Reduced Order Model Closures: A Brief Tutorial

William Snyder, Changhong Mou, Honghu Liu, Omer San, Raffaella De Vita, and Traian Iliescu

1 Introduction

Reduced order models (ROMs) are computational models whose dimensions are orders of magnitude lower than the dimensions of the full order models (FOMs) (i.e., models obtained from classical numerical methods, e.g., the finite element method). Because ROMs are relatively low-dimensional, their computational cost is orders of magnitude lower than the computational cost of FOMs. Thus, ROMs represent a promising alternative to FOMs in computationally intensive applications, e.g., digital twins of wind farms and real-time surgical procedures. ROMs are expected to play a key role in establishing mathematical modeling foundations for digital twins of many engineering, healthcare, and environmental systems. Indeed, if ROM results are nearly indistinguishable from the corresponding FOM results, then they can contribute as predictive tools in emerging digital twin infrastructures. However, despite being successfully used in simple, academic test problems, ROMs have not made a significant impact in complex, practical applications.

One of the main hurdles in the ROMs' development is their notorious inaccuracy when they are used in the *under-resolved* regime, i.e., when the ROM's dimension (i.e., its number of degrees of freedom (DOF)) is not large enough to capture

W. Snyder · H. Liu · R. De Vita · T. Iliescu (✉)
Virginia Tech, Blacksburg, VA, USA
e-mail: swilli9@vt.edu; hhliu@vt.edu; devita@vt.edu; iliescu@vt.edu

C. Mou
University of Wisconsin-Madison, Madison, WI, USA
e-mail: cmou3@wisc.edu

O. San
Oklahoma State University, Stillwater, OK, USA
e-mail: osan@okstate.edu

the complex dynamics of the underlying system. To illustrate the under-resolved regime, think of the numerical simulation of the flow around a wind farm. This simulation with a FOM (e.g., the finite element method) generally requires millions (if not billions) of DOF. Thus, performing shape optimization or real-time control of the wind farm flow, which would require many individual FOM runs, is not feasible. Replacing the costly FOM with a ROM would be a natural choice. However, in order to represent the turbulent flow dynamics in the wind farm simulation would require thousands or tens of thousands of DOF in the ROM. Despite the ROM's cost being much lower than the FOM cost, it is still too high to allow the use of the ROM in real-time control applications, where thousands of ROM runs would be required. Thus, a practical choice would be to use much cheaper ROMs, i.e., ROMs with much fewer (e.g., hundreds or even tens) DOF. However, these low-dimensional ROMs, although computationally efficient (and, therefore, practical), generally yield inaccurate results. The reason is simple: these ROMs do not have enough DOF to represent the complex dynamics of a complex flow such as the turbulent wind farm flow.

The above discussion yields the following two important conclusions:

1. The under-resolved ROM regime is critical in realistic, complex applications.
2. Under-resolved ROMs produce inaccurate results.

These conclusions naturally lead to the following question:

? Q0

How do we fix the under-resolved ROMs?

The answer to Q0 is simple:

⌣ A0

We develop good ROM closure models, i.e., correction terms that increase the standard ROM's accuracy.

To our knowledge, the first (and only) survey of ROM closure models was performed in [1], where the authors discuss dozens of ROM closures for fluids that have been developed over the last four decades. We are not aware, however, of a tutorial on ROM closures. This paper takes a first step at filling that gap.

This brief tutorial on ROM closures (also known as parameterizations [5, 10, 11, 16, 17, 34, 53] and hidden dynamics [40, 41]) is structured as a sequence of simple questions and answers that lead the reader from a simple PDE to projection ROMs, and then to ROM closures. Our paper is aimed at first year graduate students and advanced undergraduate students. Thus, we strive to keep the technical details to a level that is easily understood by students with a standard background in differential

equations and numerical methods. We also emphasize that our goal in this tutorial is not to explain the "how," but the "why." That is, we carefully explain the principles used to develop ROM closures, without focusing on particular approaches (which are carefully discussed in [1]).

The rest of the paper is organized as follows: In Sect. 2, we illustrate the ROM closure modeling concept for a three-dimensional toy problem. In Sect. 3, we present the general algorithm used to develop the classical Galerkin ROM. In Sect. 4, we first present the ROM closure problem, and then we discuss its solution, i.e., the ROM closure model. In Sect. 5, we construct the data-driven variational multiscale ROM, in which available data is used to build the ROM closure model. In Sect. 6, we illustrate how closure modeling can significantly increase the ROM accuracy in the numerical simulation of fluid flows. In Sect. 7, we survey current mathematical results for ROM closure modeling. Finally, in Sect. 8, we present conclusions and future research avenues.

2 A Crash Course in ROM Closure: A Toy Problem

Before carefully presenting the ROM closure modeling in the next sections, we illustrate the underlying *concepts* and *principles* for a *toy problem*. These concepts and principles are broadly illustrated in the schematic in Fig. 1, which is adapted from Fig. 1 in [3].

To present our toy problem, we first assume that the FOM solution, u^{FOM}, can be accurately approximated by only three ROM basis functions:

$$u^{FOM}(x, t) \approx a_1(t)\varphi_1(x) + a_2(t)\varphi_2(x) + a_3(t)\varphi_3(x), \qquad (1)$$

where $\varphi_1, \varphi_2, \varphi_3$ are the ROM basis functions, and a_1, a_2, a_3 are the sought time-dependent coefficