\tilde{A} a^{FOM}(t_j) + a^{FOM}(t_j)^\top \tilde{B} a^{FOM}(t_j)) \right\|^2,
\]  

(2.17)

where \( u_R^{FOM}, u_r^{FOM}, \) and \( a^{FOM} \) are obtained from the available FOM data. Specifically, the values \( a^{FOM}(t_j) \), computed at snapshot time instances \( t_j, j = 1, \ldots, M \), are obtained by projecting the corresponding snapshots \( u(t_j) \) onto the ROM basis functions \( \phi_i \) and using the orthogonality of the ROM basis functions: \( \forall i = 1, \ldots, R, \forall j = 1, \ldots, M, \)

\[a_i^{FOM}(t_j) = (u(t_j), \phi_i). \]  

(2.18)

In addition,

\[
u_R^{FOM}(t_j) = \sum_{k=1}^{R} a_k^{FOM}(t_j) \phi_k, \quad u_r^{FOM}(t_j) = \sum_{k=1}^{r} a_k^{FOM}(t_j) \phi_k.
\]  

(2.19)
The least squares problem (2.17) 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, (2.17) can be efficiently solved in the offline stage. For the NSE, the 2S-DD-VMS-ROM (2.15) takes the form

\[
\dot{a} = (A + \tilde{A})a + a^\top (B + \tilde{B})a,
\]

where \( A \) and \( B \) are the G-ROM operators in (2.10), and \( \tilde{A} \) and \( \tilde{B} \) are the VMS-ROM closure operators constructed in (2.17).

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

The 2S-DD-VMS-ROM (2.15) is based on the two-scale decomposition of \( u_R \in X_1 \oplus X_2 \) into resolved and unresolved scales: \( u_R = u_r + u' \). The flexibility of the hierarchical structure of the ROM space allows a three-scale decomposition of \( u_R \), which yields a *three-scale data-driven VMS-ROM (3S-DD-VMS-ROM)* that is more accurate than the 2S-DD-VMS-ROM (2.15). To construct the new 3S-DD-VMS-ROM, we first build three orthogonal spaces, \( X_1, X_2, \) and \( X_3 \): \( X_1 := \text{span}\{\phi_1, \ldots, \phi_{r_1}\} \), which represents the *large resolved* ROM scales, \( X_2 := \text{span}\{\phi_{r_1+1}, \ldots, \phi_r\} \), which represents the *small resolved* ROM scales, and \( X_3 := \text{span}\{\phi_{r+1}, \ldots, \phi_R\} \), which represents the *unresolved* ROM scales. Next, we consider the best ROM approximation of \( u \) in the space \( X_1 \oplus X_2 \oplus X_3 \), i.e., \( u_R \in X_1 \oplus X_2 \oplus X_3 \) defined as

\[
u_R = \sum_{j=1}^{R} a_j \phi_j
\]

\[
= \sum_{j=1}^{r_1} a_j \phi_j + \sum_{j=r_1+1}^{r} a_j \phi_j + \sum_{j=r+1}^{R} a_j \phi_j
\]

\[
= u_L + u_S + u',
\]

where \( u_L \in X_1 \) represents the large resolved ROM component of \( u_R \), \( u_S \in X_2 \) represents the small resolved ROM component of \( u_R \), and \( u' \in X_3 \) represents the unresolved ROM component of \( u_R \). Thus, with the notation from Section 2.2.3, \( u_r = u_L + u_S \). We plug \( u_R \)
2.2. DATA-DRIVEN VARIATIONAL MULTISCALE REDUCED ORDER MODELS (DD-VMS-ROMs)

In (2.3), and project the resulting equation onto both $X_1$ and $X_2$:

\[
\begin{align*}
(u_L, \varphi_i) &= (f(u_L + u_S), \varphi_i) + \left[ (f(u_R) - f(u_L + u_S)) \varphi_i \right], \\
& \quad \forall i = 1, \ldots, r_1, \\
\end{align*}
\]

\[
\begin{align*}
(u_S, \varphi_i) &= (f(u_L + u_S), \varphi_i) + \left[ (f(u_R) - f(u_L + u_S)) \varphi_i \right], \\
& \quad \forall i = r_1 + 1, \ldots, r .
\end{align*}
\]

The two boxed terms in (2.22)–(2.23) are the VMS-ROM closure terms, which have fundamentally different roles: The VMS-ROM closure term in (2.22) models the interaction between the large resolved ROM modes and the small resolved ROM modes; the VMS-ROM closure term in (2.23) 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:

\[
\begin{align*}
(\tau_L)_i &:= -[(u_L \cdot \nabla) u_R, \varphi_i] - [(u_L + u_S) \cdot \nabla) (u_L + u_S), \varphi_i] \\
&= (\tilde{A}_L a + a^\top \tilde{B}_L a)_{i}, \quad \forall i = 1, \ldots, r_1 , \\
(\tau_S)_i &:= -[(u_R \cdot \nabla) u_R, \varphi_i] - [(u_L + u_S) \cdot \nabla) (u_L + u_S), \varphi_i] \\
&= (\tilde{A}_S a + a^\top \tilde{B}_S a)_{i} \quad \forall i = r_1 + 1, \ldots, r,
\end{align*}
\]

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:

\[
\begin{align*}
\min_{\tilde{A}_L, \tilde{B}_L} \sum_{j=1}^{M} \left\| \tau_{L}^{FOM} - (\tilde{A}_L a^{FOM}(t_j) + a^{FOM}(t_j)^\top \tilde{B}_L a^{FOM}(t_j)) \right\|^2 , \\
\min_{\tilde{A}_S, \tilde{B}_S} \sum_{j=1}^{M} \left\| \tau_{S}^{FOM} - (\tilde{A}_S a^{FOM}(t_j) + a^{FOM}(t_j)^\top \tilde{B}_S a^{FOM}(t_j)) \right\|^2 ,
\end{align*}
\]

where $\tau_{L}^{FOM}$, $\tau_{S}^{FOM}$, and $a^{FOM}$ are obtained from the available FOM data.
For the NSE, the *three-scale data-driven VMS-ROM (3S-DD-VMS-ROM)* is

\[
\begin{bmatrix}
\dot{a}_L \\
\dot{a}_S
\end{bmatrix} = A a + a^\top B a + \begin{bmatrix}
\tilde{A}_L a + a^\top \tilde{B}_L a \\
\tilde{A}_S a + a^\top \tilde{B}_S a
\end{bmatrix},
\]

(2.28)

where \(a^\top = [a_L, a_S]^\top\), \(A\) and \(B\) are the G-ROM operators in (2.10), and \(\tilde{A}_L, \tilde{A}_S, \tilde{B}_L,\) and \(\tilde{B}_S\) are the VMS-ROM closure operators constructed in (2.26)–(2.27). 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 (2.26)–(2.27).

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 (2.28) is *more accurate* than the 2S-DD-VMS-ROM (2.20).

### 2.3 Numerical Results

In this section, we perform a numerical investigation of the new DD-VMS-ROM framework.

As noted in Section 2.2, the 2S-DD-VMS-ROM (2.20) was investigated in [85] under the name “data-driven filtered ROM” and in [49, 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 (2.28). 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 2.3.2); (ii) a 2D flow past a circular cylinder at Reynolds numbers \(Re = 100, Re = 500,\) and \(Re = 1000\) (Section 2.3.3); (iii) the quasi-geostrophic equations at Reynolds number \(Re = 450\) and Rossby number \(Ro = 0.0036\) (Section 2.3.4); and (iv) a 2D flow over a backward facing step at Reynolds number \(Re = 1000\) (Section 2.3.5). For each test case, we
2.3. NUMERICAL RESULTS

investigate three ROMs: the 2S-DD-VMS-ROM (2.20), the new 3S-DD-VMS-ROM (2.28), and (for comparison purposes) the standard G-ROM (2.10). 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 $[49, 85]$, 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 $[49, 85]$.

(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 $[49, 85]$.
2.3.1 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 (2.17) as the least squares problem

$$\min_{x \in \mathcal{R}((r^2+r^3) \times 1)} \| f - E x \|^2,$$  \hspace{1cm} (2.29)

where $x \in \mathcal{R}((r^2+r^3) \times 1)$ contains all the entries of $\tilde{A}$ and $\tilde{B}$, and the vector $f \in \mathcal{R}((Mr) \times 1)$ and matrix $E \in \mathcal{R}((Mr) \times (r^2+r^3))$ are computed from $u_{R}^{FOM}$, $u_{r}^{FOM}$, and $a^{FOM}$ (see (4.8) in [85]). The optimal $\tilde{A}$ and $\tilde{B}$ (i.e., the entries in $x$ that solves the linear least squares problem (2.29)) are used to build the 2S-DD-VMS-ROM (2.20).

Furthermore, as explained on page B843 of [85], the least squares problem (2.29) 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) [49, 85].

The algorithm for the 2S-DD-VMS-ROM (2.20) is presented in Algorithm 3. In most of our numerical experiments, we choose the optimal tolerance $tol$ in the truncated SVD step of Algorithm 3. Specifically, for each value $1 \leq m \leq R$ (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 (2.33), and choose the $\tilde{m}$ value yielding the lowest $L^2$ error. The only exception is in some of the numerical experiments for the Burgers equation (Section 2.3.2), where we fix $tol = tol_L$ or $tol = tol_S$ (see Tables 2.2–2.8).
2.3. Numerical Results

Algorithm 3: 2S-DD-VMS-ROM

1: Use all the entries of $\tilde{A}$ and $\tilde{B}$ in (2.20) to define vector of unknowns, $\mathbf{x}$.
2: Use $\mathbf{u}_{R}^{FOM}$, $\mathbf{u}_{r}^{FOM}$, and $\mathbf{a}_{FOM}$ to assemble the vector $\mathbf{f}$ and matrix $E$ in (2.29).
3: Use the truncated SVD algorithm to solve the linear least squares problem (2.29).

(i) Calculate the SVD of $E$:

$$E = U \Sigma V^\top,$$  \hspace{1cm} (2.30)

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

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

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

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

$$\tilde{E}^m = \tilde{U}^m \tilde{\Sigma}^m (\tilde{V}^m)^\top,$$  \hspace{1cm} (2.31)

where $\tilde{U}^m$ and $\tilde{V}^m$ are the entries of $U$ and $V$ in (2.30) that correspond to $\tilde{\Sigma}^m$.

(v) The solution of the least squares problem (2.29) is

$$\mathbf{x} = \left( \tilde{V}^m (\tilde{\Sigma}^m)^{-1} (\tilde{U}^m)^\top \right) \mathbf{f}.$$  \hspace{1cm} (2.32)

4: The 2S-DD-VMS-ROM (2.20) has the following form:

$$\dot{\mathbf{a}} = \left( \mathbf{A} + \tilde{\mathbf{A}}^m \right) \mathbf{a} + \mathbf{a}^\top \left( \mathbf{B} + \tilde{\mathbf{B}}^m \right) \mathbf{a},$$  \hspace{1cm} (2.33)

where $\tilde{\mathbf{A}}^m$ and $\tilde{\mathbf{B}}^m$ are the appropriate entries of $\mathbf{x}$ found in (2.32) with $tol = \sigma_m$.

5: Integrate the resulting 2S-DD-VMS-ROM in (2.20) over the given time domain and calculate the average $L^2$ error $E_m(L^2)$ by using formula (2.39). The optimal $\tilde{m}$ value (the optimal operators $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{B}}$) is found by solving the following minimization problem:

$$E_m(L^2) = \min_{1 \leq m \leq M} E_m(L^2).$$  \hspace{1cm} (2.34)
The algorithm for the 3S-DD-VMS-ROM (2.28) is the same as Algorithm 3, 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 (2.26) and (2.27) as the least squares problems

\[
\min_{x_L \in \mathbb{R}^{[r_1(r+r^2)] \times 1}} \| f_L - E_L x_L \|^2, \tag{2.35}
\]

\[
\min_{x_S \in \mathbb{R}^{[r-r_1(r+r^2)] \times 1}} \| f_S - E_S x_S \|^2, \tag{2.36}
\]

where \( x_L \in \mathbb{R}^{[r_1(r+r^2)] \times 1} \) contains all the entries of the operators \( \tilde{A}_L \) and \( \tilde{B}_L \), \( x_S \in \mathbb{R}^{[r-r_1(r+r^2)] \times 1} \) contains all the entries of the operators \( \tilde{A}_S \) and \( \tilde{B}_S \), and the vectors \( f_L \in \mathbb{R}^{(M \cdot r_1) \times 1} \), \( f_S \in \mathbb{R}^{(M \cdot (r-r_1)) \times 1} \) and the matrices \( E_L \in \mathbb{R}^{(M \cdot r_1) \times (r_1(r+r^2))} \), \( E_S \in \mathbb{R}^{(M \cdot (r-r_1)) \times ((r-r_1)(r+r^2))} \) are computed from \( u_{rFOM}^{FOM}, u_r^{FOM}, \) and \( a^{FOM} \) (see (4.8) in [85]). The optimal \( \tilde{A}_L, \tilde{B}_L \) and \( \tilde{A}_S, \tilde{B}_S \) (i.e., the entries in \( x_L \) and \( x_S \) that solve the linear least squares problems (2.35) and (2.36)) are used to build the 3S-DD-VMS-ROM (2.28). Again, to address the ill-conditioning of the least squares problems (2.35)–(2.36), we use the truncated SVD algorithm. The algorithm for the 3S-DD-VMS-ROM (2.28) is presented in Algorithm 4. 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 4: 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 (2.28) to define vectors of unknowns, $x_L$ and $x_S$, respectively.

2: Use $u^{FOM}_R$, $u^{FOM}_L$, and $a^{FOM}$ to assemble the vectors $f_L$ and $f_S$, and the matrices $E_L$ and $E_S$ in (2.35) and (2.36).

3: Use the truncated SVD algorithm to solve the linear least squares problems (2.35) and (2.36).

   (i) Calculate the SVD of $E_L$ and $E_S$:

\[
E_L = U_L \Sigma_L V_L^\top, \quad E_S = U_S \Sigma_S V_S^\top.
\] (2.37)

   (ii) Specify tolerances $tol_L = \sigma_{L,i}$, $i = 1, \cdots, M_L$, and $tol_S = \sigma_{S,j}$, $j = 1, \cdots, M_S$, where $M_L$ is the rank of $\Sigma_L (E_L)$ and $M_S$ is the rank of $\Sigma_S (E_S)$.

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

   (iv) Construct $\hat{E}_L^{m_L}$ and $\hat{E}_S^{m_S}$ 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 (2.28) 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 (2.39).

4: Find the optimal $\tilde{r}_1$, $\tilde{m}_L$ and $\tilde{m}_S$ 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:

\[
\mathcal{E}_{\tilde{r}_1, \tilde{m}_L, \tilde{m}_S}(L^2) = \min_{\begin{array}{c} 1 \leq r_1 < r \\ 1 \leq m_L \leq M_L \\ 1 \leq m_S \leq M_S \end{array}} \mathcal{E}_{r_1,m_L,m_S}(L^2).
\] (2.38)

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

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

whereas in the quasi-geostrophic equations test case we use the error metric (2.48). 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.

### 2.3.2 Burgers Equation

In this section, we investigate the 2S-DD-VMS-ROM (2.20) and the new 3S-DD-VMS-ROM (2.28) in the numerical simulation of the one-dimensional viscous Burgers equation:

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

with the initial condition

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

and $\nu = 10^{-3}$. This test problem has been used in [1, 36, 41, 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 $\Delta t = 10^{-3}$. 
2.3. Numerical Results

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 (2.33), we use Algorithm 3. To implement the 3S-DD-VMS-ROM (2.28), we use Algorithm 4. 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 3 and optimal \( tol_L \) and \( tol_s \) in Algorithm 4. 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 (2.40) with \( \nu = 10^{-3} \) in the reconstructive, cross-validation, and predictive regimes. In all the tables, we list the average \( L^2 \) error (2.39) 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 2.1, we list the ROMs errors for the reconstructive regime with optimal \( tol \) in Algorithm 3 and optimal \( tol_L \) and \( tol_s \) in Algorithm 4. 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.

<table>
<thead>
<tr>
<th>$r$</th>
<th>G-ROM $\mathcal{E}(L^2)$</th>
<th>tol</th>
<th>2S-DD-VMS-ROM $\mathcal{E}(L^2)$</th>
<th>tol</th>
<th>3S-DD-VMS-ROM $\mathcal{E}(L^2)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1.181e-01</td>
<td>1e-02</td>
<td>1.548e-03</td>
<td>1e-02</td>
<td>1.548e-03</td>
</tr>
<tr>
<td>7</td>
<td>1.828e-01</td>
<td>1e-04</td>
<td>3.542e-03</td>
<td>1e-02</td>
<td>1.688e-03</td>
</tr>
<tr>
<td>11</td>
<td>1.258e-01</td>
<td>1e-02</td>
<td>2.213e-03</td>
<td>1e-02</td>
<td>1.675e-03</td>
</tr>
<tr>
<td>17</td>
<td>6.551e-02</td>
<td>1e-02</td>
<td>2.312e-03</td>
<td>1e-02</td>
<td>1.971e-03</td>
</tr>
</tbody>
</table>

Table 2.1: Burgers equation, $\nu = 10^{-3}$, reconstructive regime, optimal tol, tol$_g$, 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 2.1, 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 2.1.
Figure 2.1: 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 2.2, 2.3, 2.4, and 2.5, 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 2.2), $10^1$ (Table 2.3), $10^0$ (Table 2.4), and $10^{-1}$ (Table 2.5). 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.

<table>
<thead>
<tr>
<th>$r$</th>
<th>G-ROM $\mathcal{E}(L^2)$</th>
<th>2S-DD-VMS-ROM $\mathcal{E}(L^2)$</th>
<th>3S-DD-VMS-ROM $\mathcal{E}(L^2)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1.181e-01</td>
<td>1.181e-01</td>
<td>1e+00</td>
</tr>
<tr>
<td>7</td>
<td>1.828e-01</td>
<td>1.828e-01</td>
<td>1e-01</td>
</tr>
<tr>
<td>11</td>
<td>1.258e-01</td>
<td>1.258e-01</td>
<td>1e-01</td>
</tr>
<tr>
<td>17</td>
<td>6.551e-02</td>
<td>6.551e-02</td>
<td>1e-02</td>
</tr>
</tbody>
</table>

Table 2.2: 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.

<table>
<thead>
<tr>
<th>$r$</th>
<th>G-ROM $\mathcal{E}(L^2)$</th>
<th>2S-DD-VMS-ROM $\mathcal{E}(L^2)$</th>
<th>3S-DD-VMS-ROM $\mathcal{E}(L^2)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1.181e-01</td>
<td>7.278e-02</td>
<td>1e+00</td>
</tr>
<tr>
<td>7</td>
<td>1.828e-01</td>
<td>1.755e-01</td>
<td>1e-03</td>
</tr>
<tr>
<td>11</td>
<td>1.258e-01</td>
<td>1.229e-01</td>
<td>1e-03</td>
</tr>
<tr>
<td>17</td>
<td>6.551e-02</td>
<td>6.456e-02</td>
<td>1e-02</td>
</tr>
</tbody>
</table>

Table 2.3: 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.

In Tables 2.6, 2.7, and 2.8, 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 $tol_S = tol$:...
### 2.3. Numerical Results

<table>
<thead>
<tr>
<th>$r$</th>
<th>G-ROM</th>
<th>2S-DD-VMS-ROM</th>
<th>3S-DD-VMS-ROM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\mathcal{E}(L^2)$</td>
<td>$\mathcal{E}(L^2)$</td>
<td>$r_1$</td>
</tr>
<tr>
<td>3</td>
<td>1.181e-01</td>
<td>1.333e-01</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1.828e-01</td>
<td>2.590e-02</td>
<td>2</td>
</tr>
<tr>
<td>11</td>
<td>1.258e-01</td>
<td>3.607e-02</td>
<td>2</td>
</tr>
<tr>
<td>17</td>
<td>6.551e-02</td>
<td>5.029e-02</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 2.4: 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.

<table>
<thead>
<tr>
<th>$r$</th>
<th>G-ROM</th>
<th>2S-DD-VMS-ROM</th>
<th>3S-DD-VMS-ROM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\mathcal{E}(L^2)$</td>
<td>$\mathcal{E}(L^2)$</td>
<td>$r_1$</td>
</tr>
<tr>
<td>3</td>
<td>1.181e-01</td>
<td>3.729e-03</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1.828e-01</td>
<td>4.232e-03</td>
<td>4</td>
</tr>
<tr>
<td>11</td>
<td>1.258e-01</td>
<td>4.556e-03</td>
<td>2</td>
</tr>
<tr>
<td>17</td>
<td>6.551e-02</td>
<td>5.962e-03</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 2.5: 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.

$10^0$ (Table 2.6), $10^{-1}$ (Table 2.7), and $10^{-2}$ (Table 2.8). 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 accurate than the 2S-DD-VMS-ROM. Overall, the 3S-DD-VMS-ROM is the most accurate ROM.

<table>
<thead>
<tr>
<th>$r$</th>
<th>G-ROM</th>
<th>2S-DD-VMS-ROM</th>
<th>3S-DD-VMS-ROM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\mathcal{E}(L^2)$</td>
<td>$\mathcal{E}(L^2)$</td>
<td>$r_1$</td>
</tr>
<tr>
<td>3</td>
<td>1.181e-01</td>
<td>1.133e-02</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>1.828e-01</td>
<td>2.590e-02</td>
<td>6</td>
</tr>
<tr>
<td>11</td>
<td>1.258e-01</td>
<td>3.607e-02</td>
<td>10</td>
</tr>
<tr>
<td>17</td>
<td>6.551e-02</td>
<td>5.029e-02</td>
<td>16</td>
</tr>
</tbody>
</table>

Table 2.6: 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.
Table 2.7: 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.

<table>
<thead>
<tr>
<th>$r$</th>
<th>G-ROM</th>
<th>2S-DD-VMS-ROM</th>
<th>3S-DD-VMS-ROM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\mathcal{E}(L^2)$</td>
<td>$\mathcal{E}(L^2)$</td>
<td>$r_1$</td>
</tr>
<tr>
<td>3</td>
<td>1.181e-01</td>
<td>3.729e-03</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>1.828e-01</td>
<td>4.232e-03</td>
<td>4</td>
</tr>
<tr>
<td>11</td>
<td>1.258e-01</td>
<td>4.556e-03</td>
<td>10</td>
</tr>
<tr>
<td>17</td>
<td>6.551e-02</td>
<td>5.962e-03</td>
<td>16</td>
</tr>
</tbody>
</table>

Table 2.8: 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.

<table>
<thead>
<tr>
<th>$r$</th>
<th>G-ROM</th>
<th>2S-DD-VMS-ROM</th>
<th>3S-DD-VMS-ROM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\mathcal{E}(L^2)$</td>
<td>$\mathcal{E}(L^2)$</td>
<td>$r_1$</td>
</tr>
<tr>
<td>3</td>
<td>1.181e-01</td>
<td>1.548e-03</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1.828e-01</td>
<td>1.062e-02</td>
<td>4</td>
</tr>
<tr>
<td>11</td>
<td>1.258e-01</td>
<td>2.213e-03</td>
<td>4</td>
</tr>
<tr>
<td>17</td>
<td>6.551e-02</td>
<td>2.312e-03</td>
<td>4</td>
</tr>
</tbody>
</table>

The results in Tables 2.2–2.8 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 2.9, we list the ROMs errors for the cross-validation regime with optimal $tol$ in Algorithm 3 and optimal $tol_L$ and $tol_S$ in Algorithm 4. 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.

In Table 2.10, we list the ROMs errors for the predictive regime with optimal $tol$ in Algorithm 3 and optimal $tol_L$ and $tol_S$ in Algorithm 4. 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.
2.3. NUMERICAL RESULTS

Table 2.9: 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.

<table>
<thead>
<tr>
<th>$r$</th>
<th>G-ROM</th>
<th>2S-DD-VMS-ROM</th>
<th>3S-DD-VMS-ROM</th>
<th>$\mathcal{E}(L^2)$</th>
<th>$tol$</th>
<th>$\mathcal{E}(L^2)$</th>
<th>$tol_S$</th>
<th>$tol_L$</th>
<th>$\mathcal{E}(L^2)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>2.015e-01</td>
<td>1e-01</td>
<td>2.028e-02</td>
<td>2</td>
<td>1e-01</td>
<td>1e+00</td>
<td>1.863e-02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1.796e-01</td>
<td>5e-02</td>
<td>1.400e-02</td>
<td>3</td>
<td>5e-02</td>
<td>1e+00</td>
<td>1.188e-02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1.163e-01</td>
<td>3e-02</td>
<td>8.981e-03</td>
<td>6</td>
<td>3e-02</td>
<td>1e+00</td>
<td>8.383e-03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>6.897e-02</td>
<td>1e-02</td>
<td>8.542e-03</td>
<td>6</td>
<td>1e-02</td>
<td>1e+00</td>
<td>8.452e-03</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2.10: 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.

<table>
<thead>
<tr>
<th>$r$</th>
<th>G-ROM</th>
<th>2S-DD-VMS-ROM</th>
<th>3S-DD-VMS-ROM</th>
<th>$\mathcal{E}(L^2)$</th>
<th>$tol$</th>
<th>$\mathcal{E}(L^2)$</th>
<th>$tol_S$</th>
<th>$tol_L$</th>
<th>$\mathcal{E}(L^2)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>2.185e-01</td>
<td>1e-01</td>
<td>3.623e-02</td>
<td>2</td>
<td>1e-01</td>
<td>1e+00</td>
<td>3.029e-02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>2.054e-01</td>
<td>3e-02</td>
<td>2.004e-02</td>
<td>6</td>
<td>5e-02</td>
<td>3e-02</td>
<td>1.428e-02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1.620e-01</td>
<td>3e-02</td>
<td>1.608e-02</td>
<td>10</td>
<td>5e-02</td>
<td>3e-02</td>
<td>1.418e-02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>1.103e-01</td>
<td>1e-02</td>
<td>1.524e-02</td>
<td>6</td>
<td>1e-02</td>
<td>1e-01</td>
<td>1.506e-02</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2.3.3 Flow Past A Cylinder

In this section, we investigate the 2S-DD-VMS-ROM (2.20) and the new 3S-DD-VMS-ROM (2.28) 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 (2.8)–(2.9). The computational domain is a $2.2 \times 0.41$ rectangular channel with a radius $= 0.05$ cylinder, centered at $(0.2, 0.2)$, see Figure 2.2.

We prescribe no-slip boundary conditions on the walls and cylinder, and the following inflow and outflow profiles [38, 49, 50, 65]:

$$u_1(0, y, t) = u_1(2.2, y, t) = \frac{6}{0.41^2} y (0.41 - y),$$  
(2.42)

$$u_2(0, y, t) = u_2(2.2, y, t) = 0,$$  
(2.43)
where $u = (u_1, u_2)$. There is no forcing and the flow starts from rest.

**Snapshot Generation** For the spatial discretization, we use the pointwise divergence-
free, LBB stable $(P_2, P_{1\text{disc}})$ 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 dif-
ferent 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 $Re = 1000$.

![Figure 2.2: Geometry of the flow past a circular cylinder numerical experiment.](image)
2.3. Numerical Results

Numerical Results for $Re = 100$

In this section, we present numerical results for the flow past a cylinder at $Re = 100$.

In Table 2.11, for different $r$ values, we list the average $L^2$ error (2.39) 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 2.11 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 2.3.2).

<table>
<thead>
<tr>
<th>$r$</th>
<th>G-ROM $\mathcal{E}(L^2)$</th>
<th>2S-DD-VMS-ROM $\mathcal{E}(L^2)$</th>
<th>3S-DD-VMS-ROM $r_1$</th>
<th>3S-DD-VMS-ROM $\mathcal{E}(L^2)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>9.902e-02</td>
<td>5.118e-04</td>
<td>1</td>
<td>5.088e-04</td>
</tr>
<tr>
<td>3</td>
<td>1.029e-01</td>
<td>3.208e-02</td>
<td>2</td>
<td>3.018e-02</td>
</tr>
<tr>
<td>4</td>
<td>5.840e-02</td>
<td>1.553e-03</td>
<td>2</td>
<td>1.479e-03</td>
</tr>
<tr>
<td>5</td>
<td>6.492e-02</td>
<td>2.270e-02</td>
<td>4</td>
<td>2.191e-02</td>
</tr>
<tr>
<td>6</td>
<td>1.370e-02</td>
<td>5.336e-04</td>
<td>1</td>
<td>4.804e-04</td>
</tr>
<tr>
<td>7</td>
<td>1.403e-02</td>
<td>6.038e-03</td>
<td>6</td>
<td>5.817e-03</td>
</tr>
<tr>
<td>8</td>
<td>1.214e-02</td>
<td>9.302e-04</td>
<td>6</td>
<td>4.415e-04</td>
</tr>
</tbody>
</table>

Table 2.11: 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 2.12, for different $r$ values, we list the average $L^2$ error (2.39) 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.

<table>
<thead>
<tr>
<th>$r$</th>
<th>G-ROM</th>
<th>2S-DD-VMS-ROM</th>
<th>3S-DD-VMS-ROM</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>4.891e-01</td>
<td>1.536e-03</td>
<td>1.458e-03</td>
</tr>
<tr>
<td>3</td>
<td>4.088e-01</td>
<td>3.514e-02</td>
<td>3.106e-02</td>
</tr>
<tr>
<td>4</td>
<td>9.291e-02</td>
<td>2.187e-03</td>
<td>2.015e-03</td>
</tr>
<tr>
<td>5</td>
<td>1.013e-01</td>
<td>2.279e-02</td>
<td>2.220e-02</td>
</tr>
<tr>
<td>6</td>
<td>3.270e-02</td>
<td>5.113e-04</td>
<td>4.921e-04</td>
</tr>
<tr>
<td>7</td>
<td>3.059e-02</td>
<td>7.476e-03</td>
<td>7.260e-03</td>
</tr>
<tr>
<td>8</td>
<td>3.600e-02</td>
<td>1.221e-03</td>
<td>4.385e-04</td>
</tr>
</tbody>
</table>

Table 2.12: 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 2.13, for different $r$ values, we list the average $L^2$ error (2.39) 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 $r \geq 4$, the 3S-DD-VMS-ROM is at least twice more accurate than the 2S-DD-VMS-ROM.

<table>
<thead>
<tr>
<th>$r$</th>
<th>G-ROM</th>
<th>2S-DD-VMS-ROM</th>
<th>3S-DD-VMS-ROM</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3.883e-01</td>
<td>8.172e-02</td>
<td>7.388e-02</td>
</tr>
<tr>
<td>3</td>
<td>3.616e-01</td>
<td>3.374e-02</td>
<td>3.141e-02</td>
</tr>
<tr>
<td>4</td>
<td>1.366e-01</td>
<td>8.127e-03</td>
<td>4.115e-03</td>
</tr>
<tr>
<td>5</td>
<td>1.464e-01</td>
<td>4.248e-02</td>
<td>2.602e-02</td>
</tr>
<tr>
<td>6</td>
<td>1.348e-01</td>
<td>5.946e-03</td>
<td>1.051e-03</td>
</tr>
<tr>
<td>7</td>
<td>1.291e-01</td>
<td>1.529e-02</td>
<td>6.613e-03</td>
</tr>
<tr>
<td>8</td>
<td>9.638e-02</td>
<td>6.798e-03</td>
<td>3.170e-03</td>
</tr>
</tbody>
</table>

Table 2.13: 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.

In Figure 2.3, 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 2.11: 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 \( r = 7 \).

![Kinetic Energy graphs for different \( r \) values](image)

**Figure 2.3:** 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 2.4, 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.

In Figure 2.5, 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 $r = 7$. 
2.3. Numerical Results

Figure 2.5: 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 2.11–2.13 and the plots in Figures 2.3–2.5 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 2.14, for different $r$ values, we list the average $L^2$ error (2.39) 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.

<table>
<thead>
<tr>
<th>$r$</th>
<th>G-ROM $\mathcal{E}(L^2)$</th>
<th>2S-DD-VMS-ROM $\mathcal{E}(L^2)$</th>
<th>3S-DD-VMS-ROM $\mathcal{E}(L^2)$</th>
<th>$r_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2.892e-01</td>
<td>7.029e-03</td>
<td>1</td>
<td>3.937e-03</td>
</tr>
<tr>
<td>3</td>
<td>3.344e-01</td>
<td>8.138e-02</td>
<td>2</td>
<td>7.517e-02</td>
</tr>
<tr>
<td>4</td>
<td>3.478e-01</td>
<td>4.195e-03</td>
<td>3</td>
<td>4.145e-03</td>
</tr>
<tr>
<td>5</td>
<td>3.795e-01</td>
<td>6.811e-02</td>
<td>2</td>
<td>5.915e-02</td>
</tr>
<tr>
<td>6</td>
<td>6.338e-02</td>
<td>3.864e-03</td>
<td>2</td>
<td>3.294e-03</td>
</tr>
<tr>
<td>7</td>
<td>5.738e-02</td>
<td>1.789e-02</td>
<td>2</td>
<td>1.563e-02</td>
</tr>
<tr>
<td>8</td>
<td>5.339e-02</td>
<td>5.734e-03</td>
<td>6</td>
<td>4.809e-03</td>
</tr>
</tbody>
</table>

Table 2.14: 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.

In Table 2.15, for different $r$ values, we list the average $L^2$ error (2.39) 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 2.16, for different $r$ values, we list the average $L^2$ error (2.39) 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 $r = 5, 6, 7, 8$, the 3S-DD-VMS-ROM is more than twice more accurate than the 2S-DD-VMS-ROM.
2.3. Numerical Results

<table>
<thead>
<tr>
<th>$r$</th>
<th>G-ROM</th>
<th>2S-DD-VMS-ROM</th>
<th>3S-DD-VMS-ROM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\mathcal{E}(L^2)$</td>
<td>$\mathcal{E}(L^2)$</td>
<td>$r_1$</td>
</tr>
<tr>
<td>2</td>
<td>1.071e+00</td>
<td>2.015e-02</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>8.280e-01</td>
<td>1.101e-01</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>6.258e-01</td>
<td>1.218e-02</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>6.440e-01</td>
<td>1.557e-01</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>1.898e-01</td>
<td>5.733e-03</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>1.531e-01</td>
<td>3.550e-02</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>1.678e-01</td>
<td>9.050e-03</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 2.15: 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.

<table>
<thead>
<tr>
<th>$r$</th>
<th>G-ROM</th>
<th>2S-DD-VMS-ROM</th>
<th>3S-DD-VMS-ROM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\mathcal{E}(L^2)$</td>
<td>$\mathcal{E}(L^2)$</td>
<td>$r_1$</td>
</tr>
<tr>
<td>2</td>
<td>7.351e-01</td>
<td>1.004e-01</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>7.088e-01</td>
<td>8.838e-02</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>5.871e-01</td>
<td>8.785e-03</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>6.231e-01</td>
<td>9.735e-02</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>1.293e-01</td>
<td>2.288e-02</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>1.069e-01</td>
<td>2.816e-02</td>
<td>6</td>
</tr>
<tr>
<td>8</td>
<td>1.130e-01</td>
<td>1.402e-02</td>
<td>6</td>
</tr>
</tbody>
</table>

Table 2.16: 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 2.6, 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 $r = 7$. 
Figure 2.6: 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 2.7, 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 $r = 7$. 
2.3. Numerical Results

Figure 2.7: 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.

In Figure 2.8, 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.
CHAPTER 2. DATA-DRIVEN VARIATIONAL MULTISCALE REDUCED ORDER MODELS

The errors listed in Tables 2.14–2.16 and the plots in Figures 2.6–2.8 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.

Numerical Results for $Re = 1000$

In this section, we present numerical results for the flow past a cylinder at $Re = 1000$. 

Figure 2.8: 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.
In Table 2.17, for different \( r \) values, we list the average \( L^2 \) error (2.39) 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.

<table>
<thead>
<tr>
<th>( r )</th>
<th>G-ROM ( \mathcal{E}(L^2) )</th>
<th>2S-DD-VMS-ROM ( \mathcal{E}(L^2) )</th>
<th>3S-DD-VMS-ROM ( r_1 ) ( \mathcal{E}(L^2) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>4.937e-01</td>
<td>6.704e-03</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>5.112e-01</td>
<td>6.804e-02</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>5.980e-01</td>
<td>1.287e-02</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>6.579e-01</td>
<td>1.794e-01</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>1.503e-01</td>
<td>1.086e-02</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>1.365e-01</td>
<td>2.848e-02</td>
<td>5</td>
</tr>
<tr>
<td>8</td>
<td>7.076e-02</td>
<td>7.550e-03</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 2.17: 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 2.18, for different \( r \) values, we list the average \( L^2 \) error (2.39) 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.

In Table 2.19, for different \( r \) values, we list the average \( L^2 \) error (2.39) 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.
In Figure 2.9, 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.
2.3. Numerical Results

Figure 2.9: 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 2.10, 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 2.11, 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 energy approximations 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.
2.3. Numerical Results

Figure 2.11: 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 2.17–2.19 and the plots in Figures 2.9–2.11 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.
2.3.4 Quasi-Geostrophic Equations (QGE)

In this section, we investigate the 2S-DD-VMS-ROM (2.20) and the new 3S-DD-VMS-ROM (2.28) in the numerical simulation of the quasi-geostrophic equations (QGE)

$$\frac{\partial \omega}{\partial t} + J(\omega, \psi) - Ro^{-1} \frac{\partial \psi}{\partial x} = Re^{-1} \Delta \omega + Ro^{-1} F,$$  \hspace{1cm} (2.44)

$$\omega = -\Delta \psi,$$ \hspace{1cm} (2.45)

which are used to model the large scale ocean circulation $[47, 82]$. In (2.44)–(2.45), $\omega$ is the vorticity, $\psi$ 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

$$F = \sin(\pi(y - 1)),$$ \hspace{1cm} (2.46)

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 $\psi$ and $\omega$ satisfy homogeneous Dirichlet boundary conditions:

$$\psi(t, x, y) = 0, \quad \omega(t, x, y) = 0 \quad \text{for} \quad (x, y) \in \partial \Omega \quad \text{and} \quad t \geq 0.$$ \hspace{1cm} (2.47)

Snapshot Generation  For the FOM discretization, we use a spectral method with a 257 $\times$ 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 [49], and Section 3.2 in [52]). Specifically, we replace \( \tau^{FOM} \) with \( \tau^d \) (for the 2S-DD-VMS-ROM) and \( \tau_{L}^{FOM} \) and \( \tau_{S}^{FOM} \) with \( \tau_{L}^{d} \) and \( \tau_{S}^{d} \), 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 (2.20) and the new 3S-DD-VMS-ROM (2.28) in the numerical simulation of the QGE (2.44)–(2.45). 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 \([10, 80]\):

\[
\overline{\mathcal{E}}(L^2) = \left\| \overline{\psi^{FOM}(x, \cdot)} - \overline{\psi^{ROM}(x, \cdot)} \right\|_{L^2}^2 ,
\]

(2.48)

where \( \overline{\cdot} \) denotes the time average over the time interval \([10, 80]\), and \( x = (x, y) \). In Table 2.20, 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.
In Figure 2.12, 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 2.20: 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.

<table>
<thead>
<tr>
<th>( r )</th>
<th>G-ROM ( \bar{E}(L^2) )</th>
<th>2S-DD-VMS-ROM ( \bar{E}(L^2) )</th>
<th>3S-DD-VMS-ROM ( \bar{E}(L^2) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>3.734e+02</td>
<td>5.174e-01</td>
<td>1.996e-01</td>
</tr>
<tr>
<td>15</td>
<td>1.035e+02</td>
<td>3.853e-01</td>
<td>1.260e-01</td>
</tr>
<tr>
<td>20</td>
<td>1.371e+01</td>
<td>1.653e-01</td>
<td>5.175e-02</td>
</tr>
<tr>
<td>25</td>
<td>3.491e+00</td>
<td>3.434e-01</td>
<td>5.640e-02</td>
</tr>
</tbody>
</table>

Table 2.20: 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.
2.3. Numerical Results

Figure 2.12: 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 2.13, for $r = 10$ and $r = 20$, we plot the time-average of the streamfunction $\psi$ 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 2.13 support the conclusions in Table 2.20: 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 2.13: QGE, $Re = 450$, $Ro = 0.0036$, reconstructive regime. Time-averaged streamfunction $\overline{\psi}$ over the interval $[10, 80]$ for FOM, G-ROM, 2S-DD-VMS-ROM, and 3S-DD-VMS-ROM for different $r$ values.
2.3.5 Backward Facing Step

In this section, we investigate the 2S-DD-VMS-ROM (2.20) and the new 3S-DD-VMS-ROM (2.28) 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 (2.8)–(2.9). We use the same computational domain as that used in Section 4.4 [5] and Section 8.2.2 in [67], i.e., a $44 \times 9$ rectangle with a unit height step placed at $(4, 0)$ (see the top plot in Figure 2.14).

Figure 2.14: 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 \((P_2, P^{disc}_1)\) Scott-Vogelius elements to be LBB stable (for details, see [40]). The mesh (see the top plot in Figure 2.14) 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 2.14 (the bottom plot), we display the magnitude of the FOM velocity field at \(t = 125\).

In Figure 2.15, 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 2.3.2–2.3.4.

![Figure 2.15: Backward facing step, Re = 1000. Time evolution of the FOM kinetic energy.](image)

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 2.3.4 and replace \(\tau^{FOM}\) with \(\tau^{3r}\) (for the 2S-DD-VMS-ROM) and \(\tau^{FOM}_L\) and \(\tau^{FOM}_S\) with \(\tau^{3r}_L\) and \(\tau^{3r}_S\), 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 \(r_1 = \lfloor r/2 \rfloor\).
2.3. Numerical Results

Numerical Results

Next, we present results for the 2S-DD-VMS-ROM (2.20) and the new 3S-DD-VMS-ROM (2.28) 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 2.21, for different \(r\) values, we list the average \(L^2\) error (2.39) 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., \(20 \leq r \leq 60\)).

<table>
<thead>
<tr>
<th>(r)</th>
<th>G-ROM (\mathcal{E}(L^2))</th>
<th>2S-DD-VMS-ROM (\mathcal{E}(L^2))</th>
<th>3S-DD-VMS-ROM (\mathcal{E}(L^2))</th>
<th>(r_1)</th>
<th>G-ROM (\mathcal{E}(L^2))</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1.0270e+00</td>
<td>9.6593e-01</td>
<td>8.6129e-01</td>
<td>1</td>
<td>1.0270e+00</td>
</tr>
<tr>
<td>5</td>
<td>1.4864e+00</td>
<td>1.1671e+00</td>
<td>1.1070e+00</td>
<td>1</td>
<td>1.4864e+00</td>
</tr>
<tr>
<td>10</td>
<td>1.8401e+00</td>
<td>1.5064e+00</td>
<td>1.2932e+00</td>
<td>2</td>
<td>1.8401e+00</td>
</tr>
<tr>
<td>15</td>
<td>1.4733e+00</td>
<td>1.0909e+00</td>
<td>7.4297e-01</td>
<td>9</td>
<td>1.4733e+00</td>
</tr>
<tr>
<td>20</td>
<td>1.0392e+00</td>
<td>7.5813e-01</td>
<td>7.0704e-01</td>
<td>3</td>
<td>1.0392e+00</td>
</tr>
<tr>
<td>40</td>
<td>4.4118e-01</td>
<td>2.9694e-01</td>
<td>2.7753e-01</td>
<td>20</td>
<td>4.4118e-01</td>
</tr>
<tr>
<td>50</td>
<td>2.5578e-01</td>
<td>1.6002e-01</td>
<td>1.5586e-01</td>
<td>25</td>
<td>2.5578e-01</td>
</tr>
<tr>
<td>60</td>
<td>1.6772e-01</td>
<td>1.1679e-01</td>
<td>1.1276e-01</td>
<td>30</td>
<td>1.6772e-01</td>
</tr>
</tbody>
</table>

Table 2.21: 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 2.16, 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 2.16: 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 2.17, for $r = 30, 40, \text{ and } 60$, 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 2.21. 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.
2.3. Numerical Results

Figure 2.17: 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 2.18, 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 2.18: 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 2.21 and the plots in Figures 2.16–2.18 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 2.3.2–2.3.4.
2.3.6 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 2.3.2), the flow past a cylinder (Section 2.3.3), the QGE (Section 2.3.4), and the flow over a backward facing step (Section 2.3.5). 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 3, 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 4: (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., [49, 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 2.4 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 2.19, 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 2.19 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 2.19: 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 \((0.43, 0.2)\).](image)

Furthermore, we follow [5] and, for the 2S-DD-VMS-ROM and 3S-DD-VMS-ROM, in Figure 2.20 we plot the first component of the vectors \( \tau^{FOM} \) and \( \tau^{ROM} \) with the FOM and ROM representations of the VMS-ROM closure terms, which are defined in (2.17) for the 2S-DD-VMS-ROM and in (2.24)-(2.25) for the 3S-DD-VMS-ROM. Specifically, at each time step \( t_j, j = 1, \ldots, M \),

\[
\tau^{FOM}(t_j) = - \left[ ((u_R^{FOM}(t_j) \cdot \nabla) u_R^{FOM}(t_j), \varphi_i) 
- ((u_r^{FOM}(t_j) \cdot \nabla) u_r^{FOM}(t_j), \varphi_i) \right],
\]  

(2.49)
where $u_{FOM}^{R}(t_j)$ and $u_{FOM}^{r}(t_j)$ are defined in (2.19), and
\[ \mathbf{r}_{ROM}(t_j) = \tilde{A} \mathbf{a}_{ROM}(t_j) + \mathbf{a}_{ROM}(t_j) \mathbf{B}\mathbf{a}_{ROM}(t_j)^T, \]
(2.50)
where $\tilde{A}$ and $\tilde{B}$ are the DD-VMS-ROM operators, and $\mathbf{a}_{ROM}(t_j)$ is the ROM solution at time step $t_j$. The plot in Figure 2.20 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 2.19).

Figure 2.20: 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 2.20 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 2.21 (see also Figures 2.16–2.18) 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 2.3.2–2.3.5) 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.

2.4 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 $g$ is similar to the structure of the Galerkin model function $f$ and we utilize a least squares approach to determine the shape of $g$. 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


80
BIBLIOGRAPHY


Chapter 3

Physically Constrained Data-Driven Variational Multiscale Reduced Order Models


The author of this dissertation contributes to the model’s conceptual development and numerical experiments for the quasi-geostrophic equation in Section 3.3.

**ABSTRACT OF CHAPTER 3**

This paper investigates the recently introduced data-driven correction reduced order model (DDC-ROM) in the numerical simulation of the quasi-geostrophic equations. The DDC-ROM uses available data to model the correction term that is generally used to represent the missing information in low-dimensional ROMs. Physical constraints are added to the DDC-ROM to create the constrained data-driven correction reduced order model (CDDC-ROM) in order to further improve its accuracy and stability. Finally, the DDC-ROM and CDDC-ROM are tested on time intervals that are longer than the time interval over which they were trained. The numerical investigation shows that, for low-dimensional ROMs, both the DDC-ROM and CDDC-ROM perform better than the standard Galerkin ROM (G-ROM) and the CDDC-ROM provides the best results.

---

3.1 Introduction

Reduced order models (ROMs) for fluid dynamics have been abundantly investigated in recent decades as a way to reduce the computational cost of high resolution numerical schemes. The success of many ROM approaches has already been documented for various scientific and engineering applications, especially for flows that are governed by relatively few recurrent dominant spatial structures; see e.g. [3, 13, 15, 18, 20, 23, 37, 42, 43, 49] and references therein.

In this article, the recently proposed data-driven correction ROM (DDC-ROM) and its variants [24, 35, 58] are investigated in the numerical simulation of a quasi-geostrophic model of the double-gyre wind-driven ocean circulation. The DDC-ROMs fall into the category of hybrid projection/data-driven ROMs [10, 15, 21, 31, 38]. More specifically, in DDC-ROMs, the interactions among the resolved modes are the same as those in the standard Galerkin projection ROMs, while the interactions involving the unresolved modes are learned through a data-driven approach by fitting, e.g., a quadratic ansatz to the data that represents these missing interactions. In the following, we provide a brief derivation of the DDC-ROMs that builds on the standard projection ROMs, and refer to [35, 58] for more details.

To construct the standard projection ROM, we start with a general nonlinear system that has the following weak form* in a suitable Hilbert space $X$:

$$
(f(u), v) \quad \forall v \in X,
$$

(3.1)

where $f$ is a general nonlinear function, $u \in X$ is the sought solution, and $(\cdot, \cdot)$ denotes the inner product on $X$. We assume that system (3.1) is parameter dependent and/or needs to be simulated for long time intervals. ROMs aim at an efficient and relatively accurate numerical simulation of (3.1). Next, we use data (snapshots) available for a few parameter values and/or a short time interval to construct orthonormal modes $\{\varphi_1, \ldots, \varphi_R\}$, which represent the recurrent spatial structures, where $R$ is the rank of the snapshot matrix and typically $R = \mathcal{O}(10^3)$ or even higher. Then, we choose the dominant modes $\{\varphi_1, \ldots, \varphi_r\}$, typically with $r = \mathcal{O}(10)$, as ROM basis functions. The $r$-dimensional Galerkin ROM (G-ROM) of (3.1) is obtained by replacing $u$ with a Galerkin truncation $u_r = \sum_{j=1}^{r} a_j \varphi_j$ and

*subject to possible further integration by parts for certain terms in $(f(u), v)$. 

restricting \( \mathbf{v} \) to the ROM subspace \( \mathbf{X}^r := \text{span}\{\varphi_1, \ldots, \varphi_r\} \):

\[
(\mathbf{u}_r', \varphi_i) = (f(\mathbf{u}_r), \varphi_i), \quad i = 1, \ldots, r.
\]  \tag{3.2}

In an offline stage, we construct the ROM, and in an online stage, we repeatedly use the G-ROM (3.2) for all the parameter values and/or the entire time interval of interest.

To construct the \textit{data-driven correction reduced order model (DDC-ROM)} \[24, 35, 58\], we use an alternative approach: We start with a new Galerkin truncation, \( \mathbf{u}_R = \sum_{j=1}^{R} a_j \varphi_j \).

We emphasize that, since \( R = \mathcal{O}(10^3) \) is the rank of the snapshot matrix, the new Galerkin truncation includes all the information in the available data (snapshots). Next, we replace \( \mathbf{u} \) with \( \mathbf{u}_R \) in (3.1) and project the resulting PDE onto \( \mathbf{X}^r \):

\[
(\mathbf{u}_R', \varphi_i) = (f(\mathbf{u}_R), \varphi_i), \quad i = 1, \ldots, r.
\]  \tag{3.3}

Since the ROM modes are orthonormal, \( (\mathbf{u}_R', \varphi_i) = (\mathbf{u}_r', \varphi_i), i = 1, \ldots, r \). Thus, (3.3) becomes

\[
(\mathbf{u}_r', \varphi_i) = (f(\mathbf{u}_R), \varphi_i), \quad i = 1, \ldots, r,
\]  \tag{3.4}

which can be written as

\[
(\mathbf{u}_r', \varphi_i) = (f(\mathbf{u}_r), \varphi_i) + [(f(\mathbf{u}_R), \varphi_i) - (f(\mathbf{u}_r), \varphi_i)], \quad i = 1, \ldots, r.
\]  \tag{3.5}

The last term on the right-hand side of (3.5) is a \textit{Correction term}

\[
\text{Correction} = [(f(\mathbf{u}_R), \varphi_i) - (f(\mathbf{u}_r), \varphi_i)].
\]  \tag{3.6}

Thus, (3.5) can be written as G-ROM + Correction.

We emphasize that (3.5) is expected to be more accurate than the G-ROM, since the former is constructed from \( \mathbf{u}_R \), whereas the latter is constructed from \( \mathbf{u}_r \), where \( r = \mathcal{O}(10) \ll R = \mathcal{O}(10^3) \). Note that (3.5) is not yet a closed system in \( \mathbf{u}_r \), since the Correction term involves \( \mathbf{u}_R \), which lives in a higher-dimensional space than \( \mathbf{X}^r \). Thus, to obtain from (3.5) an efficient \( r \)-dimensional ROM, we make the ansatz

\[
\text{Correction} = [(f(\mathbf{u}_R), \varphi_i) - (f(\mathbf{u}_r), \varphi_i)] \approx (g(\mathbf{u}_r), \varphi_i),
\]  \tag{3.7}
3.1. Introduction

where $g$ is a generic function (e.g., polynomial) whose coefficients/parameters still need to be determined. Once $g$ is determined, the ROM (3.5) with the Correction term replaced by $g$ yields the *data-driven correction ROM (DDC-ROM)*:

$$
\begin{align*}
(\mathbf{u}_r, \varphi_i) &= \left(f(\mathbf{u}_r), \varphi_i\right) + \left(g(\mathbf{u}_r), \varphi_i\right), & i = 1, \ldots, r.
\end{align*}
$$

To determine the coefficients/parameters of the function $g$ used in (3.8), we use *data-driven modeling* [5, 30, 41], i.e., we solve the following *least squares problem*:

$$
\min_{g \text{ parameters}} \sum_{j=1}^{M} \left\| \text{Correction}(t_j) - \left(g(\mathbf{u}_r(t_j)), \varphi_i\right) \right\|^2.
$$

The numerical investigations in [24, 35, 58] show that the DDC-ROM (3.8) is significantly more accurate than the standard G-ROM in the numerical simulation of two test problems: (i) the 1D Burgers equation with a small diffusion coefficient $\nu = 10^{-3}$; and (ii) a 2D flow past a circular cylinder at Reynolds numbers $Re = 100$, $Re = 500$, and $Re = 1000$.

The main goal of this paper is to investigate the new DDC-ROM (3.8) in the numerical simulation of the *quasi-geostrophic equations (QGE)*, which represent a significantly more difficult test case than the Burgers equation and the 2D flow past a circular cylinder considered in [35, 58]. Indeed, for the 2D flow past a circular cylinder with the Reynolds number $Re = 1000$, the projection of the velocity field onto the first 8 POD modes captures more than 99% of the kinetic energy. In contrast, for the QGE in the parameter regime investigated in Section 3.3, a much broader range of spatial scales are actively involved in the time evolution of the turbulent fluid field. Indeed, the total amount of kinetic energy captured by the leading POD modes increases much slower for the QGE investigated here: it requires 16 POD modes to capture 90% of the kinetic energy, 37 modes for 95% of the kinetic energy, and 49 modes for 96% of the kinetic energy.

Furthermore, given the challenges posed by the QGE, we investigate two improvements to the DDC-ROM (3.8): First, we study the role of adding *physical constraints* to the DDC-ROM [35], in which the model for the Correction term in (3.7) satisfies the same type of physical constraints as those satisfied by the underlying equations; see (3.25). We also investigate whether modeling the *commutation error*, i.e., the error that appears as a
result of interchanging spatial differentiation and ROM spatial filtering (e.g., projection) [24], improves the DDC-ROM accuracy. Finally, we study the DDC-ROM when it is trained on a time interval that is shorter than the time interval over which it is tested.

3.1.1 Connections to Previous Work

The DDC-ROM belongs to the class of ROM closure models, which model the effect of the truncated ROM modes (i.e., $\{\varphi_{r+1}, \ldots, \varphi_R\}$) on the resolved ROM modes, (i.e., $\{\varphi_1, \ldots, \varphi_r\}$). ROM closure models were first proposed in the pioneering work of Lumley and his collaborators [23] and are currently witnessing a dynamic development in several new directions, e.g., ROM spatial filtering, large eddy simulation (LES), and variational multiscale (VMS) [1, 2, 4, 44, 57], Mori-Zwanzig (MZ) formalism [40], nonlinear autoregression, moving averages with exogenous inputs (NARMAX) [9, 31], multilevel approaches and empirical model reduction (EMR) [26, 27, 32], data-adaptive harmonic decomposition and multilayer Stuart-Landau models (DAH-MSLM) [6, 25], and the parameterizing manifold (PM) approach rooted in the approximation theory of local invariant manifolds [7, 8], to name just a few. Probably the most dynamic development has been in using available data and machine learning techniques to develop ROM closure models [21, 34, 39, 46, 47, 56, 59].

The DDC-ROM is a hybrid projection/data-driven ROM, in which the standard Galerkin method is used to model the terms involving only the resolved modes and available data is used to model only the ROM closure term. This parsimonious/minimalistic data-driven approach is made possible by using ROM spatial filtering (i.e., ROM projection) and an LES/VMS framework to isolate the ROM closure term, which is then approximated by using data. The DDC-ROM minimalistic data-driven framework is similar in spirit to the NARMAX [9, 31] and PM [7, 8] ROM closure models, although they differ in the way the closure terms are handled. The DDC-ROM centers around ROM spatial filtering, whereas in the NARMAX approach the closure terms are modeled using nonlinear autoregression moving average with the resolved modes as exogenous inputs, and the PM approach parameterizes explicitly the unresolved modes in terms of the resolved modes.

The rest of the paper is organized as follows: In Section 3.2, we briefly present the QGE and the construction of the corresponding DDC-ROM. In Section 3.3, we assess the performance of the DDC-ROM using two metrics: the time-averaged streamfunction and the
kinetic energy. Finally, in Section 3.4, we summarize our findings and outline future research directions.

3.2 Data-Driven Correction ROM (DDC-ROM)

In this section, we present the construction of the DDC-ROM for the QGE.

3.2.1 Quasi-geostrophic Equations (QGE)

In what follows, we use the quasi-geostrophic equations (QGE) as a mathematical model:

\[
\frac{\partial \omega}{\partial t} + J(\omega, \psi) - Ro^{-1} \frac{\partial \psi}{\partial x} = Re^{-1} \Delta \omega + Ro^{-1} F, \tag{3.10}
\]

\[
\omega = -\Delta \psi, \tag{3.11}
\]

where \( \omega \) is the vorticity, \( \psi \) is the streamfunction, \( Re \) is the Reynolds number, and \( Ro \) is the Rossby number. As a test problem for numerical investigation, we consider the QGE (3.10)–(3.11) with a symmetric double-gyre wind forcing given by

\[
F = \sin(\pi(y - 1)). \tag{3.12}
\]

The single-layer QGE (3.10)–(3.11) (also known as a barotropic vorticity equation (BVE)), are a popular mathematical model for forced-dissipative large scale ocean circulation [12, 33, 55]. The QGE are a simplified model that allows efficient numerical simulations while preserving many of the essential features of the underlying large scale ocean flows. The QGE are similar to the streamfunction-vorticity formulation of the two-dimensional Navier Stokes equations (NSE) [17]. The main difference between the two equations is that the QGE include rotation effects (due to the Coriolis force), which yield an additional term \( -Ro^{-1} \frac{\partial \psi}{\partial x} \) and an additional parameter \( Ro \). The idealized double-gyre wind forcing setting has been often used to understand the wind-driven circulation, e.g., the role of mesoscale eddies and their effect on the mean circulation. ROMs for the QGE (3.10)–(3.11) have been used, e.g., in [11, 14, 45, 51, 52].

The spatial domain of the QGE is \( \Omega = [0, 1] \times [0, 2] \) and the time domain is \([0, 80]\). We
assume that $\psi$ and $\omega$ satisfy homogeneous Dirichlet boundary conditions:

$$
\psi(t, x, y) = 0, \quad \omega(t, x, y) = 0 \quad \text{for } (x, y) \in \partial \Omega \text{ and } t \geq 0. \quad (3.13)
$$

The QGE (3.10)–(3.11) can be cast in the general form of the nonlinear equation (3.1) by choosing

$$(f(\omega), v) = - (J(\omega, \psi), v) + Ro^{-1} \left( \frac{\partial \psi}{\partial x}, v \right) - Re^{-1} (\nabla \omega, \nabla v) + Ro^{-1} (F, v). \quad (3.14)$$

We refrain from giving the precise formulation of the functional space here since this is tangential to the numerical study carried out below. The interested readers can consult for instance [17, Chapter 11], for the case of Navier-Stokes equations in the streamfunction-vorticity formulation. We will make precise how each of the terms in (3.14) is computed numerically once a set of POD basis functions for the vorticity is computed based on the direct numerical simulation (DNS) data obtained from a spectral code; see Sections 3.2.2 and 3.3.1.

### 3.2.2 Standard Galerkin ROM (G-ROM)

In our investigation, the ROM basis is obtained by using the proper orthogonal decomposition (POD)[23, 37]. We note, however, that other bases could be used for this purpose as well; examples include the dynamic mode decomposition [50], the principal interaction patterns [19, 28, 29], and the HIGAMod [42]. See also [11, 53, 54] for recent surveys and relationships/comparisons between different modal decomposition approaches.

We focus here mainly on the functional form of the G-ROM and defer details about the POD basis construction to Section 3.3. To this end, given an $r$-dimensional ROM subspace $X^r$ spanned by the first $r$ POD basis functions for the vorticity $\omega$,

$$
X^r := \text{span}\{\varphi_1, \ldots, \varphi_r\}, \quad (3.15)
$$

the $r$-dimensional G-ROM takes the form of (3.2) with $f$ therein given by (3.14). Recall that the streamfunction $\psi$ in (3.14) is related to the vorticity $\omega$ through the Poisson equation (3.11) subject to homogeneous Dirichlet boundary conditions.
To further reduce the G-ROM to an explicit ODE system, one option would be to replace $\psi$ in (3.14) by $-\Delta^{-1}\omega$, with $\Delta^{-1}$ being the inverse of the Laplacian of $\omega$ subject to the aforementioned boundary conditions. But since one important metric we adopt to assess the performance of the ROMs concerns the time average of $\psi$, we decide to keep $\psi$ explicit in the ROM formulation, although either way would lead to the same $r$-dimensional ODE system. For this reason, we also introduce a reduced set of $r$ basis functions for $\psi$, which are subordinate to the above POD basis functions for $\omega$ in (3.15) via $\phi_i(x,y) = \Delta^{-1}\omega_i(x,y)$, i.e., they solve the following Poisson equation:

$$-\Delta \phi_i(x,y) = \varphi_i(x,y), \quad \text{subject to } \phi_i(x,y) = 0, \text{ for } (x,y) \in \partial \Omega, \quad i = 1,2,\cdots,r. \quad (3.16)$$

Note that while the POD basis $\{\varphi_i\}$ for the vorticity $\omega$ is an orthonormal basis under $L^2$ inner product, the basis $\{\phi_i\}$ for the streamfunction $\psi$ is not orthogonal. Given the G-ROM approximation $\omega_r = \sum_{i=1}^{r} a_i(t) \varphi_i(x,y)$ of $\omega$, the corresponding $\psi$ is approximated by $\psi_r = \sum_{i=1}^{r} a_i(t) \phi_i(x,y)$, which results from the ansatz $\psi_r = -\Delta^{-1}\omega_r$ and definition (3.16) of the basis function $\phi_i$.

With the above notations, the $r$-dimensional G-ROM for the problem (3.10)–(3.13) is given by:

$$\left( \frac{\partial \omega_r}{\partial t}, \varphi_i \right) + (J(\omega_r,\psi_r),\varphi_i) - Ro^{-1} \left( \frac{\partial \psi_r}{\partial x}, \varphi_i \right) + Re^{-1} (\nabla \omega_r, \nabla \varphi_i) = Ro^{-1} (F, \varphi_i), \quad (3.17)$$

where $(\cdot,\cdot)$ denotes the $L^2$ inner product over the spatial domain, and $i = 1,\cdots,r$. Plugging the vorticity and streamfunction ROM approximations in (3.17), yields the Galerkin ROM (G-ROM):

$$\dot{a} = b + A a + a^T B a, \quad (3.18)$$

where $a \overset{def}{=} (a_j(t))_{j=1}^{r}$ is the vector of time-varying ROM coefficients, $b$ is the $r$-dimensional vector corresponding to the forcing term, $A$ is the $r \times r$ matrix corresponding to the linear terms, and $B$ is the $r \times r \times r$ tensor corresponding to the nonlinear term. The G-ROM (3.18)
can be written componentwise as follows: for $i = 1, 2, \ldots, r$

$$\dot{a}_i(t) = b_i + \sum_{m=1}^{r} A_{im} a_m(t) + \sum_{m=1}^{r} \sum_{n=1}^{r} B_{imn} a_m(t) a_n(t),$$  \hspace{1cm} (3.19)

where $b_i = Ro^{-1}(F, \phi_i)$, $A_{im} = Ro^{-1} \left( \frac{\partial \phi_m}{\partial x}, \phi_i \right) - Re^{-1} (\nabla \phi_m, \nabla \phi_i)$, $B_{imn} = - (J(\varphi_m, \phi_n), \phi_i)$.

### 3.2.3 DDC-ROM

To construct the DDC-ROM (3.8) for the QGE, we adapt the general presentation in Section 3.1 to the QGE setting.

For clarity of presentation, we assume that the differentiation and the ROM projection commute (see, however, [24] for a detailed discussion of the commutation error). Thus, the linear terms in the QGE do not appear in the Correction term of the DDC-ROM. Next, we note that the Correction term (3.7) takes the following form for the QGE:

$$\text{Correction} = \left( f(\omega_R) - f(\omega_r), \phi_i \right)$$

$$= \left( J(\omega_R, \psi_R) - J(\omega_r, \psi_r), \phi_i \right)$$

$$= \left( \frac{\partial \omega_R}{\partial x} \frac{\partial \psi_R}{\partial y} - \frac{\partial \psi_R}{\partial x} \frac{\partial \omega_R}{\partial y}, \phi_i \right) - \left( \frac{\partial \omega_r}{\partial x} \frac{\partial \psi_r}{\partial y} - \frac{\partial \psi_r}{\partial x} \frac{\partial \omega_r}{\partial y}, \phi_i \right),$$  \hspace{1cm} (3.20)

where $\omega_R(x, t) = \sum_{i=1}^{R} a_i(t) \phi_i(x)$ and $\psi_R(x, t) = \sum_{i=1}^{R} a_i(t) \phi_i(x)$ are the $R$-dimensional ROM approximations of the vorticity and streamfunction in $X^R$, respectively, and $x = (x, y)$. We emphasize that the Correction (3.20) is $R$-dimensional instead of $r$-dimensional, where $R \gg r$. Thus, to include the Correction (3.20) in the DDC-ROM, we first need to find an efficient, $r$-dimensional approximation of the Correction. To this end, we make the following
3.2. **Data-Driven Correction ROM (DDC-ROM)**

**Linear ansatz:** \( \forall i = 1, \ldots, r, \)

\[
\text{Correction} = \left[ \left( \frac{\partial \omega_R}{\partial x} \frac{\partial \psi_R}{\partial y} - \frac{\partial \psi_R}{\partial x} \frac{\partial \omega_R}{\partial y}, \varphi_i \right) - \left( \frac{\partial \omega_r}{\partial x} \frac{\partial \psi_r}{\partial y} - \frac{\partial \psi_r}{\partial x} \frac{\partial \omega_r}{\partial y}, \varphi_i \right) \right] \\ \approx (g(\omega_r), \varphi_i) \\ = (\tilde{A} \mathbf{a})_i,
\]

(3.21)

where the operator \( \tilde{A} \in \mathbb{R}^{r \times r} \) needs to be determined and \((\tilde{A} \mathbf{a})_i\) denotes the \(i\)-th component of the vector \((\tilde{A} \mathbf{a})\). The ansatz (3.21) is chosen to resemble the right-hand side of the G-ROM (3.18); we note, however, that other ansatzes are possible [35, 58].

To compute the entries in the operator \( \tilde{A} \) in (3.21), we use a data-driven approach. To this end, we adapt the least squares problem (3.9) to the QGE setting:

\[
\min_{\tilde{A}} \sum_{j=1}^{M} \left\| \left( \left( \frac{\partial \omega_R}{\partial x} \frac{\partial \psi_R}{\partial y} - \frac{\partial \psi_R}{\partial x} \frac{\partial \omega_R}{\partial y}, \varphi_i \right) - \left( \frac{\partial \omega_r}{\partial x} \frac{\partial \psi_r}{\partial y} - \frac{\partial \psi_r}{\partial x} \frac{\partial \omega_r}{\partial y}, \varphi_i \right) \right) - \tilde{A} \mathbf{a}^{DNS}(t_j) \right\|^2(3.22)
\]

In (3.22), \( \mathbf{a}^{DNS}(t_j) \) is the vector of ROM coefficients obtained from the DNS data, i.e., from the snapshots, at time instances \( t_j, \ j = 1, \ldots, M, \) which are obtained by projecting the corresponding snapshots \( \omega^{DNS}(t_j) = \sum_{k=1}^{R} a_k^{DNS}(t_j) \varphi_k \) onto the POD basis functions \( \varphi_i \) and using the orthogonality of the POD basis functions: \( \forall i = 1, \ldots, r, \ \forall j = 1, \ldots, M, \)

\[
a_i^{DNS}(t_j) = \left( \omega^{DNS}(t_j), \varphi_i \right). \quad (3.23)
\]

The **data-driven correction ROM (DDC-ROM)** has the following form for the QGE:

\[
\dot{\mathbf{a}} = \mathbf{b} + (A + \tilde{A})\mathbf{a} + \mathbf{a}^\top B \mathbf{a}, \quad (3.24)
\]

where the operators \( \mathbf{b}, A, \) and \( B \) are the G-ROM operators in (3.18) and the operator \( \tilde{A} \) is the solution of the least squares problem (3.22).
3.3 Numerical Experiments

In this section, we investigate the DDC-ROM (3.24) in the numerical simulation of the QGE.

3.3.1 Computational Setting and Snapshot Generation

We investigate the QGE (3.10)–(3.11) in a computational setting similar to that used in [22, 45, 48]. In particular, we use the symmetric double-gyre wind forcing given in (3.12), homogeneous Dirichlet boundary conditions for $\psi$ and $\omega$ given in (3.13), and the parameters $Re = 450$ and $Ro = 0.0036$.

For the DNS spatial discretization, we use a spectral method with a $257 \times 513$ spatial resolution. Since both the vorticity and streamfunction have homogeneous boundary conditions, we approximate both functions with a tensor product Sine expansion in $x$ and $y$. For the DNS time discretization, we use an explicit Runge-Kutta method (Tanaka-Yamashita, an order 7 method with an embedded order 6 method for error control) and an error tolerance of $10^{-8}$ in time with adaptive time refinement and coarsening. We record the solution values every $10^{-2}$ simulation time units (starting at 0) regardless of the current time step size so that the snapshots used in the POD are equally spaced. These spatial and temporal discretizations yield numerical results that are similar to the fine resolution numerical results obtained in [45, 48]. We follow [45, 48] and run the DNS between $[0, 80]$. The time evolution of the spatially averaged kinetic energy in Fig. 1 in [45] shows that, on the time interval $[0, 10]$, the flow displays a transient behavior that is characterized by large changes in the kinetic energy of the system. After this short transient dynamics, the flow reaches a statistically steady state on the interval $[10, 80]$. Capturing the complex dynamics during the initial transient phase would require a large number of ROM modes. Thus, we follow [45, 48] and evaluate the ROMs in the statistically steady regime, i.e., on the time interval $[10, 80]$. We emphasize, however, that even in the statistically steady state regime, the flow displays a high degree of variability. Thus, the numerical approximation of this statistically steady regime remains challenging for the low resolution ROMs that we investigate in this section.
3.3.2 ROM Construction

To generate the ROM basis (see Section 3.2), we follow [45, 48] and collect 701 equally spaced snapshots of the vorticity, $\omega$, in the time interval $[T_{min}, T_{max}] = [10, 80]$ (on which the statistically steady state regime is attained) at equidistant time intervals. We also interpolate the DNS vorticity onto a uniform mesh with the resolution $257 \times 513$ over the rectangle domain $\Omega = [0, 1] \times [0, 2]$, i.e., $h = \Delta x = \Delta y = 1/256$. We then use the 701 snapshots, form the correlation matrix $C$ for vorticity, and obtain the POD basis functions $\varphi_i$’s from the eigenvectors of $C$. Recall that the element $C_{ij}$ of $C$ is simply the $L^2$ inner product of the $i$-th and the $j$-th snapshots, i.e., $C_{ij} = \int_{\Omega} \omega_i \omega_j dx dy$ [45]. Throughout the article, the $L^2$ inner product over $\Omega$ is carried out by using the two-dimensional form of Simpson’s 1/3 rule. Once the POD basis functions for the vorticity are generated, we solve the Poisson equation (3.16) to construct the streamfunction basis functions. For this purpose, we use a second order central difference (five-point stencil) spatial discretization of the Laplace operator.

In Fig. 3.1, we present the contour plots of selected streamfunction basis functions $\phi_i$’s and vorticity POD basis functions $\varphi_i$’s to give an idea about how the spatial scales are organized in these computed bases as the basis function index increases. Note that $\varphi_i$’s are much rougher than $\phi_i$’s, especially for the higher indices, whereas the roughness is smoothed out by the Laplacian when the $\phi_i$’s are computed according to (3.16). One source of roughness could be the uniform discretization mesh adopted here ($257 \times 513$) when interpolating the DNS data, since there are steep gradients in the vorticity field as time evolves, both near the western boundary and within the domain. However, as we will illustrate in the next section, such roughness does not significantly degrade the DDC-ROM accuracy.

To construct the DDC-ROM, we need to solve the least squares problem (3.22), which can be ill-conditioned, especially when the training data is relatively short with respect to the number of coefficients that need to be learnt. To tackle this ill-conditioning issue, we use the truncated singular value decomposition (SVD) (see Step 6 of Algorithm 1 in [58]) with a tolerance that yields the most accurate results. Furthermore, to increase the computational efficiency of the DDC-ROM, we replace $(\omega_R, \psi_R)$ in (3.22) with $(\omega_m, \psi_m)$, where $r \leq m \leq R$. As explained in Section 5.3 in [58], choosing an $m$ value close to $r$ decreases the cost of computing $\tilde{A}$ and $\tilde{B}$, but also reduces the DDC-ROM’s accuracy. On the other hand, choosing an $m$ value close to $R$ increases the DDC-ROM’s accuracy, but also increases the cost of computing $\tilde{A}$ and $\tilde{B}$. Our numerical experiments suggest that $m = 3r$ achieves a
good balance between numerical accuracy and computational efficiency for the considered QGE model.

![Basis functions for the streamfunction (first row) and vorticity (second row).](image)

Figure 3.1: Basis functions for the streamfunction (first row) and vorticity (second row). The vorticity basis functions $\varphi_i$'s are the POD modes computed based on the DNS snapshots for the vorticity, while each streamfunction basis function $\phi_i$ is related to $\varphi_i$ via $\phi_i = -\Delta^{-1}\varphi_i$; see Section 3.2.2.

In our numerical investigation, in addition to the DDC-ROM, we also consider the physically-constrained DDC-ROM (CDDC-ROM) [35], which aims at improving the physical accuracy of the DDC-ROM. To construct the CDDC-ROM, we add physical constraints that require that the data-driven CDDC-ROM operators satisfy the same type of physical laws as those
satisfied by the QGE. Specifically, we require that the CDDC-ROM’s Correction term’s linear component (i.e., the matrix \( \tilde{A} \)) should be dissipative. To implement these physical constraints, in the data-driven modeling step, we replace the unconstrained least squares problem (3.22) with a constrained least squares problem:

\[
\min_{\tilde{A} \in \mathbb{R}^{r \times r}, a^\dagger \tilde{A} a \leq 0} \sum_{j=1}^{M} \left\| \left[ \left( \frac{\partial \omega_R}{\partial x} \frac{\partial \psi_R}{\partial y} - \frac{\partial \psi_R}{\partial x} \frac{\partial \omega_R}{\partial y}, \varphi_i \right) - \left( \frac{\partial \omega_r}{\partial x} \frac{\partial \psi_r}{\partial y} - \frac{\partial \psi_r}{\partial x} \frac{\partial \omega_r}{\partial y}, \varphi_i \right) \right] - \tilde{A} a^{DNS}(t_j) \right\|^2.
\] (3.25)

Additionally, we also monitor the commutation error, which represents the effect of interchanging ROM spatial filtering and differentiation [24]. For this test problem, modeling the commutation error does not significantly change the DDC-ROM and CDDC-ROM results, suggesting that the commutation error does not play a significant role in the ROM construction. Thus, for clarity of presentation, we do not include the commutation error in the DDC-ROM and CDDC-ROM results.

In the online stage, for all the ROMs we utilize the fourth order Runge-Kutta scheme (RK4) for the temporal discretization. To ensure the numerical stability of the time discretization, we choose a time step size \( \Delta t = 0.001 \). We store ROM data every ten time steps to match the DNS sampling rate. We use the DNS snapshot at \( t = 10 \) to initialize the ROMs.

### 3.3.3 Numerical Results

In this section, we present numerical results for the G-ROM, DDC-ROM, and CDDC-ROM. As a benchmark for our numerical investigation, we use the DNS results.

#### Kinetic energy

In this section, we assess the performance of the ROMs using the DNS kinetic energy as a metric. As pointed out in Section 3.1, due to the involvement of a broad range of active spatial scales, it would be too demanding to require any ROMs to reproduce the statistics of the DNS kinetic energy or any other reasonable observables when the dimension is too low, at least within the POD basis framework adopted here. Thus, we confine ourselves instead to a much less ambitious goal of reproducing the range of oscillations presented in the DNS
kinetic energy. The assessment at a more quantitative level will be carried out in the next section for another metric.

Recall that the velocity field \( (u(x, \cdot), v(x, \cdot)) \) used in the computation of the kinetic energy 
\[
E(t) = \frac{1}{2} \int_{\Omega} (u^2(x, t) + v^2(x, t)) \, dx
\]
is related to the streamfunction according to 
\[
(u, v) = (\partial_y \psi, -\partial_x \psi).
\]
The first-order spatial derivatives are calculated using a 4-th order accurate central difference scheme. The kinetic energy itself is computed using the two-dimensional form of Simpson’s 1/3 rule.

In Fig. 3.2, for three different \( r \) values (\( r = 10, 15, \) and 40), we plot the time evolution of the ROM kinetic energy. For \( r = 10 \), the G-ROM kinetic energy takes off very quickly and stabilizes at a level around \( 8 \times 10^4 \), which is roughly 200 times higher than the DNS kinetic energy on average. In contrast, both the DDC-ROM and CDDC-ROM successfully stabilize the G-ROM, and produce kinetic energies almost of the same order of magnitude as the DNS kinetic energy (although there is some overdamping in the CDDC-ROM result due to the physical constraint \( a^T \tilde{A} a \leq 0 \); see (3.25)).

For \( r = 15 \), the G-ROM kinetic energy is within good range at the beginning of the simulation, but increases to an unphysical value around \( t = 40 \) and eventually stabilizes and oscillates around \( 2 \times 10^4 \). In contrast, the DDC-ROM and CDDC-ROM kinetic energies are both within the good range, and between the two, the CDDC-ROM performs better in reproducing the peaks and the corresponding frequency of the peaks.

Finally, for \( r = 40 \), the G-ROM and DDC-ROM perform similarly. The DDC-ROM kinetic energy is closer to the DNS kinetic energy over certain time windows (e.g., [50, 60]), whereas the CDDC-ROM kinetic energy is somewhat lower than the DNS kinetic energy.

The above results suggest that both the DDC-ROM and CDDC-ROM can successfully stabilize a severely truncated G-ROM. For the chosen criterion, the advantage of the DDC-ROM over the G-ROM is clearly visible for all ROM dimensions \( r \) between 10 and 30. The CDDC-ROM can produce results comparable or even better than DDC-ROM for \( r \) between 10 and 20. For even higher dimensions, the CDDC-ROM tends to overdamp the kinetic energy.

This is plausible, since the physical constraint \( a^T \tilde{A} a \leq 0 \) in the estimation of the matrix \( \tilde{A} \) for the CDDC-ROM aims to enhance the stability of the ROM, but does not also guarantee improved accuracy compared to the DDC-ROM.
Figure 3.2: Kinetic energy of DNS, G-ROM, DDC-ROM and CDDC-ROM with different $r$ values. All the ROMs are initialized at $t = 10$ using the projected DNS data.

Relative errors for the time-averaged streamfunction

In this section, we assess the ROM performance at a more quantitative level using the ROM time-averaged streamfunction over the aforementioned time interval, $[10, 80]$. It is known that the time-averaged streamfunction displays a four-gyre structure [16] even though a double-gyre wind forcing is employed; cf. (3.12). The metric that we use is the following relative error:

$$
\frac{\| \overline{\psi^{DNS}(\mathbf{x}, t)} - \overline{\psi^{ROM}(\mathbf{x}, t)} \|_{L^2}^2}{\| \overline{\psi^{DNS}(\mathbf{x}, t)} \|_{L^2}^2},
$$

(3.26)

where $\overline{\cdot}$ represents time average over $[10, 80]$, and $\mathbf{x} = (x, y)$.

In Table 3.1, we list this relative error for each of the ROMs as the ROM dimension $r$ in-
creases. For small $r$ values (i.e., $5 \leq r \leq 20$), the CDDC-ROM is the most accurate. Indeed, for $r = 5$, the CDDC-ROM is the only ROM that yields a stable approximation: all other ROMs with $r = 5$ experience exponential blowup with the given timestep. Furthermore, for $r = 10$ and $r = 15$, the CDDC-ROM error is at most half of the DDC-ROM error. Finally, for $5 \leq r \leq 50$, the G-ROM error is one to three orders of magnitude larger than the CDDC-ROM error. These results are also supported by the plots in Fig. 3.3, which display the time-average of the streamfunction $\psi$ over the time interval $[10, 80]$ for DNS, G-ROM ($r = 10$), DDC-ROM ($r = 10$), and CDDC-ROM ($r = 10$). These plots clearly show that the DDC-ROM and CDDC-ROM are able to capture the correct four-gyre structure, whereas the G-ROM fails drastically at this low ROM dimension.

For large $r$ values (i.e., $25 \leq r \leq 50$), the DDC-ROM results in Table 3.1 are the most accurate. Indeed, at $r = 25$ the CDDC-ROM error starts to increase, whereas the DDC-ROM error generally decreases. The G-ROM error also continues to decrease, but it is always larger than the DDC-ROM error.

We conclude that the CDDC-ROM is the most accurate for small $r$ values and the DDC-ROM is the most accurate for large $r$ values. These results suggest that adding physical constraints to the DDC-ROM is beneficial in the highly truncated cases (i.e., for small $r$ values), but the benefit brought by the physical constraints diminishes as $r$ is further increased, and can even produce less accurate results than DDC-ROM due to overdamping, as pointed out in Section 3.3.3. We emphasize, however, that we are using the linear ansatz to construct the DDC-ROM; further numerical investigations are needed to determine the role of physical constraints when the DDC-ROM is built with a higher-order (e.g., quadratic) ansatz [35]. We also note that the G-ROM is consistently less accurate than the DDC-ROM for all the $r$ values. Finally, we note that, as $r$ increases, the errors for all the ROMs reach a plateau instead of converging to zero. We believe that this behavior is due to the roughness present in the vorticity basis functions, especially for the higher indices (see Fig. 3.1).
3.3. Numerical Experiments

<table>
<thead>
<tr>
<th>$r$ values</th>
<th>G-ROM</th>
<th>DDC-ROM</th>
<th>CDDC-ROM</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r = 5$</td>
<td>n/a</td>
<td>n/a</td>
<td>5.07e+00</td>
</tr>
<tr>
<td>$r = 10$</td>
<td>2.06e+02</td>
<td>3.25e-01</td>
<td>9.58e-02</td>
</tr>
<tr>
<td>$r = 15$</td>
<td>3.05e+02</td>
<td>2.83e-01</td>
<td>1.03e-01</td>
</tr>
<tr>
<td>$r = 20$</td>
<td>5.05e+00</td>
<td>1.39e-01</td>
<td>1.20e-01</td>
</tr>
<tr>
<td>$r = 25$</td>
<td>1.73e+00</td>
<td>1.61e-01</td>
<td>2.96e-01</td>
</tr>
<tr>
<td>$r = 30$</td>
<td>1.47e+00</td>
<td>1.18e-01</td>
<td>4.58e-01</td>
</tr>
<tr>
<td>$r = 35$</td>
<td>6.83e-01</td>
<td>1.34e-01</td>
<td>7.69e-01</td>
</tr>
<tr>
<td>$r = 40$</td>
<td>4.69e-01</td>
<td>9.17e-02</td>
<td>4.92e-01</td>
</tr>
<tr>
<td>$r = 45$</td>
<td>2.81e-01</td>
<td>4.16e-02</td>
<td>6.83e+00</td>
</tr>
<tr>
<td>$r = 50$</td>
<td>3.66e-01</td>
<td>8.20e-02</td>
<td>4.16e-01</td>
</tr>
</tbody>
</table>

Table 3.1: The relative errors for the time-averaged streamfunction defined by (3.26).

![Figure 3.3: Time-averaged streamfunction $\psi$ over the interval $[10, 80]$ for DNS, 10-dim G-ROM, 10-dim DDC-ROM, and 10-dim CDDC-ROM.](image)

Shorter training time interval

In this subsection, we consider the situation when the ROMs are trained on a time interval that is shorter than the time interval over which the ROMs are tested. Specifically, we only use snapshots in the time interval $[10, t^*_p]$, sampled every 0.1 time units as before, to generate the ROM basis and construct the ROM operators $A, B,$ and $\tilde{A}$. This leads to a total number of $(t^*_p - 10)/0.1 + 1$ snapshots. We investigate two different cases: (I) $t^*_p = 45$ and (II) $t^*_p = 35$. 
In Table 3.2, we list the relative errors associated with the ROMs for the time-averaged streamfunction defined in (3.26). The results show that the DDC-ROM is significantly more accurate than the G-ROM, especially for small $r$ values. Moreover, for small $r$ values, the CDDC-ROM is more accurate than the DDC-ROM, whereas for larger $r$ values, the DDC-ROM is more accurate than the DDC-ROM. This is in line with the results in Sections 3.3.3 and 3.3.3. Furthermore, we note that, as expected, the shorter the time interval $[10, t^*_p]$ is, the larger the ROM relative error.

<table>
<thead>
<tr>
<th>$r$ values</th>
<th>Predictive Case I</th>
<th>Predictive Case II</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>G-ROM</td>
<td>DDC-ROM</td>
</tr>
<tr>
<td>$r = 5$</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>$r = 10$</td>
<td>1.39e+04</td>
<td>3.33e-01</td>
</tr>
<tr>
<td>$r = 15$</td>
<td>9.58e+00</td>
<td>3.76e-01</td>
</tr>
<tr>
<td>$r = 20$</td>
<td>5.37e+00</td>
<td>1.35e-01</td>
</tr>
<tr>
<td>$r = 25$</td>
<td>2.28e+00</td>
<td>9.47e-02</td>
</tr>
<tr>
<td>$r = 30$</td>
<td>5.65e-01</td>
<td>1.32e-01</td>
</tr>
<tr>
<td>$r = 35$</td>
<td>2.88e-01</td>
<td>1.76e-01</td>
</tr>
<tr>
<td>$r = 40$</td>
<td>2.07e-01</td>
<td>1.72e-01</td>
</tr>
<tr>
<td>$r = 45$</td>
<td>2.86e-01</td>
<td>2.15e-01</td>
</tr>
<tr>
<td>$r = 50$</td>
<td>1.67e-01</td>
<td>2.00e-01</td>
</tr>
</tbody>
</table>

Table 3.2: The ROM relative errors for the time-averaged streamfunction defined in (3.26) for the two predictive test cases: the POD basis functions are generated using DNS snapshots over the time interval $[10, 45]$ for Case I and over $[10, 35]$ for Case II. The ROM simulations are carried out in the time interval $[10, 80]$.

### 3.4 Conclusions

We enhanced the standard Galerkin ROM (G-ROM) for the quasi-geostrophic equations (QGE) with an additional term derived from available data and a least squares optimization procedure. These ideas are based on previous work and are usually referred to as data-driven correction ROMs (DDC-ROMs) and constrained data-driven correction ROMs (CDDC-ROMs). The latter incorporate a negative semidefiniteness constraint into the optimization problem to preserve a fundamental property of the linear operator in the DDC-ROMs.
The QGE are challenging equations that exhibit complex spatiotemporal behavior: we were able to significantly improve the G-ROM performance by adding an additional term (derived by optimization) to the linear component of the G-ROM. For a ROM with 10 POD modes, the DDC-ROM lowered the error in the mean streamfunction (compared to the G-ROM) by a factor of about 600; similarly, the CDDC-ROM lowered the error by a factor of about 2000.

In the future, we plan on investigating whether using a higher-order (e.g., quadratic) ansatz in the construction of the DDC-ROM and CDDC-ROM yields more accurate results than using a linear ansatz (i.e., the approach utilized in this paper). We also plan to study parameter sensitivity (on both the Reynolds and Rossby numbers) and examine the possibility of constructing a sequence of ROMs that work across a wide range of each value.

Bibliography


Chapter 4

Verifiability of the Data-Driven Variational Multiscale ROM §

The contents of this chapter has not been published.

The author of this dissertation contributes to part of the mathematical analysis of the model and numerical experiments for 2D flow past a cylinder in Section 4.6.4.

ABSTRACT OF CHAPTER 4

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 $Re = 1000$.

4.1 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., $O(10^6)$) systems of equations. Thus, the

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 [37], 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 [8] and theoretically [4]. (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 [40], the Mori-Zwanzig
formalism [22, 28], and the parameterized manifolds [7]. (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 [25, 38]. The DD-VMS-ROM has been investigated numerically in [18, 23, 25, 26, 38, 39]. 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 [5, 14, 32] 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 [6], 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 [11, 13]. Finally, the numerical analysis of the Samagorinsky model in a reduced basis method setting was performed in [2, 31]. (We also note that numerical analysis for regularized ROMs, which are related to but different from ROM closures, was performed in [9, 41].)

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 [25, 38]. 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., [17] as well as [5] 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 4.2, we outline the construction of the standard G-ROM. In Sections 4.3 and 4.4, 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 4.5, we prove the main theoretical result in this paper, i.e., we prove that the DD-VMS-ROM is verifiable. In Section 4.6, 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 4.7, we present the conclusions of our theoretical and numerical investigations and outline several directions for future research.

### 4.2 Galerkin ROM (G-ROM)

In this section, we outline the construction of the Galerkin ROM (G-ROM) for the Navier-Stokes equations (NSE):

\[
\begin{align*}
\frac{\partial \mathbf{u}}{\partial t} - Re^{-1} \Delta \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p &= \mathbf{f}, \\
\nabla \cdot \mathbf{u} &= 0,
\end{align*}
\]

where \( \mathbf{u} \) is the velocity, \( p \) the pressure, and \( Re \) the Reynolds number. The NSE (4.1)–(4.2) are equipped with an initial condition and, for simplicity, homogeneous Dirichlet boundary
4.2. Galerkin ROM (G-ROM)

conditions. To build the ROM basis, we assume that we have access to the snapshots \( \{u_h^0, ..., u_h^M\} \), which are the coefficient vectors of the FEM approximations of the NSE (4.1)–(4.2) at the time instances \( t_0, t_1, ..., t_M \), 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) \([10, 36]\) to construct an orthonormal ROM basis \( \{\varphi_1, ..., \varphi_d\} \), which generates the ROM space \( X^d \) defined as follows:

\[
X^d := \text{span}\{\varphi_1, ..., \varphi_d\},
\]

(4.3)

where \( d \) is the number of linearly independent snapshots \( \{u_h^0, ..., u_h^M\} \). 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 (4.3), we construct \( u_d \), which is the \( d \)-dimensional ROM approximation of NSE velocity, \( u \):

\[
u_d(x, t) = \sum_{i=1}^{d} (a_d)_i(t) \varphi_i(x).
\]

(4.4)

To find the vector of ROM coefficients \( a_d \) in (4.4), we use the Galerkin projection, i.e., we replace \( u \) with \( u_d \) in the NSE (4.1)–(4.2), and then project the resulting equations onto the ROM space, \( X^d \). This yields the \( d \)-dimensional Galerkin ROM (G-ROM):

\[
((u_d)_t, v_d) + Re^{-1}(\nabla u_d, \nabla v_d) + (u_d \cdot \nabla u_d, v_d) = (f, v_d) \quad \forall v_d \in X^d.
\]

(4.5)

We note that the G-ROM (4.5) 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 (4.5) we used a nonlinearity formulation that is equivalent with the nonlinearity formulation in the NSE (4.1) when the velocity field is incompressible (i.e., it satisfies equation (4.2)).

By using the backward Euler time discretization, we get the full discretization of the \( d \)-dimensional G-ROM (4.5) as follows: \( \forall n = 1, ..., M \)

\[
\frac{u_d^n - u_d^{n-1}}{\Delta t}, v_d) + Re^{-1}(\nabla u_d^n, \nabla v_d) + (u_d^n \cdot \nabla u_d^n, v_d) = (f^n, v_d) \quad \forall v_d \in X^d,
\]

(4.6)

where the superscript \( n \) denotes the approximation at time step \( n \). To obtain the finite-dimensional representation of the \( d \)-dimensional G-ROM (4.6), we choose \( v_d \) to be \( \varphi_1, ..., \varphi_d \),
which yields the following system of equations:

\[
\frac{a_d^n - a_d^{n-1}}{\Delta t} = b^n + A a_d^n + (a_d^n)\top B a_d^n, \tag{4.7}
\]

where \(a_d^n\) is the vector of unknown ROM coefficients, \(b\) is a \(d \times 1\) vector, \(A\) is a \(d \times d\) matrix, and \(B\) is a \(d \times d \times d\) tensor. The system of equations in (4.7) can be written componentwise as follows:

\[
\frac{(a_d^n)_i - (a_d^{n-1})_i}{\Delta t} = b^n_i + \sum_{m=1}^{d} A_{im} a_m^n + \sum_{m=1}^{d} \sum_{k=1}^{d} B_{imk} a_m^n a_k^n, \quad 1 \leq i \leq d, \tag{4.8}
\]

where, for \(1 \leq i, m, k \leq d,\)

\[
b^n_i = (f^n, \varphi_i), \tag{4.9}
\]

\[
A_{im} = -Re^{-1} (\nabla \varphi_m, \nabla \varphi_i), \tag{4.10}
\]

\[
B_{imk} = - (\varphi_m \cdot \nabla \varphi_k, \varphi_i). \tag{4.11}
\]

### 4.3 Large Eddy Simulation ROM (LES-ROM)

The ROM closure that we investigate in this paper (i.e., the DD-VMS-ROM presented in Section 4.4) 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., [37, 40]). LES-ROMs are utilizing mathematical principles used in classical LES [5, 33] 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 [40] 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 4.1 (Truth Solution).** For fixed $M$ and $d$, we define the $d$-dimensional G-ROM solution of (4.6) 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 (4.6) 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 $X^d$ into two orthogonal subspaces

\[
X^r := \text{span}\{\varphi_1, \ldots, \varphi_r\}, \quad (4.12a)
\]

\[
(X^r)^\perp := \text{span}\{\varphi_{r+1}, \ldots, \varphi_d\}, \quad (4.12b)
\]

where $X^r$ contains first $r$ dominant ROM basis functions, and $(X^r)^\perp$, which is orthogonal to $X^r$, contains the less energetic ROM basis functions. We also define the following orthogonal projections:

**Definition 4.2 (Orthogonal Projections).** Let $P_r : L^2 \to X^r$ be the orthogonal projection onto $X^r$, and $Q_r : L^2 \to (X^r)^\perp$ be the orthogonal projection onto $(X^r)^\perp$, which can be
defined as

\[ P_r(u) = \sum_{i=1}^{r} (u, \varphi_i)_{L^2} \varphi_i, \quad u \in L^2, \]  

(4.13a)

\[ Q_r(u) = \sum_{i=r+1}^{d} (u, \varphi_i)_{L^2} \varphi_i, \quad u \in L^2. \]  

(4.13b)

Next, in the LES spirit, we decompose the most accurate ROM solution at time step \( n \) (i.e., the \( d \)-dimensional G-ROM solution (4.6), which is the “truth” solution that is employed as a benchmark in our investigation) as

\[ u^n_d := P_r(u^n_d) + Q_r(u^n_d), \]  

(4.14)

where \( P_r \) and \( Q_r \) are the two orthogonal projections in Definition 4.2. Equation (4.14) represents the LES-ROM decomposition of the “truth” solution, \( u^n_d \), into its large scale component, \( P_r(u^n_d) \), and its small scale component, \( Q_r(u^n_d) \).

The ROM spatial filter that we use to construct the LES-ROM is the ROM projection filter [27, 37], i.e., the orthogonal projection \( P_r \) defined in Definition 4.2, which satisfies the following equation: For given \( u \in L^2 \),

\[ (P_r(u), \varphi_i) = (u, \varphi_i), \quad \forall i = 1, \ldots, r. \]  

(4.15)

To construct the LES-ROM, we need to construct the equation satisfied by the large scales, \( P_r(u^n_d) \), defined in (4.14). We note that, by using Definition 4.2 and the ROM orthogonality property, we obtain the following formula for the large scale component \( P_r(u^n_d) \):

\[ P_r(u^n_d) = \sum_{i=1}^{r} (a^n_{d,i}) \varphi_i. \]  

(4.16)

To construct the LES-ROM satisfied by \( P_r(u^n_d) \), we apply the ROM spatial filter, \( P_r \), to the equation satisfied by the “truth” solution, \( u^n_d \) (i.e., to the full discretization of the \( d \)-dimensional G-ROM (4.6)), we restrict the test functions in (4.6) to the \( r \)-dimensional ROM subspace \( X^r \) defined in (4.12a), and we use the decomposition (4.14). This yields the
4.3. LARGE EDDY SIMULATION ROM (LES-ROM)

Equations satisfied by the large scales, \( P_r(u^n_d) \), i.e., the LES-ROM equations:

\[
\left( \frac{P_r(u^n_d) - P_r(u^{n-1}_d)}{\Delta t}, v_r \right) + Re^{-1}(\nabla P_r(u^n_d), \nabla v_r) + (P_r(u^n_d) \cdot \nabla P_r(u^n_d), v_r) \nonumber \\
+ \mathcal{E}^n + (\tau_{FOM}(u^n_d), v_r) = (f^n, v_r), \quad \forall v_r \in X^r,
\]

where we used that, by (4.15), \((P_r(f^n), v_r) = (f^n, v_r)\). In the LES-ROM equations (4.17), the Reynolds stress tensor \( \tau_{FOM}(u^n_d) \) and commutation error \( \mathcal{E} \) are defined as follows:

\[
\tau_{FOM}(u^n_d) := u^n_d \cdot \nabla u^n_d - P_r(u^n_d) \cdot \nabla P_r(u^n_d), \quad (4.18) \\
\mathcal{E}^n := Re^{-1}(\nabla Q_r(u^n_d), \nabla v_r), \quad (4.19)
\]

respectively. We note that, to obtain the LES-ROM equations (4.17), we used the fact that the term \((Q_r(u^n_d), v_r)\) vanishes since \(Q_r(u^n_d)\) is orthogonal to any vector in \(X^r\). We also note that the term \((\nabla Q_r(u^n_d), \nabla v_r)\) in the commutation error term (4.19) does not vanish since the ROM basis functions are only \(L^2\)-orthogonal, not \(H^1_0\)-orthogonal.

**Remark 4.3 (Commutation Error).** In [18], we investigated the effect of the commutation error (4.19) on ROMs. We showed that the commutation error is generally 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 4.4 (Closure Model).** A closure model consists of replacing in (4.17) the Reynolds stress tensor \( \tau_{FOM}(u^n_d) \) by another tensor \( \tau_{ROM}(P_r(u^n_d)) \) depending only on \( P_r(u^n_d) \).

Thus, the role of the closure model \( \tau_{ROM} \) is to replace the true closure model \( \tau_{FOM}(u^n_d) \) (which cannot be computed in \(X^r\)) with a term that can actually be computed in \(X^r\). Since a closure model cannot in general be exact (i.e., \( \tau_{FOM}(u^n_d) \neq \tau_{ROM}(P_r(u^n_d)) \)), when \( \tau_{ROM}(P_r(u^n_d)) \) is inserted for \( \tau_{FOM}(u^n_d) \) in (4.17) the solution of the resulting system is just an approximation to \( P_r(u^n_d) \). We denote this LES-ROM approximation to \( P_r(u^n_d) \) as \( u^n_r \), which can be written as

\[
u^n_r = \sum_{i=1}^r (a^n_r)_i \varphi_i. \quad (4.20)
\]
Thus, the LES-ROM equations for $u^n_r$ are

\[
\frac{u^n_r - u^{n-1}_r}{\Delta t}, v_r) + Re^{-1}(\nabla u^n_r, \nabla v_r) + (u^n_r \cdot \nabla u^n_r, v_r) + (\tau^{ROM}(u^n_r), v_r) = (f^n, v_r), \quad \forall v_r \in \mathbb{X} \tag{4.21}
\]

Inserting (4.20) into (4.21) yields the following matrix form of the LES-ROM:

\[
\frac{a^n_r - a^{n-1}_r}{\Delta t} = b^n + A a^n_r + (a^n_r)^T B a^n_r + [- (\tau^{ROM}(u^n_r), \varphi_i)_{i=1,...,r}], \tag{4.22}
\]

where the vector $b^n$, the matrix $A$, and the tensor $B$ are defined in (4.9)-(4.11).

### 4.4 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 [25, 38]. 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 4.3.

To construct the DD-VMS-ROM, we start from the LES-ROM equations (4.22). First, we notice that since we used the ROM projection as a spatial filter, the LES-ROM (4.22) is in fact a variational multiscale ROM (VMS-ROM). However, the VMS-ROM (4.22) is not closed since the closure term $\tau^{ROM}(u^n_r)$ 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:

\[
-(\tau^{ROM}(u^n_r), \varphi_i)_{i=1,...,r} \approx \tilde{A} a^n_r, \tag{4.23}
\]

where $a^n_r$ is vector of ROM coefficients of the solution $u^n_r$. To determine the $r \times r$ matrix $\tilde{A}$ in (4.23), in the offline stage, we solve the following low-dimensional least squares problem:
4.5. Verifiability of the DD-VMS-ROM

\[
\min_{\tilde{A}} \sum_{n=1}^{M} \left\| \left( \mathbf{u}_d^n \cdot \nabla \mathbf{u}_d^n - P_r(\mathbf{u}_d^n) \cdot \nabla P_r(\mathbf{u}_d^n), \varphi_i \right)_{i=1,...,r} \right\|^2, \tag{4.24}
\]

where \( \mathbf{u}_d^n \) and \( P_r(\mathbf{u}_d^n) \) are obtained from the available FOM data and are defined in (4.4) and (4.16), respectively.

**Physical Constraint**  In the numerical investigation in [8], it was shown that, in the mean, the LES-ROM closure model dissipates energy. Thus, to mimic this behavior, in [23] we equipped the DD-VMS-ROM with a similar physical constraint. Specifically, in the least squares problem (4.24), we added the constraint that \( \tilde{A} \) be negative semidefinite:

\[
(a_r^n)^T \tilde{A} a_r^n \leq 0 \quad \forall a_r^n \in \mathcal{R}^r. \tag{4.25}
\]

Solving the least squares problem (4.24) with the physical constraint (4.25), using the resulting matrix \( \tilde{A} \) in the linear ansatz (4.23), and plugging this in the VMS-ROM (4.22) yields the data-driven variational multiscale ROM (DD-VMS-ROM):

\[
\frac{a_r^n - a_r^{n-1}}{\Delta t} = b^n + (A + \tilde{A})a_r^n + (a_r^n)^T B a_r^n. \tag{4.26}
\]

### 4.5 Verifiability of the DD-VMS-ROM

In this section, we prove the verifiability of the DD-VMS-ROM described in Section 4.4. In Section 4.5.1, we introduce the verifiability and mean dissipativity concepts in the ROM setting. In Section 4.5.2, we prove that the DD-VMS-ROM is verifiable.

#### 4.5.1 Definition of Verifiability and Mean Dissipativity

The goal of this subsection is to define the verifiability of ROM closure models. Verifiability of closure models has been investigated for decades in classical CFD (see, e.g., [17] as well as [5] 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 4.5.2 to prove the verifiability of the DD-VMS-ROM.

**Definition 4.5 (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:

$$
\|P_r(u_d^n) - u_r^n\|_{L^2}^2 \leq C \frac{1}{n} \sum_{j=1}^{n} \|P_r(\tau^{FOM}(u_d^j) - \tau^{ROM}(P_r(u_d^j)))\|_{L^2}^2,
$$

(4.27)

where $u_d^j$ represents the “truth” solution (i.e., the $d$-dimensional G-ROM solution of (4.6)) at $t = t_j$, $j = 1, \ldots, M$, and $u_r^n$ solves the ROM equipped with the given ROM closure model at $t = t_n$, $n = 1, \ldots, M$.

Definition 4.5 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 4.6 (Mean Dissipativity).** A ROM closure model satisfies the mean dissipativity condition if $P_r(u_d^n), u_r^n \in X^r$ satisfy the following inequalities:

$$
0 \leq (\tau^{ROM}(P_r(u_d^n))) - \tau^{ROM}(u_r^n), P_r(u_d^n) - u_r^n < \infty.
$$

(4.28)

### 4.5.2 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 4.7.** The DD-VMS-ROM with linear ansatz (4.26) and physical constraint (4.25) satisfies mean dissipativity according to Definition 4.6.

**Proof.** The least squares problem (4.24) yields the ROM operator $\tilde{A}$ for $-(\tau^{ROM}(P_r(u_d^n), \varphi_r))$, which is the VMS-ROM closure term. We emphasize that the same ROM operator $\tilde{A}$ is used
Next, we prove that the DD-VMS-ROM is verifiable. We note that, as explained in Sec-

\[ \text{Remark 4.8.} \]

4.5. VERIFIABILITY OF THE DD-VMS-ROM

To prove that the inner product \(- (\tau^{\text{ROM}}(u^r), \varphi_i)\), \(- (\tau^{\text{ROM}}(P_r(u^n_d), \varphi_i)\) is used in the linear ansatz \(- (\tau^{\text{ROM}}(P_r(u^n_d), \varphi_i)_{i=1,..,r} \approx \tilde{A} b_r\), where \(b^n_r\) is an \(r\)-dimensional vector that contains the first \(r\) entries of the vector \(a_d^n\). The same ROM operator \(\tilde{A}\) is also used in the linear ansatz (4.23) for the VMS-ROM closure term \(- (\tau^{\text{ROM}}(u^n_r), \varphi_i)\): \(- (\tau^{\text{ROM}}(u^n_r), \varphi_i)_{i=1,..,r} \approx \tilde{A} a_r\). We approximate the VMS-ROM closure terms with these ansatzes and we obtain the following equalities:

\[
(\tau^{\text{ROM}}(P_r(u^n_d)) - \tau^{\text{ROM}}(u^n_r), \varphi_i) = \left( \tau^{\text{ROM}}(P_r(u^n_d), \varphi_i) \right) - \left( \tau^{\text{ROM}}(u^n_r, \varphi_i) \right) \\
= (- \tilde{A} b^n_r)_i - (- \tilde{A} a^n_r)_i \\
= (- \tilde{A} (b^n_r - a^n_r))_i \quad \forall i = 1, .., r. \tag{4.29}
\]

To prove that the inner product \(\tau^{\text{ROM}}(P_r(u^n_d)) - \tau^{\text{ROM}}(u^n_r), \sigma\) is non-negative, we use the definitions of \(P_r(u^n_d)\) in (4.16) and \(u^n_r\) in (4.20) and rewrite it as follows:

\[
\left( \tau^{\text{ROM}}(P_r(u^n_d)) - \tau^{\text{ROM}}(u^n_r), P_r(u^n_d) - u^n_r \right) = \left( \tau^{\text{ROM}}(P_r(u^n_d)) - \tau^{\text{ROM}}(u^n_r), \sum_{i=1}^{r} (a^n_d - a^n_r)_i, \varphi_i \right) \\
= \sum_{i=1}^{r} (a^n_d - a^n_r)_i \left( \tau^{\text{ROM}}(P_r(u^n_d)) - \tau^{\text{ROM}}(u^n_r), \varphi_i \right). \tag{4.30}
\]

By applying (4.29) to (4.30) and using the physical constraint (4.25), we get

\[
(\tau^{\text{ROM}}(P_r(u^n_d)) - \tau^{\text{ROM}}(u^n_r), P_r(u^n_d) - u^n_r) = \sum_{i=1}^{r} (a^n_d - a^n_r)_i \left( - \tilde{A} (b^n_r - a^n_r) \right)_i \\
= -(b^n_r - a^n_r)^T \tilde{A} (b^n_r - a^n_r) \geq 0, \tag{4.31}
\]

since \(\tilde{A}\) is negative semi-definite. In (4.31), we have used that \(b^n_r\) is an \(r\)-dimensional vector that contains the first \(r\) entries of the \(a^n_d\). The inequality in (4.31) concludes the proof. \(\square\)

**Remark 4.8.** We note that in Theorem 4.7 we prove the ROM mean dissipativity property only for \(P_r(u^n_d)\) and \(u^n_r\). This is contrast with the FEM context, where mean dissipativity is proven for general FEM functions (see, e.g., [17])

Next, we prove that the DD-VMS-ROM is verifiable. We note that, as explained in Section 4.3, the goal for the DD-VMS-ROM solution is to approximate as accurately as possible \(P_r(u^n_d)\), which is the large scale component of the \(d\)-dimensional G-ROM solution (4.6),
which is the “truth” solution that is employed as a benchmark in our investigation. We also note that \( P_r(\mathbf{u}_d^n) \) satisfies the LES-ROM equations (4.17), which, for clarity, we rewrite below:

\[
\left( \frac{P_r(\mathbf{u}_d^n) - P_r(\mathbf{u}_d^{n-1})}{\Delta t}, \mathbf{v}_r \right) + Re^{-1}(\nabla P_r(\mathbf{u}_d^n), \nabla \mathbf{v}_r) + (P_r(\mathbf{u}_d^n) \cdot \nabla P_r(\mathbf{u}_d^n), \mathbf{v}_r) + (\tau_{FOM}(\mathbf{u}_d^n), \mathbf{v}_r) = (\mathbf{f}^n, \mathbf{v}_r),
\]

(4.32)

where we used the fact that \( (\tau_{FOM}(\mathbf{u}_d^n), \mathbf{v}_r) \) is equal to \( (P_r(\tau_{FOM}(\mathbf{u}_d^n)), \mathbf{v}_r) \). We also rewrite the full discretization of the DD-VMS-ROM (4.21):

\[
\left( \frac{\mathbf{u}_r^n - \mathbf{u}_r^{n-1}}{\Delta t}, \mathbf{v}_r \right) + Re^{-1}(\nabla \mathbf{u}_r^n, \nabla \mathbf{v}_r) + (\mathbf{u}_r^n \cdot \nabla \mathbf{u}_r^n, \mathbf{v}_r) + (\tau_{ROM}(\mathbf{u}_r^n), \mathbf{v}_r) = (\mathbf{f}^n, \mathbf{v}_r).
\]

(4.33)

Furthermore, we use the linear ansatz (4.23) and the physical constraints (4.25) for the ROM closure model in the DD-VMS-ROM (4.33). We also choose the initial condition \( \mathbf{u}_0 = P_r(\mathbf{u}_d) \).

Thus, the DD-VMS-ROM error at time step \( n \), which we denote with \( e^n \), is defined as the difference between the large scale component of the “truth” solution, \( P_r(\mathbf{u}_d^n) \) (which is the solution of (4.32)), and the DD-VMS-ROM solution of (4.33), \( \mathbf{u}_r^n \): \( e^n = P_r(\mathbf{u}_d^n) - \mathbf{u}_r^n \).

To prove the DD-VMS-ROM’s verifiability, we use the following sharper bound on the non-linear term, which is given in Lemma 22 in [21] (see also Lemma 61.1 in [34]):

**Lemma 4.9.** Let \( \Omega \subset \mathbb{R}^d \) be an open, bounded set of class \( C^2 \), with \( q = 2 \) or \( 3 \). For all \( \mathbf{u}, \mathbf{v}, \mathbf{w} \in [H_0^1(\Omega)]^d \),

\[
b(\mathbf{u}, \mathbf{v}, \mathbf{w}) \leq C(\Omega) \sqrt{||\mathbf{u}|| \ ||\nabla \mathbf{u}|| \ ||\nabla \mathbf{v}|| \ ||\nabla \mathbf{w}||},
\]

(4.34)

where the trilinear form \( b(\cdot, \cdot, \cdot) \) [21, 35] is defined as

\[
b(\mathbf{u}, \mathbf{v}, \mathbf{w}) = (\mathbf{u} \cdot \nabla \mathbf{v}, \mathbf{w}).
\]

(4.35)

**Theorem 4.10.** The DD-VMS-ROM (4.33) with linear ansatz (4.23), physical constraint (4.25), and the initial condition \( \mathbf{u}_r^0 = P_r(\mathbf{u}_d^0) \) is verifiable: For a small enough time step, \( \Delta t d_j < 1, \forall j = 1, ..., M \), where \( d_j = \left( \frac{3ReC(\Omega)^2}{4} ||\nabla P_r(\mathbf{u}_d^j)||^4 + Re \right) \) and \( C(\Omega) \) is the constant
Applying Hölder’s and Young’s inequalities to the terms \(\frac{d_j}{1-\Delta t d_j}\) from (4.39), we have the following inequality:

\[
\frac{d_j}{1-\Delta t d_j} \leq \frac{1}{2} \frac{d_j}{1-\Delta t d_j} + \frac{1}{2} \frac{d_j}{1-\Delta t d_j}.
\]

\[
\frac{d_j}{1-\Delta t d_j} \leq \frac{1}{2} \frac{d_j}{1-\Delta t d_j} + \frac{1}{2} \frac{d_j}{1-\Delta t d_j}.
\]

Then by applying (4.39) to (4.38), we get the following inequality:

\[
\Delta t^{-1} (e^j - e^{j-1}, e^j) + Re^{-1} \|\nabla e^j\|^2 + b(e^j, P_r(u^j_d), e^j)
\]

\[
+ (\tau^{ROM}(P_r(u^j_d)) - \tau^{ROM}(u^j_r), e^j) = -(\tau^{FOM}(u^j_d) - \tau^{ROM}(P_r(u^j_d)), e^j).
\]

From Theorem 4.7, we have the following inequality:

\[
(\tau^{ROM}(P_r(u^j_d)) - \tau^{ROM}(u^j_r), e^j) \geq 0.
\]

Then by applying (4.39) to (4.38), we get the following inequality:

\[
\Delta t^{-1} (e^j - e^{j-1}, e^j) + Re^{-1} \|\nabla e^j\|^2 \leq -b(e^j, P_r(u^j_d), e^j) - (\tau^{FOM}(u^j_d) - \tau^{ROM}(P_r(u^j_d), e^j))
\]

Applying Hölder’s and Young’s inequalities to the terms \((e^j - e^{j-1}, e^j)\) and \(-(\tau^{FOM}(u^j_d) - \tau^{ROM}(P_r(u^j_d), e^j))\) in (4.40) we obtain that, for any \(C_1, C_2 > 0\), the following inequalities hold:

\[
(e^j - e^{j-1}, e^j) = \|e^j\|^2 - (e^j, e^{j-1})
\]

\[
\geq \|e^j\|^2 - \|e^j\| \|e^{j-1}\|
\]

\[
\geq \|e^j\|^2 - \frac{C_1}{2} \|e^j\|^2 - \frac{1}{2C_1} \|e^{j-1}\|^2
\]
and

\[ \left| - (\boldsymbol{\tau}^{FOM}(\mathbf{u}_d^j) - \boldsymbol{\tau}^{ROM}(P_r(\mathbf{u}_d^j)), e^j) \right| = \left| - (P_r(\boldsymbol{\tau}^{FOM}(\mathbf{u}_d^j) - \boldsymbol{\tau}^{ROM}(P_r(\mathbf{u}_d^j)), e^j) \right| \leq \frac{1}{2C_2} \| P_r(\boldsymbol{\tau}^{FOM}(\mathbf{u}_d^j) - \boldsymbol{\tau}^{ROM}(P_r(\mathbf{u}_d^j))) \|^2 + \frac{C_2}{2} \| e^j \|^2. \]  

(4.42)

Applying Lemma 4.9 to the term \(-b(e^j, P_r(\mathbf{u}_d^j), e^j)\) we obtain the following inequality for any \(C_3 > 0\):

\[ \left| - b(e^j, P_r(\mathbf{u}_d^j), e^j) \right| \leq C(\Omega) \| \nabla e^j \|^3 \| \nabla P_r(\mathbf{u}_d^j) \| \| e^j \|^{1/2} \]
\[ \leq \frac{3C_3C(\Omega)}{4} \| \nabla e^j \|^2 + \frac{C(\Omega)}{4C_3} \| \nabla P_r(\mathbf{u}_d^j) \|^4 \| e^j \|^2, \]  

(4.43)

where \(C(\Omega)\) is the constant in Lemma 4.9.

By choosing \(C_1 = 1\), \(C_2 = Re\), and \(C_3 = 2Re^{-1}/3C(\Omega)\), we get the following inequality:

\[ \frac{1}{2\Delta t}(\| e^j \|^2 - \| e^{j-1} \|^2) + \frac{Re^{-1}}{2} \| \nabla e^j \|^2 \]
\[ \leq \left( \frac{3ReC(\Omega)^2}{8} \| \nabla P_r(\mathbf{u}_d^j) \|^4 + \frac{Re}{2} \right) \| e^j \|^2 + \frac{Re^{-1}}{2} \| P_r(\boldsymbol{\tau}^{FOM}(\mathbf{u}_d^j) - \boldsymbol{\tau}^{ROM}(P_r(\mathbf{u}_d^j))) \|^2. \]  

(4.44)

By multiplying (4.44) by \(2\Delta t\) and summing the resulting inequalities from \(j = 1\) to \(n\), we obtain the following inequality:

\[ \| e^n \|^2 + \Delta t \sum_{j=1}^{n} Re^{-1} \| \nabla e^j \|^2 \leq \| e^0 \|^2 + \Delta t \sum_{j=1}^{n} \left( \frac{3ReC(\Omega)^2}{4} \| \nabla P_r(\mathbf{u}_d^j) \|^4 + Re \right) \| e^j \|^2 \]
\[ + \Delta t \sum_{j=1}^{n} Re^{-1} \| P_r(\boldsymbol{\tau}^{FOM}(\mathbf{u}_d^j) - \boldsymbol{\tau}^{ROM}(P_r(\mathbf{u}_d^j))) \|^2. \]  

(4.45)
4.5. Verifiability of the DD-VMS-ROM

To apply the discrete Gronwall’s lemma, we first make the following notation:

\[
\begin{align*}
    a_j &:= \| \mathbf{e}^j \|^2 \geq 0, \\
    b_j &:= Re^{-1} \| \nabla \mathbf{e}^j \|^2 \geq 0, \\
    d_j &:= \left( \frac{3ReC(\Omega)^2}{4} \| \nabla P_r(u_d^j) \|^4 + Re \right) \geq 0, \\
    c_j &:= Re^{-1} \| P_r(\tau^FOM(u_d^j) - \tau^ROM(P_r(u_d^j))) \|^2 \geq 0, \\
    H &:= \| \mathbf{e}^0 \|^2 \geq 0.
\end{align*}
\] (4.46)

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 (4.46), we rewrite (4.45) as follows:

\[
a_n + \Delta t \sum_{j=1}^{n} b_j \leq \Delta t \sum_{j=1}^{n} d_j a_j + \Delta t \sum_{j=1}^{n} c_j + H.
\] (4.47)

By using the discrete Gronwall’s lemma (see Lemma 27 in [21]) in (4.47), we obtain the following inequality:

\[
a_n + \Delta t \sum_{j=1}^{n} b_j \leq \exp \left( \Delta t \sum_{j=1}^{n} \frac{d_j}{\Delta t d_j} \right) \left( \Delta t \sum_{j=1}^{n} c_j + H \right).
\] (4.48)

(We note that choosing the initial condition \( \mathbf{u}_r^0 = P_r(u_d^0) \), implies that \( \mathbf{e}^0 = \mathbf{u}_r^0 - P_r(u_d^0) \) and \( H = 0 \).) The inequality (4.48) proves (4.36).

\[\square\]

Remark 4.11. 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 [21]).

Remark 4.12. 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.
CHAPTER 4. VERIFIABILITY OF THE DD-VMS-ROM

4.6 Numerical Results

In Theorem 4.10, we proved that the DD-VMS-ROM presented in Section 4.4 is verifiable. In this section, we present numerical support for the theoretical results in Theorem 4.10. In Section 4.6.1, 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 4.6.3 and for the flow past a cylinder in Section 4.6.4.

4.6.1 Numerical Implementation

“Truth” Solution For computational efficiency, instead of solving the very large-dimensional G-ROM (4.5) to get the “truth” solution, \( u_d \), we simply project the FOM data on the ROM space, i.e., \( u_d = P_r(u_h), r = d \). 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 [29], the least squares problem (4.24) 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 (4.26) is ill conditioned. To alleviate the ill conditioning of the least squares problem, we proposed the use of the truncated SVD [25, 38] (see also [3] for a related approach). For completeness, in Algorithm 5, we outline the construction of the DD-VMS-ROM with the truncated SVD procedure.

The tolerance \( tol \) specified in step 3 of Algorithm 5 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 \( \tilde{A} \) 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 \( \tilde{A} \), but does not significantly alleviate the ill conditioning of the least squares problem either. In our numerical investigation, a careful
choice of the tolerance $\text{tol}$ yields optimal DD-VMS-ROM results.

**Algorithm 5:** Truncated SVD in Solving Least Square Problem

1: Formulate the standard linear least square problem for the unknown vector $\mathbf{x}_u$:

$$\min_{\mathbf{x}_u} \| E \mathbf{x}_u - \mathbf{f} \|^2, \quad (4.49)$$

where $E \in \mathbb{R}^{Mr \times r^2}$ is a matrix whose entries are determined by $\mathbf{a}_d(t_j), j = 1, \cdots, M$,
$f \in \mathbb{R}^{Mr \times 1}$ is a vector whose entries are determined by $P_r(\mathbf{r}^{FOM}(t_j))$, and
$x_u \in \mathbb{R}^{r^2 \times 1}, j = 1, \cdots, M$ is a vector whose entries are determined by $\tilde{\mathbf{A}}$.

2: Calculate the SVD of $E$:

$$E = U \Sigma V^\top. \quad (4.50)$$

3: Specify a tolerance $\text{tol}$.

4: Keep the entries in $\Sigma$ that are larger than $\text{tol}$; the result matrix is $\tilde{\Sigma}$ ($\tilde{\sigma} = \sigma$ if $\sigma > \text{tol}$;
also the singular values of $E$ can be chosen as a $\text{tol}$).

5: Construct the truncated SVD of $E$, $\tilde{E}$:

$$\tilde{E} = \tilde{U} \tilde{\Sigma} \tilde{V}^\top, \quad (4.51)$$

where $\tilde{U}$ and $\tilde{V}$ are the entries of $U, V$ that correspond to $\tilde{\Sigma}$.

6: The solution is given by

$$\mathbf{x}_u = \left( \tilde{V} \tilde{\Sigma}^{-1} \tilde{U}^\top \right) \mathbf{f}. \quad (4.52)$$

**Time Discretization** Although the DD-VMS-ROM’s verifiability was proven in Theorem 4.10 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 4.6.3) and the linearized BDF2 for the flow past a cylinder (Section 4.6.4). 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 in-
vestigation in this section. Furthermore, we believe that the mathematical arguments used to prove the DD-VMS-ROM’s verifiability in Theorem 4.10 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 4.10, 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 \( \mathbf{u}_d \) (i.e., the solution of the \( d \)-dimensional G-ROM (4.5)) is also fixed. The goal of our numerical investigation is to show that, for fixed \( M, d \), and \( \mathbf{u}_d \), there exists a constant \( C \) such that for varying \( r \) values and for varying \( tol \) values, the inequality (4.36) 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 (4.36), we use the following average \( L^2 \) norm:

\[
\mathcal{E}(L^2) = \frac{1}{M} \sum_{n=1}^{M} \| P_r(\mathbf{u}^n_d) - \mathbf{u}^n_r \|^2 = \frac{1}{M} \sum_{n=1}^{M} \| e^n \|^2 .
\]  

(4.53)

To quantify the LES-ROM closure error, i.e., the term on the RHS of inequality (4.36), we use the following metric:

\[
\eta(L^2) = \frac{1}{M} \sum_{n=1}^{M} \| P_r(\tau^{FOM}(\mathbf{u}^n_d)) - \tau^{ROM}(P_r(\mathbf{u}^n_d)) \|_{L^2}^2 .
\]  

(4.54)

4.6.2 Assessment of Results

To illustrate numerically the DD-VMS-ROM verifiability proven in Theorem 4.10, we need to show that, for varying \( r \) values, as \( \eta(L^2) \) in (4.54) decreases, so does \( \mathcal{E}(L^2) \) in (4.53). 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 (4.54). 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 (4.53). 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 $\mathcal{E}(L^2)$.

4.6.3 Burgers Equation

In this section, we investigate the DD-VMS-ROM verifiability in the numerical simulation of the one-dimensional viscous Burgers equation:

\[
\begin{align*}
  u_t - \nu u_{xx} + uu_x &= 0, \quad x \in [0, 1], \quad t \in [0, 1], \\
  u(0, t) &= u(1, t) = 0, \quad t \in (0, 1], \tag{4.55} \\
  u(x, 0) &= u_0(x), \quad x \in [0, 1],
\end{align*}
\]

with non-smooth initial condition (4.56):

\[
u_0(x) = \begin{cases} 
  1, & x \in (0, 1/2], \\
  0, & x \in (1/2, 1].
\end{cases} \tag{4.56}
\]

This test problem has been used in [1, 12, 20, 38].

**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 4.1, for three different $r$ values, we list $\mathcal{E}(L^2)$ in (4.53), which measures the DD-VMS-ROM error, and $\eta(L^2)$ in (4.54), 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 4.1, for $r = 3, 7, 11$, generally show that, as $\eta(L^2)$ decreases, so does $\mathcal{E}(L^2)$.

In Figure 4.1, 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 4.1 and Figure 4.1 support the theoretical results in Theorem 4.10.

<table>
<thead>
<tr>
<th>$r = 3$</th>
<th>$r = 7$</th>
<th>$r = 11$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\eta(L^2)$</td>
<td>$\mathcal{E}(L^2)$</td>
<td>$\eta(L^2)$</td>
</tr>
<tr>
<td>2.047e-01</td>
<td>1.131e-02</td>
<td>5.597e-01</td>
</tr>
<tr>
<td>2.040e-01</td>
<td>1.121e-02</td>
<td>5.600e-01</td>
</tr>
<tr>
<td>2.032e-01</td>
<td>1.109e-02</td>
<td>5.606e-01</td>
</tr>
<tr>
<td>1.976e-01</td>
<td>1.048e-02</td>
<td>5.543e-01</td>
</tr>
<tr>
<td>1.912e-01</td>
<td>9.150e-03</td>
<td>5.452e-01</td>
</tr>
<tr>
<td>1.596e-01</td>
<td>4.203e-03</td>
<td>4.933e-01</td>
</tr>
<tr>
<td>1.158e-01</td>
<td>2.123e-03</td>
<td>2.628e-01</td>
</tr>
</tbody>
</table>

Table 4.1: Burgers equation (4.55), reconstructive regime: $\mathcal{E}(L^2)$ and $\eta(L^2)$ values for fixed $r$ values and different tolerance values in the truncated SVD.

4.6.4 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., [23, 25, 38].

Computational Setting As a mathematical model, we use the NSE (4.1)–(4.2). The computational domain is a $2.2 \times 0.41$ rectangular channel with a radius $= 0.05$ cylinder, centered at $(0.2, 0.2)$, see Figure 4.2.
4.6. Numerical Results

Figure 4.1: Burgers equation (4.55), 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.

Figure 4.2: 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 [15, 23, 24, 30]:

\begin{align*}
  u_1(0, y, t) &= u_1(2.2, y, t) = \frac{6}{0.41^2} y(0.41 - y), \\
  u_2(0, y, t) &= u_2(2.2, y, t) = 0,
\end{align*}  

(4.57)  
(4.58)
where \( \mathbf{u} = (u_1, u_2) \). 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 [16]. The mesh yields 103\(K\) (102962) velocity and 76\(K\) (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 \( \text{Re} = 100 \), after \( t = 5s \) and for \( \text{Re} = 1000 \), 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 \( \text{Re} = 100 \), from \( t = 7 \) to \( t = 17 \) and for \( \text{Re} = 1000 \), from \( t = 13 \) to \( t = 23 \).

To train the DD-VMS-ROM closure operator \( \hat{A} \), 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 \( \text{Re} = 100 \) and from \( t = 13 \) to \( t = 13.268 \) for \( \text{Re} = 1000 \). Thus, we collect 167 snapshots for \( \text{Re} = 100 \) and 135 snapshots for \( \text{Re} = 1000 \).

**Numerical Results for \( \text{Re} = 100 \)**

In Table 4.2, for three different \( r \) values, we list \( \mathcal{E}(L^2) \) in (4.53), which measures the DD-VMS-ROM error, and \( \eta(L^2) \) in (4.54), 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 4.2, for \( r = 4, 6, \) and 8, generally show that, as \( \eta(L^2) \) decreases, so does \( \mathcal{E}(L^2) \).

In Figure 4.3, 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 4.2 and Figure 4.3 support the theoretical results in Theorem 4.10.

Table 4.2: 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.

<table>
<thead>
<tr>
<th>$r$</th>
<th>$\eta(L^2)$</th>
<th>$\mathcal{E}(L^2)$</th>
<th>$\eta(L^2)$</th>
<th>$\mathcal{E}(L^2)$</th>
<th>$\eta(L^2)$</th>
<th>$\mathcal{E}(L^2)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>3.545e+03</td>
<td>1.231e-01</td>
<td>5.571e-01</td>
<td>4.186e-03</td>
<td>8.361e-01</td>
<td>4.334e-03</td>
</tr>
<tr>
<td>6</td>
<td>9.360e+02</td>
<td>8.742e-02</td>
<td>3.261e-01</td>
<td>3.223e-03</td>
<td>6.380e-01</td>
<td>4.062e-03</td>
</tr>
<tr>
<td>8</td>
<td>1.126e+01</td>
<td>1.536e-02</td>
<td>3.642e-02</td>
<td>1.323e-03</td>
<td>2.393e-02</td>
<td>2.368e-03</td>
</tr>
<tr>
<td></td>
<td>3.094e-02</td>
<td>4.021e-03</td>
<td>1.507e-03</td>
<td>1.837e-04</td>
<td>5.515e-03</td>
<td>2.593e-04</td>
</tr>
<tr>
<td></td>
<td>1.379e-02</td>
<td>9.043e-04</td>
<td>5.862e-05</td>
<td>5.407e-06</td>
<td>2.586e-03</td>
<td>3.106e-05</td>
</tr>
<tr>
<td></td>
<td>2.549e-04</td>
<td>2.313e-04</td>
<td>5.615e-05</td>
<td>5.382e-06</td>
<td>2.144e-04</td>
<td>4.229e-06</td>
</tr>
</tbody>
</table>

Figure 4.3: 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 4.3, for three different $r$ values, we list $\mathcal{E}(L^2)$ in (4.53), which measures the DD-VMS-ROM error, and $\eta(L^2)$ in (4.54), 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 4.3, for $r = 4, 6,$ and $8$, generally show that, as $\eta(L^2)$ decreases, so does $\mathcal{E}(L^2)$. 
In Figure 4.4, for $r = 4, 6, \text{ and } 8$, 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 4.3 and Figure 4.4 support the theoretical results in Theorem 4.10, which is identical to the conclusion in Section 4.6.4.

<table>
<thead>
<tr>
<th></th>
<th>$r = 4$</th>
<th></th>
<th>$r = 6$</th>
<th></th>
<th>$r = 8$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\eta(L^2)$</td>
<td>$\mathcal{E}(L^2)$</td>
<td>$\eta(L^2)$</td>
<td>$\mathcal{E}(L^2)$</td>
<td>$\eta(L^2)$</td>
<td>$\mathcal{E}(L^2)$</td>
</tr>
<tr>
<td>3.679e+02</td>
<td>4.427e-01</td>
<td>3.265e+00</td>
<td>2.138e-01</td>
<td>2.684e+01</td>
<td>2.585e-02</td>
</tr>
<tr>
<td>1.966e+00</td>
<td>3.315e-01</td>
<td>1.927e+00</td>
<td>1.840e-01</td>
<td>7.423e+00</td>
<td>1.519e-02</td>
</tr>
<tr>
<td>1.757e+00</td>
<td>3.743e-02</td>
<td>1.017e+00</td>
<td>1.095e-01</td>
<td>1.550e+00</td>
<td>9.833e-03</td>
</tr>
<tr>
<td>7.400e-01</td>
<td>2.636e-01</td>
<td>9.313e-02</td>
<td>2.291e-02</td>
<td>2.586e-01</td>
<td>4.936e-03</td>
</tr>
<tr>
<td>5.783e-01</td>
<td>2.041e-02</td>
<td>5.425e-02</td>
<td>2.451e-03</td>
<td>9.122e-02</td>
<td>2.085e-03</td>
</tr>
<tr>
<td>4.991e-02</td>
<td>1.448e-03</td>
<td>2.899e-02</td>
<td>1.889e-04</td>
<td>3.382e-02</td>
<td>1.760e-04</td>
</tr>
</tbody>
</table>

Table 4.3: 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.

Figure 4.4: 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.

### 4.7 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 [25, 38]) 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., [5]) and extend them to a ROM setting.
Bibliography


Chapter 5

Hybrid Data-Driven Reduced Order Models§

ABSTRACT OF CHAPTER 5

We propose a new hybrid reduced order model (ROM) framework for the numerical simulation of fluid flows. This hybrid framework incorporates two closure modeling strategies: (i) A structural closure modeling component that involves the recently proposed data-driven variational multiscale ROM approach, and (ii) A functional closure modeling component that introduces an artificial viscosity term. We also utilize physical constraints for the structural ROM operators in order to add robustness to the hybrid ROM. We perform a numerical investigation of the hybrid ROM for the three-dimensional turbulent channel flow at a Reynolds number $Re = 13,750$.

5.1 Introduction

Reduced order model (ROM) is a popular computational approach in numerical simulation of fluids, which can decrease the computational cost dramatically and provide a relative accurate scenario. For most ROMs, an offline stage is performed using the full-order model (FOM) data, i.e., to obtain the ROM basis functions and pre-compute the ROM operators. Then in an online stage, these ROM operators, i.e. in a low-dimensional subspace of FOM, are repeatedly used to perform online computation with different parameter settings.

In fluid dynamics, the projection ROMs (Proj-ROMs) are widely used in approximating the

§The material in this chapter is based on the following unpublished report: C. MOU, H. LIU, L. G. REBHOLZ, AND T. ILIESCU, Hybrid data-driven reduced order models for 3d turbulent channel flows. In preparation
5.1. Introduction

incompressible Navier-Stokes equation (NSE) with forcing:

\[
\frac{\partial \mathbf{u}}{\partial t} - \frac{1}{Re} \Delta \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla p = \mathbf{f}, \tag{5.1}
\]

\[
\nabla \cdot \mathbf{u} = 0, \tag{5.2}
\]

where \( \mathbf{u} = [u, v]^T \) (or \( \mathbf{u} = [u, v, w]^T \) in the three-dimensional case) is the velocity vector field, \( p \) the pressure, \( Re \) the Reynolds number, and \( \mathbf{f} \) the forcing term. The Proj-ROM framework often yields the following steps: (i). select a set of dominant modes as basis, i.e., \( \{\varphi_1, \cdots, \varphi_R\} \), (ii). replace \( \mathbf{u} \) with a low-dimensional representation of the solution \( \mathbf{u}_r \) (\( r \) is small) in (5.1) and (5.2); and (iii). use a Galerkin projection to project the NSE (5.1) and (5.2) onto a low dimensional subspace span\{\( \varphi_1, \cdots, \varphi_r \)\}, and thus obtain the Proj-ROM \([10, 19, 24, 25]\).

To increase the accuracy of Proj-ROMs, one strategy is to apply the closure modeling. Closure problems originate from the large eddy simulation (LES) for turbulence in the CFD community. The LES closure models are mostly classified into two categories \([3, 7, 18]\): functional closure models and structural closure models. The functional closure models attempt to capture the physical effects of the closure terms, e.g., by adapting the Kolmogorov’s theory of turbulence and the energy cascade to dissipate energy from the system. The Smagorinsky model is one famous example in this category \([20]\). The structure closure models, on the other hand, try to extract the closure term representation through mathematical derivations, for example, the approximate deconvolution (AD) model uses the truncated series expansion to approximate the nonfiltered field \([19, 21, 22, 23, 27]\).

The recently proposed data-driven variational Multiscale ROM (DD-VMS-ROM) \([12, 14, 26]\) (see Chapter 2) can be categorized as a closure model. Specifically, the DD-VMS-ROM employs the data-driven and variational multiscale techniques to model the closure terms for the Proj-ROM. The data-driven approach, which relies on solving an optimization problem (i.e., the unconstrained least square problem in \([26]\)) is a structural ROM closure modeling approach. However, this approach can be inaccurate for some high Reynolds number flows (e.g., flows that contain many coherent structures). To improve the physical accuracy of the DD-VMS-ROM, one may consider the physically constrained DD-VMS-ROM scheme, which was introduced in \([12]\). In this approach, instead of the unconstrained least square problem, a physically constrained optimization problem is solved to obtain the DD-VMS operators for the closure term. Admittedly, the data-driven closure modeling falls into
the structural modeling category; imposing constraints in the data-driven approach allows
options for closure operators to be in either a functional or a structural form: For example,
in [12], a structural type of constraints was chosen for the closure modeling.

In the CFD community, the three-dimensional turbulent channel flow is well recognized as a
challenging numerical problem [6, 8, 13], which shares high Reynolds number flow properties,
e.g., the turbulent kinetic energy is produced and dissipated approximately in balance in the
log layer. Numerical solutions often fail due to the lack of physical dissipation of energy
from the smallest resolved scales to the unresolved scales. To build up an accurate closure
ROM solution for such a problem is challenging. One simple approach to achieve a stable
ROM is adding artificial viscosity [1, 2, 4, 5, 9] to the standard Galerkin ROM although it
may compromise the accuracy for lacking the ROM closure counterparts. Therefore, to get
a robust ROM for three-dimensional channel flow requires a treatment that can stabilize the
ROM easily and improve the accuracy at the same time. One may consider combining the
functional and the structural closure modeling approaches.

In this chapter, we propose a new class of hybrid data-driven ROMs which incorporate
both functional closure modeling, i.e., an artificial viscosity approach, and structural closure
modeling, i.e., the DD-VMS approach. In addition, we test both the functional and the
structural options for physical constraints imposed in the DD-VMS operators in the hybrid
ROM framework. The new hybrid framework allows the modular construction of the closure
term based on the known physics of the fluid model. For example, one can add more
functional closure into the hybrid ROM if one knows that the flow model requires more
dissipation. The numerical results show that for the three-dimensional turbulent channel
flow, the hybrid ROM is much more accurate in capturing the flow’s physical properties,
i.e., kinetic energy and second-order statistics, compared with the AV-G-ROM and the DD-
VMS-ROM. Furthermore, applying functional type of constraints in the DD-VMS operators
can further improve the performance of the hybrid ROM.

The rest of the chapter is organized as follows: In Section 5.2, we briefly introduce the
G-ROM’s simple variant, G-ROM with an artificial viscosity (AV-G-ROM). In Section 5.3,
we briefly describe the data-driven variational multiscale ROMs (DD-VMS-ROMs) and the
constrained data-driven variational multiscale ROMs (CDD-VMS-ROMs). In Section 5.4,
we describe the new hybrid data-driven ROMs. In Section 5.5, we test the hybrid data-driven
ROMs in the numerical simulation of the three-dimensional channel flow.
In the following sections, we will discuss different types of ROMs; in Table 5.1, we list the *acronyms* of ROMs as they help make this chapter more concise.

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>G-ROM</td>
<td>Galerkin ROM</td>
</tr>
<tr>
<td>AV-G-ROM</td>
<td>Galerkin ROM with an artificial viscosity</td>
</tr>
<tr>
<td>2S-DD-VMS-ROM</td>
<td>Two-scale data-driven variational multiscale ROM</td>
</tr>
<tr>
<td>3S-DD-VMS-ROM</td>
<td>Three-scale data-driven variational multiscale ROM</td>
</tr>
<tr>
<td>2S-CDD-VMS-ROM</td>
<td>Physically constrained two-scale data-driven variational multiscale ROM</td>
</tr>
<tr>
<td>3S-CDD-VMS-ROM</td>
<td>Physically constrained three-scale data-driven variational multiscale ROM</td>
</tr>
<tr>
<td>2S-H-DD-VMS-ROM</td>
<td>2S-DD-VMS-ROM with an artificial viscosity</td>
</tr>
<tr>
<td>3S-H-DD-VMS-ROM</td>
<td>3S-DD-VMS-ROM with an artificial viscosity</td>
</tr>
<tr>
<td>2S-CH-DD-VMS-ROM</td>
<td>Physically constrained 2S-DD-VMS-ROM with an artificial viscosity</td>
</tr>
<tr>
<td>3S-CH-DD-VMS-ROM</td>
<td>Physically constrained 3S-DD-VMS-ROM with an artificial viscosity</td>
</tr>
</tbody>
</table>

Table 5.1: Acronyms for different ROMs
5.2 Galerkin ROM with an Artificial Viscosity

Often the standard G-ROM suffers an unphysical increase in energy time evolution. One simple remedy would be adding an artificial viscosity term. Following [5], we can add an artificial viscosity term \( \nu_T \nabla u_r, \nabla \varphi_i \) to the G-ROM (2.10) in Chapter 2 and obtain the following:

\[
\frac{\partial u_r}{\partial t}, \varphi_i \rightleftharpoons (u_r \cdot \nabla) u_r, \varphi_i + \nu_T \nabla u_r, \nabla \varphi_i + \nu_T \nabla u_r, \nabla \varphi_i = (f, \varphi_i),
\]  
\( (5.3) \)

where \( i = 1, \ldots, r \), and \( \nu_T \) is a constant parameter. We denote the resulting ROM as artificial viscosity Galerkin ROM (AV-G-ROM). In the discrete form, to distinguish the matrix from the artificial viscosity, the AV-G-ROM yields the following:

\[
\dot{a} = b + (A + \tilde{A}_T)a + a^T B a,
\]  
\( (5.4) \)

where \( b, A, \) and \( B \) correspond to the constant, linear, and quadratic terms in G-ROM (1.20), and \( \tilde{A}_T \) corresponds to the linear term of artificial viscosity, which yields the following pointwise form:

\[
(A_T)_{im} = -\nu_T \nabla \varphi_i, \nabla \varphi_m.
\]  
\( (5.5) \)

5.3 DD-VMS-ROM

In Chapter 2 (also see [14]), a data-driven based variational multiscale (VMS) ROM framework is proposed. In Sections 5.3.1 and 5.3.2, we briefly review the closure modeling used to build 2S-DD-VMS-ROM and the 3S-DD-VMS-ROM.
5.3. DD-VMS-ROM

5.3.1 2S-DD-VMS-ROM

In the 2S-DD-VMS-ROM construction (see Section 2.2.3 for details), we approximate the VMS-ROM closure term (the boxed term in equation (2.12)) with a generic function $g(u_r)$:

\[
(\tau)_i := -\left[\left( (u_R \cdot \nabla) u_R , \varphi_i \right) - \left( (u_r \cdot \nabla) u_r , \varphi_i \right) \right] \\
\approx \left( g(u_r) , \varphi_i \right), \quad \forall i = 1, \ldots, r. \tag{5.6}
\]

The coefficients/parameters of $g(u_r)$ are determined by solving the following low-dimensional least squares problem in the offline stage:

\[
\min_{\text{g parameters}} \sum_{j=1}^{M} \left\| \tau_{FOM}(t_j) - g(a_{FOM}(t_j)) \right\|^2,
\tag{5.7}
\]

where $\tau_{FOM}(t_j)$ and $a_{FOM}(t_j)$ are obtained from the FOM data and $M$ is the number of snapshots.

After the least squares problem (5.7) for the generic function $g(a)$ is solved, we obtain the 2S-DD-VMS-ROM:

\[
\dot{a} = b + Aa + a^T B a + g(a), \tag{5.8}
\]

where $b$, $A$, and $B$ are the G-ROM operators in (1.20).

5.3.2 3S-DD-VMS-ROM

In the 3S-DD-VMS-ROM construction (see Section 2.2.4 for details), with more flexibility, we can approximate the two VMS-ROM closure terms in equations (2.22) and (2.23) with two different generic functions,

\[
(\tau_L)_i := -\left[\left( (u_R \cdot \nabla) u_R , \varphi_i \right) - \left( (u_L + u_S) \cdot \nabla) (u_L + u_S) , \varphi_i \right) \right] \\
\approx \left( g_L(u_r) , \varphi_i \right), \quad \forall i = 1, \ldots, r_1 , \tag{5.9}
\]

\[
(\tau_S)_i := -\left[\left( (u_R \cdot \nabla) u_R , \varphi_i \right) - \left( (u_L + u_S) \cdot \nabla) (u_L + u_S) , \varphi_i \right) \right] \\
\approx \left( g_S(u_r) , \varphi_i \right) \quad \forall i = r_1 + 1, \ldots, r. \tag{5.10}
\]
To determine the entries in $g_L$ and $g_S$, we solve two least squares problems:

\[
\begin{align*}
\min_{g_L \text{ parameters}} & \sum_{j=1}^{M} \| \tau_{L}^{FOM}(t_j) - g_L(a^{FOM}(t_j)) \|^2, \\
\min_{g_S \text{ parameters}} & \sum_{j=1}^{M} \| \tau_{S}^{FOM}(t_j) - g_S(a^{FOM}(t_j)) \|^2,
\end{align*}
\]

(5.11) \hspace{1cm} (5.12)

where $\tau_{L}^{FOM}(t_j)$, $\tau_{S}^{FOM}(t_j)$, and $a^{FOM}(t_j)$ are obtained from the FOM data and $M$ is the number of snapshots.

After the least square problems (5.11) and (5.12) for the generic function $g_L(a)$ and $g_S(a)$ are solved, we obtain the 3S-DD-VMS-ROM:

\[
\begin{bmatrix}
\dot{a}_L \\
\dot{a}_S
\end{bmatrix} = b + A a + a^\top B a + \begin{bmatrix}
g_L(a) \\
g_S(a)
\end{bmatrix},
\]

(5.13)

where $b$, $A$, and $B$ are the G-ROM operators in (1.20).

### 5.3.3 Choices of generic functions

In this chapter, we consider two different choices of the generic functions in 2S-DD-VMS-ROM (5.8), i.e., $g(a)$, and in 3S-DD-VMS-ROM (5.13), i.e., $g_L(a)$ and $g_S(a)$.

**A linear ansatz**

We first consider that the generic functions arisen in DD-VMS-ROM take the linear form, i.e., a linear ansatz. In particular, for the 2S-DD-VMS-ROM, we assume that $g(a)$ is of the following form:

\[
g(a) = \tilde{A} a,
\]

(5.14)
where $\tilde{A} \in \mathbb{R}^{r \times r}$ is the VMS-ROM closure operator constructed in (5.7). Similarly, for the 3S-DD-VMS-ROM, we assume that $g_L(a)$ and $g_S(a)$ are of the following form:

$$g_L(a) = \tilde{A}_L a,$$

and

$$g_S(a) = \tilde{A}_S a,$$

(5.15)

where $\tilde{A}_L \in \mathbb{R}^{r_1 \times r}$ and $\tilde{A}_S \in \mathbb{R}^{(r-r_1) \times r}$ are the VMS-ROM closure operators constructed in (5.11) and (5.12).

A quadratic ansatz

To further improve the accuracy in approximating the closure terms, we assume that the structures of $g$ and $f$ are similar, i.e., the generic functions yield a quadratic ansatz. In particular, for the 2S-DD-VMS-ROM, we assume that $g(a)$ has the following form:

$$g(a) = \tilde{A} a + a^\top \tilde{B} a,$$

(5.16)

where $\tilde{A} \in \mathbb{R}^{r \times r}$ and $\tilde{B} \in \mathbb{R}^{r \times r}$ are the VMS-ROM closure operators constructed in (5.7). Similarly, for the 3S-DD-VMS-ROM, we assume that $g_L(a)$ and $g_S(a)$ have the following forms:

$$g_L(a) = \tilde{A}_L a + a^\top \tilde{B}_L a,$$

and

$$g_S(a) = \tilde{A}_S a + a^\top \tilde{B}_S a,$$

(5.17)

where $\tilde{A}_L \in \mathbb{R}^{r_1 \times r}$, $\tilde{B}_L \in \mathbb{R}^{r_1 \times r}$, $\tilde{A}_S \in \mathbb{R}^{(r-r_1) \times r}$, and $\tilde{B}_S \in \mathbb{R}^{(r-r_1) \times r}$ are the VMS-ROM closure operators constructed in (5.11) and (5.12).

5.3.4 Constrained DD-VMS-ROM

To improve the physical accuracy in modeling the closure term in 2S-DD-VMS-ROM, we can require the generic functions in DD-VMS-ROMs to satisfy certain constraints [12]. For example, we can add constraints in (5.7) for 2S-DD-VMS-ROM and rewrite the low-dimensional least squares problem as follows:

$$\min_{g \text{ parameters}} \sum_{j=1}^{M} \left\| \tau^{FOM}(t_j) - g(a^{FOM}(t_j)) \right\|^2,$$

(5.18)
where \( \tau^{FOM}(t_j) \) and \( a^{FOM}(t_j) \) are obtained from the FOM data and \( M \) is the number of snapshots. We denote the resulting ROM as two-scale constrained DD-VMS-ROM, abbreviated as 2S-CDD-VMS-ROM.

Similarly, we can rewrite the low-dimensional least squares problems (5.11) and (5.12) for the 3S-DD-VMS-ROM as follows:

\[
\begin{align*}
\min_{g_L \text{ parameters}} & \quad \sum_{j=1}^{M} \left\| \tau^{FOM}_L(t_j) - g_L(a^{FOM}(t_j)) \right\|^2, \\
\min_{g_S \text{ parameters}} & \quad \sum_{j=1}^{M} \left\| \tau^{FOM}_S(t_j) - g_S(a^{FOM}(t_j)) \right\|^2,
\end{align*}
\]

where \( \tau^{FOM}_L(t_j) \), \( \tau^{FOM}_S(t_j) \), and \( a^{FOM}(t_j) \) are obtained from the FOM data and \( M \) is the number of snapshots. We denote the resulting ROM as three-scale constrained DD-VMS-ROM, abbreviated as 3S-CDD-VMS-ROM.

There are two strategies to impose the constraints in the least square problems (5.18), (5.19), and (5.20):

- Functional approach: We use the training data and require that the generic functions satisfy some functional properties, e.g., energy dissipation. For example, in the 2S-DD-VMS-ROM, we suppose \( a^\top g(a) < 0 \).

- Structural approach: We suppose the DD-VMS operators yield some necessary structures. For example, if we assume \( g(a) \) in a form of a quadratic ansatz (5.16) for the 2S-DD-VMS-ROM, we can consider \( \tilde{A} \) is negative semidefinite and \( \tilde{B} \) is in a skew-symmetric formulation [12].

### Functional Constraints

For the 2S-DD-VMS-ROM, we impose the following constraints:

\[
(a^{FOM}(t_j))^\top g(a^{FOM}(t_j)) < \epsilon, \quad j = 1, \cdots, M,
\]
where $\epsilon$ is a control parameter and $M$ is the number of training data points. For the 3S-DD-VMS-ROM, we impose the following constraints:

$$
\begin{align*}
(a_{L}^{FOM}(t_j))^\top g_{L}(a_{L}^{FOM}(t_j)) &< \epsilon_L, \quad j = 1, \cdots, M, \\
(a_{S}^{FOM}(t_j))^\top g_{S}(a_{S}^{FOM}(t_j)) &< \epsilon_S, \quad j = 1, \cdots, M,
\end{align*}
$$

where $\epsilon_L$ and $\epsilon_S$ are control parameters for the large scales and the small scales, respectively, $M$ is the number of training data points, and $a = [a_L; a_S]^\top$.

**Structural Constraints**

For the 2S-DD-VMS-ROM, we consider that $\tilde{A}$ is negative semi-definite and $\tilde{B}$ is skew-symmetric; in particular, if we choose $g(a)$ to be a linear ansatz, i.e., (5.14), we consider the following conditions for $\tilde{A}$:

$$
\begin{align*}
\tilde{A}_{ii} &\leq \epsilon, \quad 1 \leq i \leq r, \\
\tilde{A}_{ij} + \tilde{A}_{ji} &= 0, \quad 1 \leq i, j \leq r, \ i \neq j.
\end{align*}
$$

If we choose $g(a)$ to be a quadratic ansatz, i.e., (5.16), we consider the following conditions for $\tilde{A}$ and $\tilde{B}$:

$$
\begin{align*}
\tilde{A}_{ii} &\leq \epsilon, \quad \forall i = 1, \cdots, r, \\
\tilde{A}_{ij} + \tilde{A}_{ji} &= 0, \quad \forall i, j = 1, \cdots, r, \ i \neq j, \\
\tilde{B}_{ii} &= 0, \quad \forall i = 1, \cdots, r, \\
\tilde{B}_{ij} + \tilde{B}_{ji} + \tilde{B}_{kij} + \tilde{B}_{kji} + \tilde{B}_{kij} + \tilde{B}_{kji} &= 0, \\
\tilde{B}_{ijk} + \tilde{B}_{ikj} + \tilde{B}_{jik} + \tilde{B}_{kji} + \tilde{B}_{kij} &= 0, \\
\forall i, j, k = 1, \cdots, r, \ i \neq j \neq k, \ i \neq j.
\end{align*}
$$

where $\epsilon$ is a control parameter.

**Remark 5.1.** For the 3S-DD-VMS-ROM, the structural constraints are more challenging to impose as $\tilde{A}_L$ and $\tilde{A}_S$ are approximated separately and, as a result, $\tilde{A} = [\tilde{A}_L; \tilde{A}_S]^\top$ could hardly be endowed with a structure.
5.4 Hybrid Data-Driven ROMs

Since the three-dimensional channel flow can yield energy stability issues, we may consider combining artificial viscosity stabilization schemes with the DD-VMS-ROM, i.e., developing a hybrid ROM that contains stabilization and closure modeling. In Figure 5.1, we sketch the framework for the hybrid ROM.

![Figure 5.1: Framework of the hybrid ROM](image)

In general, we can write the hybrid ROM in the following form:

\[
\dot{a} = b + (A + \tilde{A}_T + \tilde{A}_{vms})a + a^\top B a ,
\]  

(5.31)

for the linear ansatz, and

\[
\dot{a} = b + (A + \tilde{A}_T + \tilde{A}_{vms})a + a^\top (B + \tilde{B}_{vms})a ,
\]  

(5.32)

for the quadratic ansatz, where \( \tilde{A}_T \) is an artificial viscosity operator, \( \tilde{A}_{vms}, \tilde{B}_{vms} \) are the DD-VMS operators, and the subscripts are used to distinguish the terms. The DD-VMS operators, i.e., \( \tilde{A}_{vms}, \tilde{B}_{vms} \), can be generated by solving the unconstrained or constrained least square problems. We note that \( \tilde{A}_{vms} = [\tilde{A}_L; \tilde{A}_S]^\top \) and \( \tilde{B}_{vms} = [\tilde{B}_L; \tilde{B}_S]^\top \) if three scales are considered (5.17). In addition, if \( \tilde{A}_{vms}, \tilde{B}_{vms} \) are generated by solving unconstrained least square problems, we denote the ROMs as hybrid data-driven ROMs (abbreviated as 2S-H-DD-VMS-ROM or 3S-H-DD-VMS-ROM); and if \( \tilde{A}_{vms}, \tilde{B}_{vms} \) are generated by solving constrained least square problems, we denote the ROMs as constrained hybrid data-driven ROMs (abbreviated as 2S-CH-DD-VMS-ROM or 3S-CH-DD-VMS-ROM).
5.5 Numerical Results

In this section, we investigate the hybrid ROM in the numerical simulation of the three-dimensional channel flow at Reynolds number $Re = 13,750$.

5.5.1 Numerical Setting

Computational setting We use the NSE (5.1) and (5.2) as the mathematical model. The domain is a rectangular box with $\Omega = (-2\pi, 2\pi) \times (0, 2) \times (-2\pi/3, 2\pi/3)$ (see Figure 5.2). No slip boundary conditions are enforced on the walls $y = 0$ and $y = 2$, periodic boundary conditions are imposed on the remaining walls. We also assume that the forcing term is given by $f = <1, 0, 0>^T$. We run the cases with $Re_\tau = 395$ ($Re = 13,750$) where the kinematic viscosity is taken to be $\nu_\tau = \frac{1}{Re_\tau}$.

![Figure 5.2: Geometry of the three-dimensional channel flow](image)

Snapshot Generation To generate the snapshots, we run a large eddy simulation (LES) model using the rNS-$\alpha$ scheme (see [17] and [16]) with the time step size $\Delta t = 2 \times 10^{-3}$.

ROM Construction We collect a total of 5000 snapshots from $t = 60$ to $t = 70$ and generate the POD basis for the ROMs. For the ROM time discretization, we utilize the commonly used linearized BDF2 temporal discretization with a time step size $\Delta t = 0.002$. We use the projections of two snapshots with $t = 60$ and $t = 60.002$ from the LES data as the initial conditions for all ROMs. For convenience, we denote $T = 0$ as $t = 60s$ in the LES model.
To train $\hat{A}$, $\hat{B}$ (for the 2S-DD-VMS-ROM and 2S hybrid ROM) and $\tilde{A}_L$, $\tilde{B}_L$ and $\tilde{A}_S$, $\tilde{B}_S$ (for the 3S-DD-VMS-ROM and 3S hybrid ROM), we use the same 5000 snapshots as for the ROM basis functions. We test all ROMs in the time interval $[0, 10]$ for the reconstructive regime and in the time interval $[0, 20]$ for the cross-validation regime.

In Figure 5.3, we plot the scaled eigenvalues $\lambda_k/\lambda_1$ for three flow settings: (i) the 3D channel flow with $Re = 13,750$, (ii) the 2D flow past a circular cylinder with $Re = 1000$ and (iii) the quasi-geostrophic equation (QGE) with $Re = 450$ and $Ro = 0.0036$. This plot shows that the eigenvalues decay much faster for the flow past a cylinder case than for the QGE and 3D channel flow cases; moreover, the eigenvalues decay the slowest for the 3D channel flow case.

In Figure 5.4, we plot the centering trajectory $U_0$ and the selected ROM velocity basis functions, $\varphi_1$, $\varphi_{25}$, and $\varphi_{50}$. We observe that, as the ROM basis index increases, the spatial structures displayed by the ROM basis functions become smaller; but even for $\varphi_{50}$, its spatial structure is not negligible.
5.5. Numerical Results

Figure 5.4: Centering trajectory $U_0$ and POD bases, $\varphi_1$, $\varphi_{25}$, and $\varphi_{50}$ for the three-dimensional channel flow.

5.5.2 Numerical Criteria

In contrast to the two-dimensional flow past a circular cylinder, it is not practical to use the $L^2$ errors of the velocity field to compare the ROM models with LES data. To compare the ROMs’ performance, we use two different criteria to compare the different ROM models, (i) the kinetic energy $E(t)$ and (ii) the second order statistics.

**Kinetic energy** We follow the definition of kinetic energy over the given domain $\Omega$: 

$$E(t) = \int_{\Omega} \frac{1}{2} \left( \frac{\partial u}{\partial t} + (u \cdot \nabla) u \right)^2 \, dx$$
\[ E(t) = \frac{1}{2} \int_{\Omega} ((u(x,t))^2 + v(x,t)^2 + w(x,t)^2) \, dx. \]  \hspace{1cm} (5.33)

In Figure 5.5, we plot the kinetic energy of LES data as well as its projections, \(u^{proj}\), of LES data to different \(r\)-dimensional POD bases:

\[ u^{proj}_r(x,t) = \sum_{i=1}^{r} \left( \langle u(x,t), \varphi_i(x) \rangle \right) \varphi_i(x), \]  \hspace{1cm} (5.34)

where \(\{\varphi_i\}_{i=1}^{r}\) are the first \(r\) POD basis functions.

![Figure 5.5: Kinetic energy over \(t \in [0,10]\) for LES data and projection](image)

**Second Order Statistics** Following [16], we consider two quantities from the second-order statistics from the flow: (i). \(U_{RMS}\): the normalized root mean square (RMS):

\[ U_{RMS} := \left| \tilde{\mathbf{R}}_{11} - \frac{1}{3} \sum_{j=1}^{3} \tilde{\mathbf{R}}_{jj} \right|^{1/2} \left/ \mathbf{u}_{r,h} \right|. \]  \hspace{1cm} (5.35)

and (ii). \(\mathbf{R}_{12}\): the streamwise-spanwise Reynolds stress tensor component:

\[ \mathbf{R}_{12} := \frac{\tilde{\mathbf{R}}_{12}}{\mathbf{u}_{r,h}^2}. \]  \hspace{1cm} (5.36)
For a given velocity $\mathbf{u}_h = (u_1, u_2, u_3)$, the Reynolds stress tensor components are calculated as follows:

$$R_{ij} = \langle \langle u_i u_j \rangle_s \rangle_t - \langle \langle u_i \rangle_s \rangle_t \langle \langle u_j \rangle_s \rangle_t.$$  

(5.37)

In the comparison, the DNS is from the benchmark [11, 13]. In Figure 5.6, we plot the second-order statistics, $R_{12}$, and the root mean square intensities. We use LES data for different sampling intervals, i.e., snapshots with $t$ in [60, 62], [60, 65], [60, 68], and [60, 70]. These results suggest that the time interval [60, 65] is appropriate for our numerical investigation.

Figure 5.6: Second-order statistics for different sampling time intervals. Left: RMS turbulent intensities; Right: The $R_{12}$ Reynolds stress components.

### 5.5.3 Qualitative Summary of Numerical Results

Before we present the quantitative details for the hybrid data-driven ROMs, we first present the qualitative results. Table 5.2 provides a qualitative summary of the numerical results.
<table>
<thead>
<tr>
<th>ROM Type</th>
<th>Ansatz Type</th>
<th>Constraint Type</th>
<th>Kinetic Energy</th>
</tr>
</thead>
<tbody>
<tr>
<td>G-ROM</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>AV-G-ROM</td>
<td>N/A</td>
<td>N/A</td>
<td>Stable</td>
</tr>
<tr>
<td>2S-DD-VMS-ROM</td>
<td>Linear</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Quadratic</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>2S-CDD-VMS-ROM</td>
<td>Linear</td>
<td>Functional</td>
<td>Stable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Structural</td>
<td>Stable</td>
</tr>
<tr>
<td></td>
<td>Quadratic</td>
<td>Functional</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Structural</td>
<td>N/A</td>
</tr>
<tr>
<td>3S-DD-VMS-ROM</td>
<td>Linear</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Quadratic</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>3S-CDD-VMS-ROM</td>
<td>Linear</td>
<td>Functional</td>
<td>Stable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Structural</td>
<td>Stable</td>
</tr>
<tr>
<td></td>
<td>Quadratic</td>
<td>Functional</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Structural</td>
<td>N/A</td>
</tr>
<tr>
<td>2S-H-DD-VMS-ROM</td>
<td>Linear</td>
<td>N/A</td>
<td>Stable</td>
</tr>
<tr>
<td></td>
<td>Quadratic</td>
<td>N/A</td>
<td>Stable</td>
</tr>
<tr>
<td>2S-CH-DD-VMS-ROM</td>
<td>Linear</td>
<td>Functional</td>
<td>Stable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Structural</td>
<td>Stable</td>
</tr>
<tr>
<td></td>
<td>Quadratic</td>
<td>Functional</td>
<td>Stable</td>
</tr>
<tr>
<td>3S-H-DD-VMS-ROM</td>
<td>Linear</td>
<td>N/A</td>
<td>Stable</td>
</tr>
</tbody>
</table>

Table 5.2: Qualitative description of the different ROMs
5.5.4 Numerical Results for $r = 10$

In this section, we present the numerical results, i.e., the kinetic energy and second-order statistics, for the DD-VMS-ROM and the hybrid ROM. For each case, the ROM kinetic energy is calculated over $t \in [0, 20]$ and we take the LES projection data as a benchmark for comparison purposes. The second-order statistics are calculated over $t \in [0, 10]$ for both ROMs and LES projection data. For the LES projection data, we only have for $[0, 10]$, so for a fair comparison, second-order statistics are calculated for $[0, 10]$ for all ROMs and LES. We consider the performance of low-dimensional ROMs, i.e., $r = 10$, which is usually challenging for the turbulent channel flow. In Figure 5.7, for $r = 10$, we plot the kinetic energies for the LES projection and different ROMs: (i). AV-G-ROM, (ii). 2S-CDD-VMS-ROM with a linear ansatz, (iii). 2S-H-DD-VMS-ROM with a linear ansatz, (iv). 2S-H-DD-VMS-ROM with a quadratic ansatz, and (v). 2S-CH-DD-VMS-ROM with a linear ansatz. Figure 5.7 shows that the AV-G-ROM, 2S-CDD-VMS-ROM with a linear ansatz, and 2S-H-DD-VMS-ROM with a linear ansatz can successfully stabilize the kinetic energies over the long time integration, but they cannot accurately capture the variations of the kinetic energy. On the other hand, the 2S-H-DD-VMS-ROM with a linear ansatz and the 2S-CH-DD-VMS-ROM with a linear ansatz can reproduce the stable kinetic energy and capture its variations at the same time; between the two, the 2S-CH-DD-VMS-ROM with a linear ansatz is better because its kinetic energy’s amplitude is closer to the LES projection data.

![Energy (r = 10)](image)

Figure 5.7: Kinetic energy, $t \in [0, 20]$, for the LES projection and different ROMs with $r = 10$: (i). AV-G-ROM, (ii). 2S-CDD-VMS-ROM with a linear ansatz, (iii). 2S-H-DD-VMS-ROM with a linear ansatz, (iv). 2S-H-DD-VMS-ROM with a quadratic ansatz, and (v). 2S-CH-DD-VMS-ROM with a linear ansatz.
In Figure 5.8, for $r = 10$, we plot the spectrum of kinetic energy for the LES projection and different ROMs: (i) AV-G-ROM, (ii) 2S-CDD-VMS-ROM with a linear ansatz, (iii) 2S-H-DD-VMS-ROM with a linear ansatz, (iv) 2S-H-DD-VMS-ROM with a quadratic ansatz, and (v) 2S-CH-DD-VMS-ROM with a linear ansatz. These plots support the conclusions in Figure 5.7: the 2S-H-DD-VMS-ROM with a linear ansatz and 2S-CH-DD-VMS-ROM with a linear ansatz can reproduce relatively accurately the kinetic energy spectrums at almost all frequencies, while the AV-G-ROM, 2S-CDD-VMS-ROM with a linear ansatz, and 2S-H-DD-VMS-ROM with a linear ansatz can be relatively close to the correct kinetic energy spectrum for low and medium frequencies, but they fail to capture the high frequency of the spectrum.

![Figure 5.8](image.png)

**Figure 5.8**: The spectrum of kinetic energy, $t \in [0, 20]$, for the LES projection and different ROMs with $r = 10$: (i) AV-G-ROM, (ii) 2S-CDD-VMS-ROM with a linear ansatz, (iii) 2S-H-DD-VMS-ROM with a linear ansatz, (iv) 2S-H-DD-VMS-ROM with a quadratic ansatz, and (v) 2S-CH-DD-VMS-ROM with a linear ansatz.

In Figure 5.9, for $r = 10$, we plot the second order statistics, $U_{RMS}$ and $R_{12}$, for the LES projection and different ROMs: (i) AV-G-ROM, (ii) 2S-CDD-VMS-ROM with a linear ansatz, (iii) 2S-H-DD-VMS-ROM with a linear ansatz, (iv) 2S-H-DD-VMS-ROM with a quadratic ansatz, and (v) 2S-CH-DD-VMS-ROM with a linear ansatz. Figure 5.9 shows that among all five different ROMs, the 2S-H-DD-VMS-ROM with a linear ansatz and the 2S-CH-DD-VMS-ROM with a linear ansatz have the second-order statistics that are the closest to the LES projection data.
5.5. Numerical Results

Figure 5.9: Second-order statistics: (Left), RMS turbulent intensities; (Right), The $R_{12}$ Reynolds stress components, for the LES projection and different ROMs with $r = 10$: (i). AV-G-ROM, (ii). 2S-CDD-VMS-ROM with a linear ansatz, (iii). 2S-H-DD-VMS-ROM with a linear ansatz, (iv). 2S-H-DD-VMS-ROM with a quadratic ansatz, and (v). 2S-CH-DD-VMS-ROM with a linear ansatz.

In Table 5.3, we list the artificial viscosity values in the AV-G-ROM and the hybrid data-driven ROMs.

<table>
<thead>
<tr>
<th>ROM types</th>
<th>$\nu_T$ values</th>
</tr>
</thead>
<tbody>
<tr>
<td>AV-G-ROM</td>
<td>$2.6\nu$</td>
</tr>
<tr>
<td>2S-H-DD-VMS-ROM with a linear ansatz</td>
<td>$2.0\nu$</td>
</tr>
<tr>
<td>2S-H-DD-VMS-ROM with a quadratic ansatz</td>
<td>$2.5\nu$</td>
</tr>
<tr>
<td>2S-CH-DD-VMS-ROM with a linear ansatz</td>
<td>$0.6\nu$</td>
</tr>
</tbody>
</table>

Table 5.3: Artificial viscosity values in AV-G-ROM and hybrid data-driven ROMs

Discussion of the numerical results  In Table 5.3, 2S-CH-DD-VMS-ROM with a linear ansatz uses much less artificial viscosity than AV-G-ROM and, at the same time, it reproduces better the kinetic energy (see Figure 5.7) and the second-order statistics (see
Figure 5.9). These results show that hybrid data-driven ROMs can improve the robustness of AV-G-ROM.

**Remark 1** The following ROMs display large numerical oscillations with the given timestep:

- G-ROM
- (2S-3S-DD-VMS-ROM with a linear ansatz
- (2S-)3S-DD-VMS-ROM with a quadratic ansatz.

This implies that for the three-dimensional turbulent channel flow, the DD-VMS-ROM does not necessarily stabilize the solutions.

**Bibliography**


Chapter 6

Verifiability of Hybrid Data-Driven ROMs

6.1 Introduction

The verifiability of DD-VMS-ROM was proved in Chapter 4. In this chapter, our main goal is to extend the verifiability proof to the hybrid data-driven ROM proposed in Chapter 5. The hybrid data-driven ROM is a modified DD-VMS-ROM that involves an additional artificial viscosity term. In Chapter 5, we showed that the hybrid data-driven ROM is numerically more robust than the DD-VMS-ROM in the numerical simulation of the turbulent channel flow.

The notation in this chapter is the same as the notation in Chapter 4 unless otherwise stated.

6.1.1 Artificial Viscosity Model

Different ROM closures are proposed in [2]. One such closure ROM is the following:

\[ \tau_{av}^{ROM}(u_r) = -\nu_T \Delta u_r, \]  

where \( \nu_T \) is the artificial viscosity and \( u_r \) is the ROM solution.

ROM Verifiability
Hybrid Data-Driven ROM Closure Term  We define the hybrid closure term in the hybrid data-driven ROM (6.7) as follows:

\[
\mathcal{H}(u^n_r) = -\nu_T \Delta u^n_r + \tau^{ROM}(u^n_r), \quad u^n_r \in X^r,
\]

where \(\nu_T\) is the artificial viscosity, \(\tau^{ROM}\) is the VMS-ROM closure term defined in equation (4.23), and \(u^n_r\) is the hybrid data-driven ROM’s solution at time \(t_n\). In particular, we assume that this VMS-ROM closure term uses a linear ansatz, which is defined in equation (4.23) in Chapter 4.

6.2 Verifiability of Hybrid Data-Driven ROMs

6.2.1 Hybrid Data-Driven ROM Closure Terms

To construct the hybrid data-driven ROM satisfied by \(P_r(u^n_d)\), we apply the ROM spatial filter, \(P_r\) (see Definition 4.2), to the equation satisfied by the “truth” solution, \(u^n_d\) (i.e., to the full discretization of the \(d\)-dimensional G-ROM (4.6)), we restrict the test functions in (4.6) to the \(r\)-dimensional ROM subspace \(X_r\) defined in (4.12a), and we use the decomposition (4.14). Following [1] (Chapter 4), the LES-ROM equations yields:

\[
\frac{P_r(u^n_d) - P_r(u^{n-1}_d)}{\Delta t}, v) + Re^{-1}(\nabla P_r(u^n_d), \nabla v) + b(P_r(u^n_d), P_r(u^n_d), v) + \mathcal{E}^n + (\tau^{FOM}(u^n_d), v) = (f, v), \quad \forall v \in X^r,
\]

where we used that, by (4.15), \(P_r(f^n), v_r) = (f^n, v_r)\). In the LES-ROM equations (6.3), the Reynolds stress tensor \(\tau^{FOM}(u^n_d)\) and commutation error \(\mathcal{E}\) are defined as follows:

\[
\tau^{FOM}(u^n_d) := P_r(u^n_d \cdot \nabla u^n_d) - P_r(u^n_d) \cdot \nabla P_r(u^n_d),
\]

\[
\mathcal{E}^n := Re^{-1}(\nabla Q_r(u^n_d), \nabla v),
\]

respectively. \(P_r\) and \(Q_R\) are defined in Definition 4.2 in Section 4.3.

Following Remark 4.3 and Definition of closure model 4.4 in Chapter 4, we replace the true closure model \(\tau^{FOM}(u^n_d)\) (which cannot be computed in \(X^r\)) with two terms that can actually be computed in \(X^r\): (1). the closure model \(\tau^{ROM}\), and (2). the artificial viscosity
model $\tau_{av}^{ROM}$ (equation (6.1)).

Since a closure model cannot in general be exact (i.e., $\tau^{FOM}(u^n_d) \neq \tau^{ROM}(P_r(u^n_d))$), when $\tau^{ROM}(P_r(u^n_d))$ is inserted for $\tau^{FOM}(u^n_d)$ in (4.17) the solution of the resulting system is just an approximation to $P_r(u^n_d)$. We denote this hybrid data-driven ROM approximation to $P_r(u^n_d)$ as $u^n_r$, which can be written as

$$u^n_r = \sum_{i=1}^{r} (a^n_r)_i \varphi_i.$$  \hfill (6.6)

Thus, the hybrid data-driven ROM equations for $u^n_r$ are

$$\left( \frac{u^n_r - u^{n-1}_r}{\Delta t}, v_r \right) + Re^{-1}(\nabla u^n_r, \nabla v_r) + (u^n_r \cdot \nabla u^n_r, v_r)
+ (\tilde{H}(u^n_r), v_r) = (f^n, v_r), \quad \forall v_r \in X^r,$$ \hfill (6.7)

where $\nu_T$ is the artificial viscosity and the weak form of the hybrid data-driven closure term yields:

$$(\tilde{H}(u^n_r), v_r) = \nu_T(\nabla u^n_r, \nabla v_r) + (\tau^{ROM}(u^n_r), v_r), \quad u^n_r, v_r \in X^r.$$ \hfill (6.8)

**Remark 6.1.** The hybrid data-driven ROM error at time step $j$, which we denote with $e^j$, is defined as the difference between the large scale component of the “truth” solution, $P_r(u^n_d)$ (which is the solution of (4.32)), and the hybrid data-driven ROM solution of (6.7), $u^n_r$: $e^j = P_r(u^n_d) - u^n_r$.

**Theorem 6.2.** The hybrid data-driven ROM with linear ansatz (4.26) and physical constraint (4.25) satisfies the mean dissipativity condition according to Definition 4.6.

**Proof.** Consider the following:

$$-\nu_T(\Delta (P_r(u^n_d) - u^n_r), P_r(u^n_d) - u^n_r) = \nu_T(\nabla (P_r(u^n_d) - u^n_r), \nabla (P_r(u^n_d) - u^n_r))$$

$$= \nu_T \| \nabla (P_r(u^n_d) - u^n_r) \| \geq 0.$$ \hfill (6.9)

From the proof of Theorem 4.7, we know that

$$(\tau^{ROM}(P_r(u^n_d)) - \tau^{ROM}(u^n_r), P_r(u^n_d) - u^n_r) \geq 0.$$ \hfill (6.10)
Then, combining the inequalities (6.9) and (6.10), we have the following inequality for the hybrid data-driven ROM closure term:

\[
\begin{align*}
(\tilde{H}(P_r(u_d^n)) - \tilde{H}(u^n), P_r(u_d^n) - u^n) \\
= \nu_T \left( \nabla (P_r(u_d^n) - u^n), \nabla (P_r(u_d^n) - u^n) \right) \\
+ (\tau_{\text{ROM}}(P_r(u_d^n)) - \tau_{\text{ROM}}(u^n), P_r(u_d^n) - u^n)
\end{align*}
\]

(6.11)

\[\geq 0.\]

The inequality in (6.11) concludes the proof.

\[\Box\]

### 6.2.2 Verifiability

**Theorem 6.3** (Verifiability of the hybrid data-driven ROM). The hybrid data-driven ROM with linear ansatz (6.7), physical constraint (4.25), and the initial condition \(u^0_r = P_r(u_0^d)\) is verifiable: For a small enough time step, \(\Delta t d_j < 1, \forall j = 1, \ldots, M\), where \(d_j = \left(\frac{3ReC(\Omega)^2}{4}||\nabla P_r(u_j^d)||^4 + Re\right)\) and \(C(\Omega)\) is the constant in Lemma 4.9 in Chapter 4, the following inequality holds for all \(n = 1, \ldots, M\):

\[
\|e^{i=n}\|^2 + \Delta t \sum_{j=1}^{n} Re^{-1}\|\nabla e^j\|^2 \leq \exp \left( \Delta t \sum_{j=1}^{n} \frac{d_j}{1 - \Delta t d_j} \right) \left( \Delta t \sum_{j=1}^{n} Re^{-1}||P_r(\tau_{\text{FOM}}(u_j^d) - \tilde{H}(P_r(u_j^d)))||^2 \right).
\]

(6.12)

**Proof.** The proof of inequality (6.12) is similar to the proof of Theorem 4.10. The main modification is that we use the mean dissipativity of the hybrid data-driven ROM (6.7) in Theorem 6.2 and the DD-VMS-ROM closure term \(\tau_{\text{ROM}}\) is replaced with hybrid data-driven-closure term \(\tilde{H}\).

\[\Box\]

### Bibliography

Chapter 7

Lagrangian Lagrangian Reduced Order Modeling Using Finite Time Lyapunov Exponents

The contents of this chapter have appeared in Fluids, 5 (2020).

The author of this dissertation contributes to the model’s conceptual development and ROM numerical experiments for the quasi-geostrophic equation in Section 7.3.5.

ABSTRACT OF CHAPTER 7

There are two main strategies for improving the projection-based reduced order model (ROM) accuracy: (i) improving the ROM, i.e., adding new terms to the standard ROM; and (ii) improving the ROM basis, i.e., constructing ROM bases that yield more accurate ROMs. In this paper, we use the latter. We propose new Lagrangian inner products that we use together with Eulerian and Lagrangian data to construct new Lagrangian ROMs. We show that the new Lagrangian ROMs are orders of magnitude more accurate than the standard Eulerian ROMs, i.e., ROMs that use standard Eulerian inner product and data to construct the ROM basis. Specifically, for the quasi-geostrophic equations, we show that the new Lagrangian ROMs are more accurate than the standard Eulerian ROMs in approximating not only Lagrangian fields (e.g., the finite time Lyapunov exponent (FTLE)), but also Eulerian fields (e.g., the streamfunction). We emphasize that the new Lagrangian ROMs do not employ any closure modeling to model the effect of discarded modes (which is standard procedure for low-dimensional ROMs of complex nonlinear systems). Thus, the dramatic increase in the new Lagrangian ROMs’ accuracy is entirely due to the novel Lagrangian inner products used to build the Lagrangian ROM basis.

7.1 Introduction

Projection-based reduced order models (ROMs) have been successful in the numerical simulation of fluid flows [9, 33, 36, 58, 66, 83]. To approximate the dynamics of a given flow variable $u$, the ROM strategy proceeds as follows: (i) Choose modes $\{\varphi_1, \ldots, \varphi_R\}$, which represent the recurrent spatial structures in the flow. (ii) Choose the dominant modes $\{\varphi_1, \ldots, \varphi_r\}$, $r \leq R$, as basis functions for the ROM. (iii) Use a Galerkin truncation $u_r = \sum_{j=1}^{r} a_j \varphi_j$. (iv) Replace $u$ with $u_r$ in the underlying equations. (v) Use a Galerkin projection of the PDE obtained in step (iv) onto the ROM space $X_r := \text{span}\{\varphi_1, \ldots, \varphi_r\}$ to obtain a low-dimensional dynamical system, which represents the ROM. (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). The low-dimensional ROMs can decrease the computational cost of traditional full order models (FOMs) by orders of magnitude. ROMs, however, can be inaccurate in the numerical simulation of complex flows [36, 58]. There are two main approaches to increasing ROM accuracy:

The first approach is to improve the model, i.e., to add new terms to the standard projection-based ROM. Classical examples are ROM closure (see, e.g., [16, 34, 55, 62, 69, 86]) and ROM stabilization (see, e.g., [3, 29, 31, 88]). We will not follow this approach in this paper.

The second approach to improving the ROM accuracy is to improve the ROM basis, i.e., to construct ROM bases that yield more accurate ROMs. One of the earliest examples in this class is the $H^1$-basis proposed in [38], in which the $H^1$ inner product is used instead of the standard $L^2$ inner product to construct the ROM basis in order to increase the ROM stability. Similarly, an enstrophy-based ROM for rotational flows was proposed in [79], in which the inner product is defined for vorticity instead of velocity. Other examples in this class are the ROM bases proposed for compressible flows [5, 38, 39, 71], which use new inner products and different flow variables to construct the ROM basis (see [40] for recent work on magnetohydrodynamics). Improved ROM bases were also proposed for data assimilation [21]. The inner products used to define these improved ROM bases are Eulerian inner products, i.e., they are defined only for Eulerian data. To our knowledge, there are only a few Lagrangian inner products, i.e., inner products that are defined on both Eulerian and Lagrangian data, that have been recently proposed. In [45, 53], the authors proposed inner products that are defined for velocity (which is an Eulerian variable) and the Lagrangian mesh coordinates (which are Lagrangian variables).
In this paper, we use the second strategy to improve the ROM accuracy, i.e., we propose improved ROM bases. Specifically, we propose new Lagrangian inner products that utilize both Eulerian and Lagrangian data. In the new Lagrangian inner products, Lagrangian data steers the resulting Lagrangian ROM basis toward an accurate approximation of Lagrangian quantities, whereas Eulerian data helps the Lagrangian ROM basis yield an accurate approximation of Eulerian quantities. We emphasize that the Lagrangian inner products that we propose are different from the Lagrangian inner products in [45, 53]. As Lagrangian data, we use the finite time Lyapunov exponents (FTLE) field, whereas [45, 53] use the Lagrangian mesh coordinates. To construct the new Lagrangian ROMs, we utilize the new Lagrangian inner products, the resulting Lagrangian ROM bases, and the Galerkin projection. In the numerical simulation of the quasi-geostrophic equations [20, 47, 50, 84] (which model large scale ocean circulation), the new Lagrangian ROMs are orders of magnitude more accurate than standard Eulerian ROMs (i.e., ROMs that use standard Eulerian data and inner products to build the ROM bases). Furthermore, the new Lagrangian ROMs are more accurate than the standard Eulerian ROMs in approximating not only Lagrangian fields (e.g., the finite time Lyapunov exponent (FTLE)), but also Eulerian fields (e.g., the streamfunction).

For complex nonlinear systems, it is well known that the low-dimensional ROMs generally need to be equipped with a closure model (see, e.g., [16, 17, 34, 48, 49, 55, 62, 69, 86]) or a stabilization mechanism (see, e.g., [3, 29, 31]) to model the effect of the discarded ROM modes. We emphasize, however, that we investigate the new Lagrangian ROMs without any closure or stabilization (a challenging test) in order to separate the ROM closure problem from the ROM basis generation, which is the main focus of our paper. This allows us to conclude that the orders of magnitude increase in the new Lagrangian ROMs’ accuracy over the standard Eulerian ROMs’ accuracy is entirely due to the new Lagrangian inner products used to build the Lagrangian ROMs’ bases. Of course, we envision that using closure modeling in addition to the novel Lagrangian inner product will increase even further the Lagrangian ROMs’ accuracy.

The rest of the paper is organized as follows: In Section 7.2, we propose the novel Lagrangian inner products and construct the new Lagrangian ROMs. In Section 7.3, for the quasi-geostrophic equations, we show that the new Lagrangian ROMs increase the numerical accuracy of standard Eulerian ROMs by orders of magnitude. Finally, in Section 7.4, we present conclusions and outline future research directions.
7.2 Lagrangian Reduced Order Models

In this section, we propose two new Lagrangian ROMs, which we build as follows: First, we use the quasi-geostrophic equations (QGE) as a mathematical model (although general models, e.g., the Boussinesq and Navier-Stokes equations, could be used instead). Next, we perform numerical simulations to generate the QGE velocity field (which is Eulerian data) and the QGE finite time Lyapunov exponent (FTLE) field (which is Lagrangian data). (We note that other Lagrangian fields could be used instead of the FTLE field.) Finally, we propose two new Lagrangian inner products that use both Eulerian and Lagrangian data to construct new Lagrangian ROM bases, which yield the new Lagrangian ROMs. For comparison purposes, we also outline standard Eulerian ROMs [18, 55, 75, 78, 81], which use only Eulerian data (i.e., the velocity field) to generate the ROM basis. In Section 7.3, we compare the new Lagrangian ROMs with the standard Eulerian ROM in the numerical simulation of the QGE.

The QGE [20, 47, 50, 84] are written as the following PDE:

\[
\begin{align*}
\frac{\partial \omega}{\partial t} + J(\omega, \psi) - Ro^{-1} \frac{\partial \psi}{\partial x} &= Re^{-1} \Delta \omega + Ro^{-1} F, \\
\omega &= -\Delta \psi,
\end{align*}
\]

where \( \omega \) is the vorticity, \( \psi \) is the streamfunction, \( Re \) is the Reynolds number, and \( Ro \) is the Rossby number, \( J(\omega, \psi) = \omega_x \psi_y - \omega_y \psi_x \) is the Jacobian term, and \( F \) is the forcing term. The velocity can be computed from the streamfunction according to the following formula:

\[
v = \left( \frac{\partial \psi}{\partial y}, -\frac{\partial \psi}{\partial x} \right).
\]

Details regarding the parameters and nondimensionalization of the QGE (7.1) are given in, e.g., [25, 54, 55, 75, 77].

7.2.1 Finite Time Lyapunov Exponents (FTLE) Computation

Next, we briefly describe the calculation of the FTLE field (see, e.g., [32] for details). Given a velocity field \( \mathbf{v}(\mathbf{x}, t) \) (e.g., the QGE velocity field (7.2)), the trajectories are obtained from
the solutions of the ODE system $\dot{x} = v(x, t)$. Each trajectory $x(t; t_0, x_0)$ is a function of time, but it also depends on the initial position $x_0$ and the initial time $t_0$.

For a given initial time $t_0$ and a given final time $t$, the flow map is the function

$$x_0 \mapsto \phi_{t_0}^t(x_0) = x(t; t_0, x_0).$$

(7.3)

Consider two particles, simultaneously released at time $t_0$; one at location $x$, the other at location $x + \delta x$. Under the effect of the flow map, the small displacement vector between two particles, $\delta x$, changes. After an elapsed time $T = t - t_0$, the new vector between the two particles is

$$\delta x(t_0 + T) = \phi_{t_0}^{t_0+T}(x + \delta x) - \phi_{t_0}^{t_0+T}(x) = D\phi_{t_0}^{t_0+T}(x) \delta x + O(\|\delta x(t_0)\|^2),$$

where $D\phi_{t_0}^{t_0+T} = d\phi_{t_0}^{t_0+T}(x)/dx$ is the Jacobian of the flow map, and $\|\cdot\|$ is the usual Euclidean norm. Consider the right Cauchy-Green strain tensor,

$$C(x, t_0, T) = D\phi_{t_0}^{t_0+T}(x)^T D\phi_{t_0}^{t_0+T}(x).$$

(7.4)

The maximum possible separation between the released particles after a time interval $T$, assuming a sufficiently small initial distance $\|\delta x(t_0)\|$, is

$$\max \|\delta x(t_0 + T)\| = \sqrt{\mu_{\text{max}}(C(x, t_0, T))} \|\delta x(t_0)\|,$$

(7.5)

where $\mu_{\text{max}}$ the largest eigenvalue of the right Cauchy-Green strain tensor $C(x, t_0, T)$. The FTLE, with $t_0$ and $T$ fixed, is considered a scalar field of the Lyapunov exponent as a function of initial position, $x$,

$$\chi_{t_0}^T(x) = \frac{1}{|T|} \ln \sqrt{\mu_{\text{max}}(C(x, t_0, T))}.$$

(7.6)

### 7.2.2 Eulerian Reduced Order Model (E-ROM)

To generate the ROM basis for the standard Eulerian ROM, we use the proper orthogonal decomposition (POD) [36, 58]. We emphasize, however, that the novel Lagrangian ROMs can be used with other ROM bases [9, 18, 33, 65, 66, 83]. The POD starts by collecting the
7.2. Lagrangian Reduced Order Models

snapshots \( \{ \omega_h^1, \ldots, \omega_h^M \} \), which are, e.g., finite element (FE) approximations of the vorticity in the QGE (7.1) at \( M \) different time instances. The POD seeks a low-dimensional basis that approximates the snapshots optimally with respect to a certain norm. Probably the most popular inner product is the \( L^2 \) inner product:

\[
(\omega_1, \omega_2) = \int_{\Omega} \omega_1(x) \omega_2(x) \, dx .
\] (7.7)

The solution of the resulting minimization problem is equivalent to the solution of the eigenvalue problem

\[
Y^T M_h Y \varphi_j = \lambda_j \varphi_j, \quad j = 1, \ldots, N, \tag{7.8}
\]

where \( Y \) denotes the snapshot matrix, whose columns correspond to the FE coefficients of the snapshots, \( M_h \) denotes the FE mass matrix, and \( N \) is the dimension of the FE space. The eigenvalues are real and non-negative, so they can be ordered as follows: \( \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_R \geq \lambda_{R+1} = \ldots = \lambda_N = 0 \). The POD vorticity basis \( \{ \varphi_j \}_{j=1}^r \) are obtained from the eigenfunctions in (7.8) that correspond to the first \( r \leq R \) largest eigenvalues. Thus, the ROM vorticity space is defined as \( X^r := \text{span}\{ \varphi_1, \ldots, \varphi_r \} \). We follow [55, 75] and define the POD streamfunction basis as the normalized functions \( \{ \phi_j \}_{j=1}^r \), which are chosen such that

\[
-\Delta \phi_j = \varphi_j, \quad j = 1, \ldots, r. \tag{7.9}
\]

The ROM approximations of the vorticity and streamfunction are

\[
\omega_r(x, t) = \sum_{j=1}^r a_j(t) \varphi_j(x) , \quad \psi_r(x, t) = \sum_{j=1}^r a_j(t) \phi_j(x) , \tag{7.10}
\]

where \( \{ a_j(t) \}_{j=1}^r \) are the sought time-varying ROM coefficients. We emphasize that, with the choices in (7.9)–(7.10), once the coefficients \( a_j \) are determined from (7.1a), equation (7.1b) is automatically satisfied. Replacing the vorticity \( \omega \) by \( \omega_r \) in the QGE (7.1a) and then using a Galerkin projection onto \( X^r \), we obtain the Eulerian ROM (E-ROM) for the QGE: \( \forall i = 1, \ldots, r, \)

\[
\left( \frac{\partial \omega_r}{\partial t}, \varphi_i \right) + \left( J(\omega, \psi), \varphi_i \right) - Ro^{-1} \left( \frac{\partial \psi}{\partial x}, \varphi_i \right) + Re^{-1} \left( \nabla \omega_r, \nabla \varphi_i \right) = Ro^{-1} \left( F, \varphi_i \right) . \tag{7.11}
\]
The E-ROM (7.11) yields the following autonomous dynamical system for the vector of time coefficients, $\mathbf{a}(t)$:

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

(7.12)

where $\mathbf{b}$, $\mathbf{A}$, and $\mathbf{B}$ correspond to the constant, linear, and quadratic terms in the numerical discretization of the QGE (7.1), respectively. The finite dimensional system (7.12) can be written componentwise as follows: For all $i = 1, \ldots, r$,

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

(7.13)

where

$$b_i = \text{Ro}^{-1} \left( F, \varphi_i \right),$$

(7.14)

$$A_{im} = \text{Ro}^{-1} \left( \frac{\partial \phi_m}{\partial x}, \varphi_i \right) - R\text{e}^{-1} \left( \nabla \varphi_m, \nabla \varphi_i \right),$$

(7.15)

$$B_{imn} = - \left( J(\varphi_m, \phi_n), \varphi_i \right).$$

(7.16)

The E-ROM (7.11) has been investigated in the numerical simulation of the QGE (7.1) (see, e.g., [55, 75, 78, 81]), where it was shown that it can decrease the computational cost of standard algorithms by orders of magnitude. However, the numerical simulations in [55, 75] have also shown that a low-dimensional E-ROM is not able to produce accurate approximations of standard Eulerian quantities, such as the streamfunction and the velocity fields. (We will also show in Section 7.3 that the standard E-ROM (7.11) produces inaccurate approximations of Lagrangian quantities, such as the FTLE field.) The E-ROM’s numerical inaccuracy in [55, 75] is due to the lack of a closure model [55, 86], i.e., a model for the effect of the discarded ROM modes. Thus, to alleviate its numerical inaccuracy, in [55, 75] the standard E-ROM (7.11) was supplemented with a stabilizing mechanism that yielded relatively accurate results.

In the next section, we pursue a fundamentally different research avenue to improve the standard E-ROM’s numerical accuracy. Instead of modifying the ROM (i.e., adding a closure model, as done in [75]), we propose using a novel set of basis functions that combine Lagrangian and Eulerian data.
7.2.3 Lagrangian ROMs

In this section, we put forth two Lagrangian ROMs, in which both the snapshots and the inner product use Lagrangian data (i.e., the FTLE field, \( \lambda \)) in addition to the Eulerian data (i.e., the vorticity field, \( \omega \)). The Eulerian data helps the resulting ROM basis yield an accurate approximation of the Eulerian output. On the other hand, the Lagrangian data “steers” the ROM basis toward an accurate approximation of the Lagrangian output.

The main tools that we use to construct the new Lagrangian ROMs for FTLE computation are two novel Lagrangian inner products, which are fundamentally different from the standard \( L^2 \) inner product (7.7) used to develop the Eulerian ROM (i.e., the E-ROM (7.11)). These new inner products are Lagrangian inner products \( (\cdot, \cdot)_{\text{FTLE}} \), which aim at including both Eulerian data (i.e., the vorticity field) and Lagrangian data (i.e., the FTLE field) in the ROM basis generation. We emphasize that adding FTLE data to the snapshots is not redundant, since this changes the relative ordering of the eigenpairs of the POD eigenproblem and, therefore, yields a significantly different ROM basis (see snapshot difference quotients used for E-ROM (7.11) in [37] for similar behavior in an Eulerian setting). The two new Lagrangian inner products generate ROM basis functions that are different from the standard E-ROM modes, which are built with the standard \( L^2 \) inner product (see Fig. 7.3). These two new bases yield two new Lagrangian ROMs, which we present in Sections 7.2.3 and 7.2.3. In Section 7.3, in the numerical simulation of the QGE, we show that these two novel Lagrangian ROMs are orders of magnitude more accurate than the standard E-ROM (7.11).

\( \alpha \)-ROM

The first Lagrangian inner product that we propose is

\[
\left( (\omega_1, \lambda_1), (\omega_2, \lambda_2) \right)_{\text{FTLE}} = \int \omega_1(x) \omega_2(x) + \alpha \lambda_1(x) \lambda_2(x) \, dx,
\]

(7.17)

where \( \omega_1 \) and \( \omega_2 \) are vorticity fields and \( \lambda_1 \) and \( \lambda_2 \) are FTLE fields. Thus, the Lagrangian inner product (7.17) combines Lagrangian data \((\lambda_1, \lambda_2)\) with Eulerian data \((\omega_1, \omega_2)\). The parameter \( \alpha \) in (7.17) is a weighting parameter that measures the Lagrangian data’s contribution to the inner product: When \( \alpha = 0 \), the Lagrangian data does not play any role, so the inner product (7.17) is the standard \( L^2 \) inner product (7.7) used to build the standard
When $\alpha > 0$, the Lagrangian data plays a significant role: The higher the $\alpha$ value, the more important the Lagrangian data contribution to the inner product (7.17).

**Remark 7.1 (Nondimensional inner product).** The Lagrangian inner product (7.17) combines data (the vorticity field and the FTLE field) that has the same dimensional units (i.e., inverse time). Thus, the two different types of variables ($\omega$ and $\lambda$) in (7.17) can be added together. Furthermore, the QGE (7.1) used to compute $\omega$ and $\lambda$ are **nondimensionalized**, so the two types of variables could be added even if they did not have the same dimensional units. Finally, if the QGE were left in their original dimensional form, we would need to scale the variables $\omega$ and $\lambda$ appropriately.

We use the new Lagrangian inner product (7.17) to generate the ROM basis for a new Lagrangian ROM. First, we collect snapshots that consist of both vorticity and FTLE approximations. (Note that this is different from the standard E-ROM (7.11) basis generation, where only vorticity snapshots were collected.) Then, we construct the new Lagrangian ROM basis that approximates the snapshots optimally with respect to the Lagrangian norm

$$
\|\omega\| + \sqrt{\alpha} \|\lambda\|. 
$$

(7.18)

(Again, we note that this is different from the approach used for the standard E-ROM (7.11), which utilizes the norm $\|\omega\|$.) Finally, from the resulting ROM basis functions, we only use their vorticity components in the ROM (7.11).

The novel Lagrangian ROM for the FTLE computation is the ROM (7.11) in which the ROM basis is generated by using the new Lagrangian inner product (7.17) instead of the standard $L^2$ inner product (7.7) used to build the E-ROM (7.11). In what follows, we will denote by $\alpha$-ROM the resulting new Lagrangian ROM. Since the new $\alpha$-ROM includes FTLE data (through both the snapshots and the Lagrangian inner product (7.17)), we expect it to yield a more accurate FTLE approximation than the standard E-ROM (7.11), which does not explicitly include FTLE data.
The second Lagrangian inner product that we propose is

\[
(\omega_1, \omega_2)_{FTLE} = \int_\Omega \langle \lambda \rangle(x) \omega_1(x) \omega_2(x) \, dx ,
\]

(7.19)

where \(\langle \lambda \rangle\) is the time average of the FTLE field, \(\lambda\). The Lagrangian inner product (7.19) is similar to the Lagrangian inner product (7.17) in that both use Lagrangian data (i.e., the FTLE field). We note, however, that the way in which Lagrangian and Eulerian data is combined in the two inner products is different: the Lagrangian data is added to the Eulerian data in (7.17), whereas in (7.19) the Lagrangian data is first time averaged and then it is used as a scaling factor for the Eulerian data. We also note that in the numerical investigation in Section 7.3, we use a QGE setting in which the time averages of the streamfunction and FTLE fields play an important role. Thus, we expect the Lagrangian inner product (7.19) to produce accurate results in that setting. Finally, as noted in Remark 7.1, the data used in the Lagrangian inner product (7.19) is nondimensional, so the definition (7.19) is appropriate.

When we use the Lagrangian inner product (7.19) to generate the Lagrangian ROM basis, these basis functions approximate the snapshots optimally with respect to the norm

\[
\left( \int_\Omega \langle \lambda \rangle(x) \omega^2(x) \, dx \right)^{\frac{1}{2}} .
\]

(7.20)

Note that, by definition, the FTLE field (7.6) is always positive. Thus, the Lagrangian inner product (7.19) and the associated norm (7.20) are well defined.

The second new Lagrangian ROM for the FTLE computation is the ROM (7.11) in which the ROM basis is generated by using the new Lagrangian inner product (7.19) instead of the standard \(L^2\) inner product (7.7) used to build the E-ROM (7.11). In what follows, we will denote by \(\lambda\)-ROM the resulting new Lagrangian ROM. Again, since the new \(\lambda\)-ROM includes FTLE information, (through both the snapshots and the Lagrangian inner product (7.17)), we expect it to yield a more accurate FTLE approximation than the E-ROM (7.11), which does not explicitly include FTLE information.


7.2.4 Previous Relevant Work

To our knowledge, there is only little work on reduced order modeling for the FTLE calculation [4, 41, 82]. We emphasize that the Lagrangian ROMs proposed in this paper are fundamentally different from the ROMs used in [4, 41], which are Eulerian ROMs. The Lagrangian ROMs are also different from the ROM used in [82], since the FTLE field is used in [82] only to choose the number, not the actual form of ROM modes, whereas we explicitly use the FTLE field to define the FTLE inner product (7.17) and, thus, to construct the ROM basis.

The Lagrangian ROMs proposed in this paper are related to ROMs that aim at tackling the challenges posed by transport-dominated problems, e.g., wave-like phenomena, moving interfaces, and moving shocks. The ROMs for transport-dominated problems are surveyed in [45, 53, 60] and include development of, e.g., local bases [2, 74], domain decomposition [46], adaptivity [11, 63], symmetry and self similarity transformations [72, 73], approximated Lax pairs [26], and transport maps [7, 10, 57, 60, 61, 68, 70, 85].

There are also connections between the new Lagrangian ROMs and the ROMs that preserve Lagrangian structure [12, 43] (see also [1, 8, 14, 27, 64] for ROMs that preserve Hamiltonian structure) as well as the energy-conserving ROMs for the Navier-Stokes equations [13, 23, 24, 44, 51, 52, 55, 89].

We also note that including Lagrangian information to build the ROM basis is similar to the difference quotients used in [37, 42] and collecting snapshots for the nonlinear terms in the Empirical Interpolation Method (EIM) [6] and its discrete version, the Discrete Empirical Interpolation Method (DEIM) [15]. Indeed, in all these methods, one collects linear combinations of the snapshots. Of course, this does not change the rank of the snapshot matrix, but can change the ordering of its singular values and, thus, yield different ROM bases. Adding Lagrangian information to the set of snapshots is similar in spirit: We do not necessarily add new information, but we “steer” the ROM basis in a certain direction.

7.3 Numerical Results

In Section 7.2.3, we proposed two new Lagrangian ROMs (i.e., the $\alpha$-ROM and the $\lambda$-ROM) for the numerical simulation of the FTLE field. For clarity, Table 7.1 summarizes the inner
products used to build the basis functions of the two new Lagrangian ROMs, as well as the standard Eulerian ROM (i.e., the E-ROM (7.11)). In this section, we perform a numerical investigation of the two new Lagrangian ROMs. To separate the ROM closure modeling from the ROM basis generation, we investigate the two new Lagrangian ROMs without any closure model or stabilization mechanism.

Table 7.1: The new Lagrangian ROMs ($\alpha$-ROM and $\lambda$-ROM), the standard Eulerian ROM (E-ROM), and the inner products used to construct their bases.

<table>
<thead>
<tr>
<th>inner product</th>
<th>ROM</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>equation (7.7)</td>
<td>E-ROM</td>
<td>Eulerian</td>
</tr>
<tr>
<td>equation (7.17)</td>
<td>$\alpha$-ROM</td>
<td>Lagrangian</td>
</tr>
<tr>
<td>equation (7.19)</td>
<td>$\lambda$-ROM</td>
<td>Lagrangian</td>
</tr>
</tbody>
</table>

In this section, we investigate the Lagrangian $\alpha$-ROM and $\lambda$-ROM in the numerical simulation of the velocity and FTLE fields of the QGE (7.1). For comparison purposes, we also test the standard Eulerian ROM (i.e., the E-ROM (7.11)). As a benchmark, we use the full order model (FOM), which is outlined in Algorithm 6:

**Algorithm 6: Full Order Model (FOM)**

1. Compute high resolution streamfunction $\psi^{FOM}$ on $[T_{min}, T_{max}]$.
2. Use $\psi^{FOM}$ in (1) and formula (7.2) to compute high resolution velocity $v^{FOM}$ on $[T_{min}, T_{max}]$.
3. Use $v^{FOM}$ in (2) to calculate (see § 7.2.1) high resolution FTLE field $\lambda^{FOM}$ on $[T_{min}, T_{max}]$.

In the numerical investigation of the three ROMs (i.e., $\alpha$-ROM, $\lambda$-ROM, and E-ROM), we use Algorithm 7. We also use two types of regimes: (i) the *reconstructive* regime; and (ii) the *predictive* regime. The two regimes have fundamentally different goals: The reconstructive regime is an easier test, in which the ROM is validated on the same time interval as the time interval used to train the ROM. The predictive regime is a harder test case, in which the ROM is trained on a short time interval, e.g., $[T_{min}, \frac{T_{max}}{2}]$ and validated on a longer time
interval $[T_{\text{min}}, T_{\text{max}}]$.

**Algorithm 7: Reduced Order Model (ROM)**

1. Compute high resolution streamfunction $\psi^{\text{FOM}}$ on $[T_{\text{min}}, T_{\text{max}}]$.

2. Use $\psi^{\text{FOM}}$ in (1) and formula (7.2) to compute high resolution velocity $\mathbf{v}^{\text{FOM}}$ on $[T_{\text{min}}, T_{\text{max}}]$.

3. Use $\mathbf{v}^{\text{FOM}}$ in (2) on $[T_{\text{min}}, T_{\text{max}}]$ to construct Lagrangian ROMs ($\alpha$-ROM and $\lambda$-ROM) and Eulerian ROM (E-ROM).

4. Use ROMs in (3) to compute low resolution ROM streamfunction $\psi^{\text{ROM}}$ on $[T_{\text{min}}, T_{\text{max}}]$.

5. Use low resolution streamfunction $\psi^{\text{ROM}}$ in (4) and formula (7.2) to compute low resolution velocity $\mathbf{v}^{\text{ROM}}$ on $[T_{\text{min}}, T_{\text{max}}]$.

6. Use low resolution velocity $\mathbf{v}^{\text{ROM}}$ in (5) to calculate low resolution ROM-FTLE field $\lambda^{\text{ROM}}$ on $[T_{\text{min}}, T_{\text{max}}]$.

### 7.3.1 Test Problem Setup

As a test problem in our numerical investigation, we consider the QGE (7.1) with a symmetric double-gyre wind forcing given by $F = \sin(\pi (y - 1))$, which yields a four-gyre circulation in the time mean. This test problem has been used in numerous studies (see, e.g., [19, 28, 35, 54, 55, 56, 67, 75, 76, 77]) as a simplified model for more realistic ocean dynamics.

We emphasize that the four-gyre QGE test problem represents a significant challenge for standard numerical methods: Indeed, as shown in [28], although a double-gyre wind forcing is used, the long term time-average yields a *four-gyre* pattern (see Fig. 7.1). On realistic coarse meshes, classical numerical methods (e.g., finite element and finite volume methods) generally produce inaccurate approximations to this test problem. In particular, standard numerical discretizations fail to recover the correct four-gyre pattern (see, e.g., [75, 77]).

In the QGE (7.1), we use the same parameters as those used in [35, 55, 75, 77]: $Re = 450$ and $Ro = 0.0036$. The spatial domain of the QGE is $[0, 1] \times [0, 2]$. In the FTLE field computation (7.6), we use $T = 0.05$. 
Figure 7.1: Streamfunction contour plots at $t = 40$ (top, left), $t = 60$ (top, middle), and time-averaged (top, right). FTLE contour plots at $t = 40$ (bottom, left), $t = 60$ (bottom, middle), and time-averaged (bottom, right). An FTLE movie is available at https://youtu.be/JXqdcBVfhMw.
7.3.2 Criteria

To investigate the numerical accuracy of the three ROMs (i.e., $\alpha$-ROM, $\lambda$-ROM, and E-ROM), we compare the ROM results with the FOM results. To this end, we use two fundamentally different types of criteria:

The first type of criteria are Eulerian criteria. Specifically, we compute the $L^2$ norm of the time-averaged streamfunction errors between $\psi^{FOM}$ obtained in Step (2) of Algorithm 6 and $\psi^{ROM}$ obtained in Step (5) of Algorithm 7:

$$\left\| \frac{1}{M} \sum_{j=1}^{M} \psi^{FOM}(t_j) - \frac{1}{M} \sum_{j=1}^{M} \psi^{ROM}(t_j) \right\|_{L^2}^2.$$ \hfill (7.21)

In addition to the quantitative criterion (7.21), we are also using the following qualitative Eulerian criterion: We investigate whether the three ROMs can recover the four-gyre pattern of the time average of the streamfunction in Fig. 7.1, which represents a challenging test for standard numerical methods at realistic low resolutions (see, e.g., [75, 77]).

The second type of criterion we use in our numerical investigation is a Lagrangian criterion. Specifically, we compute the $L^2$ norm of the time-averaged FTLE errors between $\lambda^{FOM}$ obtained in Step (3) of Algorithm 6 and $\lambda^{ROM}$ obtained in Step (6) of Algorithm 7:

$$\left\| \frac{1}{M} \sum_{j=1}^{M} \lambda^{FOM}(t_j) - \frac{1}{M} \sum_{j=1}^{M} \lambda^{ROM}(t_j) \right\|_{L^2}^2.$$ \hfill (7.22)

7.3.3 ROM Snapshot Generation

For the FOM (see Algorithm 6) spatial discretization, we use a spectral method with a $257 \times 513$ spatial resolution [55]. For the FOM time discretization, we utilize a time step $\Delta t = 0.01$ and an explicit Runge-Kutta method (Tanaka-Yamashita, an order 7 method with an embedded order 6 method for error control) and an error tolerance of $1.0 \times 10^{-8}$ in time with adaptive time refinement and coarsening [55]. These spatial and temporal discretizations yield numerical results that are similar to the fine resolution numerical results obtained in [75, 77]. In Fig. 7.2, we plot the time evolution of the spatially averaged kinetic energy, $E(t)$. Figure 7.2 (see also Fig. 1 in [75]) shows that the flow converges to a statistically
steady state, after a short transient interval that ends around $t = 10$. Thus, in our numerical
investigation, we follow [55, 75, 77] and consider the FOM results only on $[T_{\text{min}}, T_{\text{max}}] = [10, 80]$. In Fig. 7.1, we display the instantaneous contour plot for the streamfunction field
at $t = 40$ and $t = 60$. We emphasize that, although $t = 40$ and $t = 60$ are well within the
statistically steady state regime, the flow displays a high degree of variability. Thus, the
numerical approximation of this statistically steady regime remains challenging for the low
resolution ROMs that we investigate in this section.

![Figure 7.2: Time evolution of the spatially averaged kinetic energy of the FOM.](image)

To generate the ROM basis (see Section 7.2.2), we follow [55, 75, 77] and collect 701 snapshots
in the time interval $[T_{\text{min}}, T_{\text{max}}] = [10, 80]$ (on which the statistically steady state regime is attained) at equidistant time intervals.

### 7.3.4 ROM Basis Investigation

The new Lagrangian ROM (i.e., $\alpha$-ROM and $\lambda$-ROM) bases are fundamentally different from
the standard E-ROM (7.11) basis. Indeed, the E-ROM basis is built only from Eulerian
data (i.e., the vorticity $\omega$) by using the standard $L^2$ inner product (7.7). On the other hand,
the $\alpha$-ROM and $\lambda$-ROM bases are constructed from Lagrangian data (i.e., the FTLE field
$\lambda$) in addition to Eulerian data (i.e., the vorticity $\omega$) by using the new Lagrangian inner
product (7.17) and the new Lagrangian inner product (7.19), respectively.

To investigate whether the $\alpha$-ROM and $\lambda$-ROM bases are different from the E-ROM (7.11) basis, in Fig. 7.3 we display the ROM basis functions $\psi_{10}$, $\psi_{20}$, and $\psi_{30}$ generated with the standard $L^2$ inner product (7.7) (i.e., the E-ROM basis functions), the new Lagrangian inner product (7.17) (i.e., the new $\alpha$-ROM basis functions) with $\alpha = 10^4$, and the new Lagrangian inner product (7.19) (i.e., the new $\lambda$-ROM basis functions).

The $\alpha$-ROM basis functions (second row of Fig. 7.3) are completely different from the E-ROM basis functions (first row of Fig. 7.3) for $\psi_{10}$, $\psi_{20}$, and $\psi_{30}$. The $\alpha$-ROM basis functions are also completely different from the $\lambda$-ROM basis functions (third row of Fig. 7.3). The $\lambda$-ROM basis functions (third row of Fig. 7.3) are also different from the E-ROM basis functions (first row of Fig. 7.3), although this time the differences are not as dramatic as before: there are large differences in $\psi_{30}$, moderate differences in $\psi_{20}$, and minor differences in $\psi_{10}$. Overall, the results in Fig. 7.3 show that the new Lagrangian inner product (7.17), the new Lagrangian inner product (7.19), and the standard Eulerian $L^2$ inner product (7.7) generate completely different bases for the Lagrangian $\alpha$-ROM, the Lagrangian $\lambda$-ROM, and the standard E-ROM, respectively. In the next section, we investigate which of these bases yields more accurate ROMs in the FTLE field computation.

7.3.5 ROM Numerical Accuracy

In this section, we perform a numerical investigation of the accuracy of the two Lagrangian ROMs (i.e., $\alpha$-ROM, $\lambda$-ROM). We only consider the effect of the basis functions on the ROM accuracy without using a ROM closure model or ROM stabilization mechanism, which is a challenging test.

We compare the Lagrangian $\alpha$-ROM and $\lambda$-ROM accuracy with the standard E-ROM accuracy. As a benchmark for our comparison, we use the FOM results (Algorithm 6). In Section 7.3.5, we perform an Eulerian investigation of the three ROMs, i.e., we investigate the ROMs’ accuracy in approximating the streamfunction (which is an Eulerian quantity). In Section 7.3.5, we perform a Lagrangian investigation of the three ROMs, i.e., we investigate the ROMs’ accuracy in approximating the FTLE field (which is a Lagrangian quantity). For both the Eulerian and Lagrangian investigations, we consider both the reconstructive regime and the predictive regime.
7.3. NUMERICAL RESULTS

Figure 7.3: ROM basis functions $\psi_{10}$ (first column), $\psi_{20}$ (second column), and $\psi_{30}$ (third column) for the standard E-ROM (first row), new Lagrangian $\alpha$-ROM with $\alpha = 10^4$ (second row), and new Lagrangian $\lambda$-ROM (third row).
In our numerical experiments, we use the following parameter values: For the \( \alpha \)-ROM, we use \( \alpha = 1, \alpha = 10^2, \alpha = 10^3 \), and \( \alpha = 10^4 \). We choose this wide range of parameter values to elucidate the effect of the Lagrangian data on the new \( \alpha \)-ROM. Indeed, the parameter \( \alpha \) in (7.17) is a weighting parameter that measures the Lagrangian data’s contribution to the inner product: The higher the \( \alpha \) value, the more important the Lagrangian data contribution to the inner product (7.17). For all the ROM simulations, we use an RK4 time discretization with \( \Delta t = 10^{-3} \). Finally, for all the ROMs, we utilize the following \( r \) values: 5, 10, 15, 20, 25, 30, 35, 40, 45, and 50. We choose this wide range of values to clarify the effect of the ROM dimension on the ROM accuracy.

**Eulerian Investigation**

In this section, we perform an Eulerian investigation of the accuracy of the two Lagrangian ROMs (i.e., the \( \alpha \)-ROM and \( \lambda \)-ROM). First, we consider the reconstructive regime and then the more challenging predictive regime. In both regimes, we use the two Eulerian criteria described in Section 7.3.2: (i) the quantitative Eulerian criterion (7.21), i.e., the \( L^2 \) norm of the the time-averaged streamfunction errors between \( \psi^{FOM} \) obtained in Step (2) of Algorithm 6 and \( \psi^{ROM} \) obtained in Step (5) of Algorithm 7; and (ii) the qualitative Eulerian criterion based on the ability of the ROMs to recover the four-gyre pattern of the time average of the streamfunction in Fig. 7.1 (which is a challenging test for standard numerical methods at realistic low resolutions [75, 77]).

**Reconstructive Regime:** For the reconstructive regime, we check whether the ROMs can reproduce the dynamics of the underlying system on the same time interval as that used to generate the ROM basis functions, i.e., we validate the ROMs on the same time interval as the time interval used to train the ROM.

In Table 7.2, for different \( r \) values, we list the \( L^2 \) norm of the errors in the time-averaged streamfunction (7.21) for E-ROM, \( \lambda \)-ROM, and \( \alpha \)-ROM with \( \alpha = 1, \alpha = 10^2, \alpha = 10^3 \), and \( \alpha = 10^4 \). These results yield the following conclusions: The E-ROM yields inaccurate results for low \( r \) values. As expected, the E-ROM results get better for large \( r \) values. The \( \lambda \)-ROM results are slightly worse than or similar to the E-ROM results for low \( r \) values and somewhat better for large \( r \) values. The results for \( \alpha \)-ROM with \( \alpha = 1 \) are generally worse than the E-ROM results. The results for \( \alpha \)-ROM with \( \alpha = 10 \) and \( \alpha = 10^2 \) are better than the E-ROM
results for all $r$ values except $r = 10$. The results for $\alpha$-ROM with $\alpha = 10^3$ and $\alpha = 10^4$ are dramatically better than the E-ROM results: For example, for $r = 5$, the only ROM that yields acceptable results is the $\alpha$-ROM with $\alpha = 10^4$; all other ROMs simply blow up (denoted by “N/A” in Table 7.2). Furthermore, for $r = 10$ and $r = 15$, the errors of $\alpha$-ROM with $\alpha = 10^3$ and $\alpha = 10^4$ are two orders of magnitude lower than the E-ROM error. For the larger $r$ values, the errors of $\alpha$-ROM with $\alpha = 10^3$ and $\alpha = 10^4$ continue to be lower than the E-ROM errors, although (as expected) the differences decrease as the $r$ values increase.

As expected, when the $r$ values increase, the results in Table 7.2 show that all the ROMs’ errors converge until they reach a plateau around 5.00e-01 (which is probably due to the ROM truncation error). Overall, the results in Table 7.2 show that the Lagrangian $\alpha$-ROM with high $\alpha$ values (i.e., $\alpha = 10^3$ and $\alpha = 10^4$) yields significantly more accurate results than the standard E-ROM and the Lagrangian $\lambda$-ROM, especially for the small $r$ values.

Table 7.2: Eulerian investigation, reconstructive regime: $L^2$ norm of the errors in the time-averaged streamfunction (7.21) for E-ROM (second column), $\lambda$-ROM (third column), and $\alpha$-ROM for $\alpha = 1$ (fourth column), $\alpha = 10$ (fifth column), $\alpha = 10^2$ (sixth column), $\alpha = 10^3$ (seventh column), and $\alpha = 10^4$ (eighth column).

<table>
<thead>
<tr>
<th>$r$</th>
<th>E-ROM</th>
<th>$\lambda$-ROM</th>
<th>$\alpha = 1$</th>
<th>$\alpha = 10$</th>
<th>$\alpha = 10^2$</th>
<th>$\alpha = 10^3$</th>
<th>$\alpha = 10^4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>10</td>
<td>3.6e+02</td>
<td>4.0e+02</td>
<td>7.9e+02</td>
<td>6.2e+03</td>
<td>6.5e+02</td>
<td>2.9e+00</td>
<td>1.2e+00</td>
</tr>
<tr>
<td>15</td>
<td>1.8e+02</td>
<td>5.5e+02</td>
<td>4.1e+02</td>
<td>1.3e+01</td>
<td>1.1e+01</td>
<td>1.1e+00</td>
<td>2.3e+00</td>
</tr>
<tr>
<td>20</td>
<td>1.3e+01</td>
<td>8.0e+00</td>
<td>1.5e+01</td>
<td>4.2e+00</td>
<td>2.7e+00</td>
<td>1.1e+00</td>
<td>1.1e+00</td>
</tr>
<tr>
<td>25</td>
<td>3.3e+00</td>
<td>2.7e+00</td>
<td>4.7e+00</td>
<td>3.0e+00</td>
<td>2.1e+00</td>
<td>3.9e-01</td>
<td>3.4e-01</td>
</tr>
<tr>
<td>30</td>
<td>2.8e+00</td>
<td>2.0e+00</td>
<td>2.5e+00</td>
<td>2.5e+00</td>
<td>1.1e+00</td>
<td>5.9e-01</td>
<td>4.2e-01</td>
</tr>
<tr>
<td>35</td>
<td>1.5e+00</td>
<td>9.3e-01</td>
<td>1.2e+00</td>
<td>1.3e+00</td>
<td>8.6e-01</td>
<td>5.1e-01</td>
<td>3.1e-01</td>
</tr>
<tr>
<td>40</td>
<td>1.0e+00</td>
<td>6.1e-01</td>
<td>9.3e-01</td>
<td>9.9e-01</td>
<td>5.4e-01</td>
<td>4.6e-01</td>
<td>5.4e-01</td>
</tr>
<tr>
<td>45</td>
<td>5.4e-01</td>
<td>5.2e-01</td>
<td>7.0e-01</td>
<td>6.8e-01</td>
<td>5.1e-01</td>
<td>4.0e-01</td>
<td>7.0e-01</td>
</tr>
<tr>
<td>50</td>
<td>4.5e-01</td>
<td>5.0e-01</td>
<td>5.0e-01</td>
<td>6.2e-01</td>
<td>5.9e-01</td>
<td>6.0e-01</td>
<td>3.8e-01</td>
</tr>
</tbody>
</table>

Next, we use the qualitative Eulerian criterion to investigate the ability of the ROMs to recover the four-gyre pattern of the time average of the streamfunction in Fig. 7.1. In Fig. 7.4, for $r = 10, 15, 20$, and 30, we plot the mean streamfunction for E-ROM, $\lambda$-ROM, and $\alpha$-ROM with $\alpha = 1$ and $\alpha = 10^4$. These results yield the following conclusions: The
E-ROM, the λ-ROM, and the α-ROM with a low α value (i.e., α = 1) yield similar results: These ROMs cannot recover the four-gyre pattern for any of the four r values. However, the α-ROM with a large α value (i.e., $\alpha = 10^4$) yields dramatically better results: This ROM can clearly capture the four-gyre pattern for $r = 30$; for $r = 20$, the pattern is somewhat captured, although not as clearly as for $r = 30$; finally, for $r = 10$ and $r = 15$, only hints of the four-gyre pattern are present. Overall, the plots in Fig. 7.4 show that the Lagrangian α-ROM with a large α value (i.e., $\alpha = 10^4$) can capture the four-gyre pattern, whereas the standard E-ROM and the Lagrangian λ-ROM cannot.

The results in Table 7.2 and Fig. 7.4 consistently show that the new Lagrangian α-ROM with large α values outperforms the standard E-ROM and the Lagrangian λ-ROM with respect to the two Eulerian metrics used in this section. These results also show that the Lagrangian data used to construct the new Lagrangian α-ROM play an important role: the higher the Lagrangian data contribution (i.e., the higher the α value), the more accurate the results.
Figure 7.4: Eulerian investigation, reconstructive regime: Mean streamfunction from E-ROM (first row), λ-ROM (second row) α-ROM with α = 1 (third row), and α = 10^4 (fourth row), for r = 10 (first column), r = 15 (second column), r = 20 (third column), and r = 30 (fourth column).
Predictive Regime: For the predictive regime, we check whether the investigated ROMs can predict the dynamics of the underlying system. Specifically, we validate the ROMs on a time interval that is twice as long as the time interval used to train the ROMs. In Table 7.3, for different $r$ values, we list the $L^2$ norm of the errors in the time-averaged streamfunction (7.21) for E-ROM, $\lambda$-ROM, and $\alpha$-ROM with $\alpha = 1, \alpha = 10, \alpha = 10^2, \alpha = 10^3$, and $\alpha = 10^4$. These results yield conclusions that are similar to those drawn in the reconstructive regime: For low $r$ values, the E-ROM yields inaccurate results. The $\lambda$-ROM results are similar to or slightly better than the E-ROM results for all $r$ values. The results for $\alpha$-ROM with $\alpha = 1, \alpha = 10$, and $\alpha = 10^2$ are generally better than or similar to the E-ROM results. The results for $\alpha$-ROM with $\alpha = 10^3$ and $\alpha = 10^4$ are dramatically better than the E-ROM results: For example, for $r = 5$, the only ROM that yields acceptable results is the $\alpha$-ROM with $\alpha = 10^4$; all other ROMs simply blow up (this is denoted by “N/A” in Table 7.3). Furthermore, for $r = 10$, $r = 15$ and $r = 25$, the errors of $\alpha$-ROM with $\alpha = 10^3$ and $\alpha = 10^4$ are at least one order of magnitude lower than the E-ROM error. For the larger $r$ values, the errors of $\alpha$-ROM with $\alpha = 10^3$ and $\alpha = 10^4$ continue to be lower than the E-ROM errors, although (as expected) the differences decrease as the $r$ values increase. Overall, the results in Table 7.3 show that, in the predictive regime, the Lagrangian $\alpha$-ROM with high $\alpha$ values (i.e., $\alpha = 10^3$ and $\alpha = 10^4$) yields significantly more accurate results than the standard E-ROM and the Lagrangian $\lambda$-ROM, especially for the small $r$ values.
Table 7.3: Eulerian investigation, predictive regime: $L^2$ norm of the errors in the time-averaged streamfunction (7.21) for E-ROM (second column), $\lambda$-ROM (third column), and $\alpha$-ROM for $\alpha = 1$ (fourth column), $\alpha = 10$ (fifth column), $\alpha = 10^2$ (sixth column), $\alpha = 10^3$ (seventh column), and $\alpha = 10^4$ (eighth column).

<table>
<thead>
<tr>
<th>$r$</th>
<th>E-ROM</th>
<th>$\lambda$-ROM</th>
<th>$\alpha = 1$</th>
<th>$\alpha = 10$</th>
<th>$\alpha = 10^2$</th>
<th>$\alpha = 10^3$</th>
<th>$\alpha = 10^4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>2.8e+01</td>
<td>1.2e+01</td>
</tr>
<tr>
<td>10</td>
<td>2.6e+04</td>
<td>8.6e+03</td>
<td>1.6e+04</td>
<td>5.3e+03</td>
<td>4.8e+01</td>
<td>8.1e+00</td>
<td>7.7e+00</td>
</tr>
<tr>
<td>15</td>
<td>1.1e+01</td>
<td>1.8e+01</td>
<td>1.9e+01</td>
<td>2.2e+01</td>
<td>7.3e+00</td>
<td>1.8e+00</td>
<td>1.3e+00</td>
</tr>
<tr>
<td>20</td>
<td>9.0e+00</td>
<td>6.3e+00</td>
<td>8.8e+00</td>
<td>9.5e+00</td>
<td>1.7e+00</td>
<td>3.5e-01</td>
<td>5.8e-01</td>
</tr>
<tr>
<td>25</td>
<td>4.0e+00</td>
<td>2.5e+00</td>
<td>3.8e+00</td>
<td>2.7e+00</td>
<td>8.4e-01</td>
<td>3.3e-01</td>
<td>3.5e-01</td>
</tr>
<tr>
<td>30</td>
<td>1.0e+00</td>
<td>7.7e-01</td>
<td>8.8e-01</td>
<td>6.8e-01</td>
<td>6.0e-01</td>
<td>1.9e-01</td>
<td>6.2e-01</td>
</tr>
<tr>
<td>35</td>
<td>4.3e-01</td>
<td>4.9e-01</td>
<td>5.8e-01</td>
<td>5.5e-01</td>
<td>5.4e-01</td>
<td>3.4e-01</td>
<td>5.2e-01</td>
</tr>
<tr>
<td>40</td>
<td>4.8e-01</td>
<td>3.8e-01</td>
<td>3.8e-01</td>
<td>4.9e-01</td>
<td>4.4e-01</td>
<td>7.6e-01</td>
<td>5.7e-01</td>
</tr>
<tr>
<td>45</td>
<td>6.4e-01</td>
<td>4.4e-01</td>
<td>4.6e-01</td>
<td>4.9e-01</td>
<td>6.3e-01</td>
<td>7.5e-01</td>
<td>5.3e-01</td>
</tr>
<tr>
<td>50</td>
<td>4.0e-01</td>
<td>4.9e-01</td>
<td>3.52e-01</td>
<td>3.8e-01</td>
<td>6.1e-01</td>
<td>4.0e-01</td>
<td>4.3e-01</td>
</tr>
</tbody>
</table>
Next, we use the qualitative Eulerian criterion to investigate the ability of the ROMs to predict the four-gyre pattern of the time average of the streamfunction. In Fig. 7.5, for \( r = 10, 15, 20, \) and 30, we plot the mean streamfunction for E-ROM, \( \lambda \)-ROM, and \( \alpha \)-ROM with \( \alpha = 1 \) and \( \alpha = 10^4 \). These results yield the following conclusions: The E-ROM, the \( \lambda \)-ROM, and the \( \alpha \)-ROM with a low \( \alpha \) value (i.e., \( \alpha = 1 \)) yield similar results. These ROMs cannot recover the four-gyre pattern for \( r = 10, 15, \) and 20, although they can capture the four-gyre pattern for \( r = 30 \). However, the \( \alpha \)-ROM with a large \( \alpha \) value (i.e., \( \alpha = 10^4 \)) yields dramatically better results. This ROM can clearly capture the four-gyre pattern not only for \( r = 30 \), but also for \( r = 30 \); for \( r = 15 \), the pattern is somewhat captured, although not as clearly as for \( r = 20 \) and \( r = 30 \); finally, for \( r = 10 \), only hints of the four-gyre pattern are present. Overall, the plots in Fig. 7.5 show that the Lagrangian \( \alpha \)-ROM with a large \( \alpha \) value (i.e., \( \alpha = 10^4 \)) can capture the four-gyre pattern, whereas the standard E-ROM and the Lagrangian \( \lambda \)-ROM cannot.

The results in Table 7.3 and Fig. 7.5 consistently show that, in the predictive regime, the new Lagrangian \( \alpha \)-ROM with large \( \alpha \) values outperforms the standard E-ROM and the Lagrangian \( \lambda \)-ROM with respect to the two Eulerian metrics used in this section. These results also support the conclusion from the reconstructive regime, i.e., the higher the Lagrangian data contribution (i.e., the higher the \( \alpha \) value), the more accurate the results.
Figure 7.5: Eulerian investigation, predictive regime: Mean streamfunction from E-ROM (first row), λ-ROM (second row) α-ROM with α = 1 (third row), and α = 10^4 (fourth row), for r = 10 (first column), r = 15 (second column), r = 20 (third column), and r = 30 (fourth column).
Lagrangian Investigation

In this section, we perform a Lagrangian investigation of the accuracy of the two Lagrangian ROMs (i.e., the $\alpha$-ROM and $\lambda$-ROM). We follow the same format as that used in the Eulerian investigation in Section 7.3.5.

Reconstructive Regime: In Table 7.4, we list the $L^2$ norm of the errors in the time-averaged FTLE (7.22) for E-ROM, $\lambda$-ROM, and $\alpha$-ROM for different $r$ values. The results in Table 7.4 show that the $\alpha$-ROM with high $\alpha$ values (i.e., $\alpha = 10^3$ and $\alpha = 10^4$) consistently outperform the $\lambda$-ROM and the E-ROM for all $r$ values, but especially for the small $r$ values. We also note that the relatively high magnitudes of the errors in Table 7.4 are due to the errors in the ROM velocity field approximations. Decreasing the magnitude of the errors in the ROM velocity field approximations (e.g., by increasing the ROM dimension, $r$) would probably decrease the magnitude of the errors in the FTLE field approximation in Table 7.4.

<table>
<thead>
<tr>
<th>$r$</th>
<th>E-ROM</th>
<th>$\lambda$-ROM</th>
<th>$\alpha = 1$</th>
<th>$\alpha = 10$</th>
<th>$\alpha = 10^2$</th>
<th>$\alpha = 10^3$</th>
<th>$\alpha = 10^4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>6.0e+03</td>
<td>5.9e+03</td>
<td>5.4e+03</td>
<td>4.2e+03</td>
<td>5.2e+03</td>
<td>4.8e+01</td>
<td>2.5e+01</td>
</tr>
<tr>
<td>15</td>
<td>4.6e+02</td>
<td>1.2e+03</td>
<td>1.2e+02</td>
<td>8.3e+01</td>
<td>7.8e+01</td>
<td>3.2e+01</td>
<td>6.1e+01</td>
</tr>
<tr>
<td>20</td>
<td>9.8e+01</td>
<td>7.4e+01</td>
<td>8.5e+01</td>
<td>5.9e+01</td>
<td>2.2e+01</td>
<td>1.2e+01</td>
<td>1.0e+01</td>
</tr>
<tr>
<td>25</td>
<td>8.2e+01</td>
<td>6.2e+01</td>
<td>8.5e+01</td>
<td>7.6e+01</td>
<td>1.8e+01</td>
<td>1.0e+01</td>
<td>9.5e+00</td>
</tr>
<tr>
<td>30</td>
<td>9.4e+01</td>
<td>8.7e+01</td>
<td>9.2e+01</td>
<td>7.0e+01</td>
<td>1.8e+01</td>
<td>1.2e+01</td>
<td>8.3e+00</td>
</tr>
<tr>
<td>35</td>
<td>7.8e+01</td>
<td>5.8e+01</td>
<td>7.6e+01</td>
<td>5.9e+01</td>
<td>1.3e+01</td>
<td>1.1e+01</td>
<td>1.2e+01</td>
</tr>
<tr>
<td>40</td>
<td>7.5e+01</td>
<td>5.3e+01</td>
<td>5.8e+01</td>
<td>5.7e+01</td>
<td>1.3e+01</td>
<td>2.0e+01</td>
<td>2.6e+01</td>
</tr>
<tr>
<td>45</td>
<td>5.9e+01</td>
<td>5.0e+01</td>
<td>5.5e+01</td>
<td>4.9e+01</td>
<td>2.4e+01</td>
<td>2.4e+01</td>
<td>2.4e+01</td>
</tr>
<tr>
<td>50</td>
<td>4.8e+01</td>
<td>4.4e+01</td>
<td>4.6e+01</td>
<td>3.8e+01</td>
<td>1.4e+01</td>
<td>1.4e+01</td>
<td>2.2e+01</td>
</tr>
</tbody>
</table>

Predictive Regime: In Table 7.5, we list the $L^2$ norm of the errors in the time-averaged FTLE (7.22) for E-ROM, $\lambda$-ROM, and $\alpha$-ROM for different $r$ values. The results in Table 7.5 show that, as in the reconstructive regime, the $\alpha$-ROM with high $\alpha$ values (i.e., $\alpha = 10^3$ and $\alpha = 10^4$) consistently outperforms the $\lambda$-ROM and the E-ROM for all $r$ values.
### Table 7.5: Lagrangian investigation, predictive regime: $L^2$ norm of the errors in the time-averaged FTLE (7.22) for E-ROM (second column) $\lambda$-ROM (third column), and $\alpha$-ROM for $\alpha = 1$ (fourth column), $\alpha = 10$ (fifth column), $\alpha = 10^2$ (sixth column), $\alpha = 10^3$ (seventh column), and $\alpha = 10^4$ (eighth column).

<table>
<thead>
<tr>
<th>$r$</th>
<th>E-ROM</th>
<th>$\lambda$-ROM</th>
<th>$\alpha = 1$</th>
<th>$\alpha = 10$</th>
<th>$\alpha = 10^2$</th>
<th>$\alpha = 10^3$</th>
<th>$\alpha = 10^4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>3.6e+03</td>
<td>4.4e+03</td>
<td>3.0e+03</td>
<td>4.1e+03</td>
<td>3.0e+03</td>
<td>4.4e+02</td>
<td>2.4e+02</td>
</tr>
<tr>
<td>15</td>
<td>5.7e+02</td>
<td>5.2e+02</td>
<td>4.1e+02</td>
<td>5.0e+02</td>
<td>1.0e+02</td>
<td>1.8e+01</td>
<td>1.2e+01</td>
</tr>
<tr>
<td>20</td>
<td>1.5e+02</td>
<td>8.4e+01</td>
<td>1.5e+02</td>
<td>1.7e+02</td>
<td>7.7e+01</td>
<td>1.3e+01</td>
<td>8.9e+00</td>
</tr>
<tr>
<td>25</td>
<td>1.2e+02</td>
<td>9.2e+01</td>
<td>9.5e+01</td>
<td>1.1e+02</td>
<td>5.8e+01</td>
<td>8.9e+00</td>
<td>9.3e+00</td>
</tr>
<tr>
<td>30</td>
<td>7.6e+01</td>
<td>7.3e+01</td>
<td>1.2e+02</td>
<td>6.5e+01</td>
<td>2.1e+01</td>
<td>6.6e+00</td>
<td>6.7e+00</td>
</tr>
<tr>
<td>35</td>
<td>6.7e+01</td>
<td>4.6e+01</td>
<td>5.9e+01</td>
<td>4.6e+01</td>
<td>1.6e+01</td>
<td>6.6e+00</td>
<td>5.6e+00</td>
</tr>
<tr>
<td>40</td>
<td>3.1e+01</td>
<td>2.8e+01</td>
<td>2.6e+01</td>
<td>3.0e+01</td>
<td>1.1e+01</td>
<td>1.3e+01</td>
<td>2.0e+01</td>
</tr>
<tr>
<td>45</td>
<td>2.2e+01</td>
<td>1.8e+01</td>
<td>2.0e+01</td>
<td>1.6e+01</td>
<td>9.3e+00</td>
<td>1.5e+01</td>
<td>1.9e+01</td>
</tr>
<tr>
<td>50</td>
<td>1.9e+01</td>
<td>1.6e+01</td>
<td>1.8e+01</td>
<td>1.5e+01</td>
<td>9.1e+00</td>
<td>1.4e+01</td>
<td>1.5e+01</td>
</tr>
</tbody>
</table>

#### 7.3.6 ROM Computational Efficiency

In this section, we investigate the computational efficiency of the new Lagrangian ROMs (i.e., $\alpha$-ROM and $\lambda$-ROM).

**Computational Environments**

We use the following computational environments: To generate the FOM velocity fields, we run the code on one processor (and one thread) on a Dell workstation with a 2.00 GHZ Intel Xeon CPU running on a 64-bit Linux system. To generate the ROM velocity fields, we use one Apple laptop with a single 2.70 GHZ CPU, running on a 64-bit Macintosh operating systems. To generate the FTLE fields, we utilize: (i) One computing cluster composed of 5 nodes, each node comprised of dual, quad core, hyperthreaded 2.4GHz Intel Xeon E5620 CPUs (16 processor threads), 24GB RAM, and a 40Gbps InfiniBand host card and cable; and (ii) Five nodes at 12 threads per node, for a total of 60 threads, and 4749mb of memory for each thread.
Speed-Up Factors

The FOM CPU time has two components: the CPU time of generating the velocity field; and the CPU time of generating the FTLE field from the velocity field. The ROM CPU time has three components: the CPU time of the offline phase (i.e., the construction of ROM operators); the CPU time of the online phase (i.e., running the ROMs to generate the velocity field); and the CPU time of generating the FTLE field from the velocity field.

In this section, we investigate the CPU times of the velocity computation, since this is the main target of the proposed Lagrangian ROMs. Thus, we first investigate the ROM speed-ups in the velocity computation and then briefly discuss the CPU times in the FTLE field computation. Furthermore, as often done in ROM investigations, we monitor only the CPU time of the online phase of the ROMs, since the offline CPU time is offset by running the ROMs in the predictive regime, i.e., for longer time intervals (as done in this paper) or for different parameters.

To compute the computational efficiency of the new Lagrangian ROMs, we compute the ROM speed-up factors \( S_f \), which are defined as follows:

\[
S_f = \frac{\text{FOM CPU time}}{\text{ROM CPU time}},
\]

where the FOM CPU time is the CPU time of generating the velocity field and the ROM CPU time is the CPU time of the ROM online phase.

In Table 7.6, for different \( r \) values, we list the speed-up factors (7.23) for E-ROM, \( \lambda \)-ROM (second column), and \( \alpha \)-ROM with \( \alpha = 1, \alpha = 10^2, \alpha = 10^3 \), and \( \alpha = 10^4 \). These results show that the new Lagrangian ROMs and the standard Eulerian ROM are more than three orders of magnitude faster than the FOM.

Although the speed-up factors for the ROM velocity computation in Table 7.6 are the main focus of the proposed Lagrangian ROMs, we briefly comment on the CPU time of the ROM computation of the FTLE field. Overall, the CPU time of the ROM computation of the FTLE field is generally slightly higher than the CPU time of the FOM computation of the FTLE field, especially when relative low \( r \) values are used. We also note that, for low \( r \) values, this CPU time increase is generally lower for the new Lagrangian \( \alpha \)-ROM with \( \alpha = 10^3 \) and \( \alpha = 10^4 \) than for the standard E-ROM. We believe that the reason for this
Table 7.6: Speed-up factors (7.23) for velocity field computation: E-ROM (second column), \( \lambda \)-ROM (third column), and \( \alpha \)-ROM for \( \alpha = 1 \) (fourth column), \( \alpha = 10 \) (fifth column), \( \alpha = 10^2 \) (sixth column), \( \alpha = 10^3 \) (seventh column), and \( \alpha = 10^4 \) (eighth column).

<table>
<thead>
<tr>
<th>( r )</th>
<th>E-ROM</th>
<th>( \lambda )-ROM</th>
<th>( \alpha = 1 )</th>
<th>( \alpha = 10 )</th>
<th>( \alpha = 10^2 )</th>
<th>( \alpha = 10^3 )</th>
<th>( \alpha = 10^4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>1.4e+04</td>
<td>1.5e+04</td>
<td>1.5e+04</td>
<td>1.5e+04</td>
<td>1.5e+04</td>
<td>1.5e+04</td>
<td>1.5e+04</td>
</tr>
<tr>
<td>15</td>
<td>8.4e+03</td>
<td>8.2e+03</td>
<td>8.2e+03</td>
<td>8.3e+03</td>
<td>7.8e+03</td>
<td>8.0e+03</td>
<td>8.0e+03</td>
</tr>
<tr>
<td>20</td>
<td>5.6e+03</td>
<td>5.7e+03</td>
<td>5.7e+03</td>
<td>5.9e+03</td>
<td>6.1e+03</td>
<td>5.8e+03</td>
<td>5.8e+03</td>
</tr>
<tr>
<td>25</td>
<td>4.8e+03</td>
<td>4.6e+03</td>
<td>4.7e+03</td>
<td>4.8e+03</td>
<td>4.5e+03</td>
<td>4.6e+03</td>
<td>4.6e+03</td>
</tr>
<tr>
<td>30</td>
<td>3.7e+03</td>
<td>3.6e+03</td>
<td>3.7e+03</td>
<td>3.9e+03</td>
<td>3.6e+03</td>
<td>3.7e+03</td>
<td>3.7e+03</td>
</tr>
<tr>
<td>35</td>
<td>2.9e+03</td>
<td>2.9e+03</td>
<td>3.0e+03</td>
<td>3.0e+03</td>
<td>2.9e+03</td>
<td>2.9e+03</td>
<td>2.9e+03</td>
</tr>
<tr>
<td>40</td>
<td>2.5e+03</td>
<td>2.6e+03</td>
<td>2.6e+03</td>
<td>2.6e+03</td>
<td>2.5e+03</td>
<td>2.6e+03</td>
<td>2.5e+03</td>
</tr>
<tr>
<td>45</td>
<td>2.0e+03</td>
<td>2.0e+03</td>
<td>2.1e+03</td>
<td>2.1e+03</td>
<td>2.0e+03</td>
<td>2.0e+03</td>
<td>2.0e+03</td>
</tr>
<tr>
<td>50</td>
<td>1.7e+03</td>
<td>1.7e+03</td>
<td>1.8e+03</td>
<td>1.8e+03</td>
<td>1.7e+03</td>
<td>1.7e+03</td>
<td>1.7e+03</td>
</tr>
</tbody>
</table>

A slight CPU time increase is that, as expected, the ROM velocity accuracy is lower than the FOM velocity accuracy, which results in a slight increase in the CPU time of the FTLE field computation. We plan to investigate this in a future study.

To conclude, the overall CPU time of the FTLE field computation is generally several times lower for the two new Lagrangian ROMs (as well as the standard E-ROM) than for the DNS. Indeed, the two new Lagrangian ROMs decrease the CPU time of the FOM velocity field computation by orders of magnitude and only slightly increase the CPU time of the FOM FTLE field computation.

## 7.4 Conclusions and Outlook

In this paper, we proposed Lagrangian ROMs that use new Lagrangian inner products to build the ROM basis. In these Lagrangian inner products, Lagrangian data “steers” the resulting Lagrangian ROM basis toward an accurate approximation of Lagrangian quantities, whereas Eulerian data helps the Lagrangian ROM basis yield an accurate approximation of Eulerian quantities.

For complex nonlinear systems, the low-dimensional ROMs generally need to be equipped with closure models or stabilization mechanisms [55, 55, 86]. We emphasize, however, that we studied the new Lagrangian ROMs without any closure model (a challenging test) in
order to separate the ROM closure problem from the ROM basis generation, which is the main focus of our paper.

We investigated the new Lagrangian ROMs in the numerical simulation of the QGE. We considered both the reconstructive regime (in which the ROM is validated on the same time interval as the time interval used to train the ROM) and the predictive regime (in which the ROM is trained on a short time interval and validated on a longer time interval). In both the reconstructive and predictive regimes, we showed that the new Lagrangian ROMs numerical accuracies are orders of magnitude higher than the standard Eulerian ROM accuracy in approximating both Eulerian fields (i.e., the velocity field) and Lagrangian fields (i.e., the FTLE field). We emphasize that, since the new Lagrangian ROMs did not employ any closure modeling, the dramatic increase in the new Lagrangian ROMs’ accuracy is entirely due to the new Lagrangian inner products used to build the Lagrangian ROM basis. Furthermore, we showed that, for the velocity field computations, the online CPU times of the new Lagrangian ROMs are orders of magnitude lower than the CPU time of the corresponding FOM.

There are numerous research directions that could provide improvements both in the efficiency and the accuracy of the new Lagrangian ROMs. Probably the most important research avenue is the investigation of ROM closure models for the new Lagrangian ROMs. Indeed, the new Lagrangian ROMs improved the standard Eulerian ROM’s accuracy solely by using a ROM basis constructed with the new Lagrangian inner products. We plan to further increase the accuracy of the new Lagrangian ROMs by adding ROM closure models for the effect of the discarded ROM modes, e.g., data-driven ROM closure models [55, 55, 86] or eddy viscosity ROM closure models [75]. Another potential research direction is finding the optimal $\alpha$ value in the new Lagrangian $\alpha$-ROM. Although the $\alpha$-ROM with higher $\alpha$ values yielded the most accurate results in our numerical investigation, finding the optimal $\alpha$ value is still an open question. To find this optimal parameter value, one could try to extend to the Lagrangian setting the mathematical tools developed for Eulerian ROMs [42, 88].

Another research avenue is the extension of the new Lagrangian ROMs and the novel Lagrangian inner products to the computation of other structures that characterize transport and mixing. For example, instead of geometric approaches (such as the FTLE field), one could approximate probabilistic measures, such as the almost invariant sets [22, 30]. Finally, although the new Lagrangian ROMs dramatically reduced the computational cost of velocity field computation, we intend to explore different approaches for speeding up the FTLE field computation from available ROM velocity data. To this end, we plan to use the
new Lagrangian ROMs in conjunction with the algorithms that have been recently proposed in [59, 80].

Bibliography


Chapter 8

Conclusions and Future Work

8.1 Conclusions

In this dissertation, we mainly focus on two different perspectives in improving projection-based ROMs: (I) ROM closure modeling and (II) ROM basis. For (I), we investigate the DD-VMS-ROM in Chapter 2 and the physically constrained DD-VMS-ROM in Chapter 3. Also, we investigate the numerical accuracy of a hybrid data-driven ROM for high Reynolds number turbulent channel flows in Chapter 5. In addition, we prove the verifiability for the DD-VMS-ROM and hybrid data-driven ROM in Chapter 4 and Chapter 6, respectively. For (II), we investigate the numerical accuracy for the Lagrangian ROM in Chapter 7.

In Chapter 2, we propose a new data-driven variational multiscale reduced order model (DD-VMS-ROM) framework. 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, where one ROM closure model is constructed 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, where two ROM closure models are constructed: one for the interaction between the resolved large and resolved small scales, and another 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; (ii) a 2D flow past a circular cylinder; (iii) the quasi-geostrophic equations; (iv) a 2D flow over a backward facing step; 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.

In Chapter 3, we investigate the physically constrained DD-VMS-ROM (denoted as CDD-VMS-ROM or CDDC-ROM) for the quasi-geostrophic equations (QGE). These ideas are
based on previous work in Chapter 2 and in [6]. The CDD-VMS-ROM incorporates a negative semidefiniteness constraint into the optimization problem to preserve a fundamental property of the linear operator in the DD-VMS-ROMs. The QGE are challenging equations that exhibit complex spatiotemporal behavior and the numerical results imply that the DD-VMS-ROM and CDD-VMS-ROM can significantly improve the G-ROM performance by adding a closure term (derived by optimization) to the linear component of the G-ROM. For a ROM with 10 POD modes, the DD-VMS-ROM lowered the error in the mean streamfunction (compared to the G-ROM) by a factor of about 600; similarly, the CDD-VMS-ROM lowered the error by a factor of about 2000.

In Chapter 4, we proved that a recently introduced data-driven ROM closure model (i.e., the DD-VMS-ROM [7, 12]) is verifiable. Specifically, we defined a ROM closure model as verifiable if the ROM error is bound (in some norm) by the ROM closure model error. Also, we showed numerically that the DD-VMS-ROM closure is verifiable: in particular, 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.

In Chapter 5, we investigate the hybrid data-driven ROM in the numerical simulation of three-dimensional channel flow at Reynolds number $13,750$. The hybrid data-driven ROM incorporates both functional and structural closure modeling strategies, i.e., the data-driven variational multiscale ROM approach and the artificial viscosity ROM approach, respectively, and it yields more accurate results than both the DD-VMS-ROM and the artificial viscosity Galerkin ROM (AV-G-ROM) in kinetic energy and second-order statistics approximations.

In Chapter 6, we extend the verifiability proof of the DD-VMS-ROM (presented in Chapter 4) to the hybrid data-driven ROM. In particular, we show theoretically that the hybrid data-driven ROM error is bound by its ROM closure model error.

In Chapter 7, we propose Lagrangian ROMs that use new Lagrangian inner products to build the ROM basis. In these Lagrangian inner products, Lagrangian data “steers” the resulting Lagrangian ROM basis toward an accurate approximation of Lagrangian quantities, whereas Eulerian data helps the Lagrangian ROM basis yield an accurate approximation of Eulerian quantities. In our numerical investigation of the QGE, we showed that the new Lagrangian ROMs’ numerical accuracy is orders of magnitude higher than the standard
Eulerian ROM accuracy in approximating both Eulerian fields (i.e., the velocity field) and Lagrangian fields (i.e., the FTLE field). We emphasize that, since the new Lagrangian ROMs did not employ any closure modeling, the dramatic increase in the new Lagrangian ROMs’ accuracy is entirely due to the new Lagrangian inner products used to build the Lagrangian ROM basis.

8.2 Future Work

We plan to further investigate several other research directions.

**Numerical Analysis of the Hybrid Data-Driven ROM** In Theorem 6.3 of Chapter 6, we proved verifiability of the hybrid data-driven ROM by assuming it is equipped with a linear ansatz (6.7) and a physical constraint (4.25). In the next step, we plan to investigate the verifiability of the hybrid data-driven ROM without the physical constraint (4.25).

**Dynamic Mode Decomposition** In [13] (Chapter 7), we propose a new Lagrangian type ROM, FTLE-ROM: we use finite time Lyapunov exponents (FTLE) data for proper orthogonal decomposition (POD) basis generations. Numerical results show that for QGE, FTLE-ROM is statistically more accurate than the standard G-ROM. On the other hand, realistic applications, e.g., ocean modeling, require short time accuracy for ROMs. The chaotic nature of QGE hints that the POD basis which captures the largest possible kinetic energy on average [4] might be insufficient for such ROMs. In the future, we plan to apply the dynamic mode decomposition (DMD) as an alternative for ROM basis since DMD provides a set of modes that contain temporal information and share correlated spatial structures [3].

**Closure Learning via Non-intrusive Methods** In the DD-VMS-ROM/hybrid ROM (Chapters 2 and 5), we solve the optimization problems for closure terms that are assumed to be linear/quadratic functions. However, these models may lack parametric robustness. As a remedy, we can consider non-intrusive methods developed in areas of non-intrusive ROMs, e.g., operator inference [2, 8] and radial basis function (RBF) multi-dimensional interpolation [11]. In the future, we plan to investigate applying non-intrusive methods in ROM closure modeling.
**Data Assimilation**  Data assimilation is a method which combines the observational data and model output to improve the model [1]. In the future, we plan to investigate ROM closure modeling in four dimensional variational (4DVAR) data assimilation to identify the best estimate of the closure terms [5, 9, 10].

**Bibliography**


Appendices
List of All the References


LIST OF ALL THE REFERENCES


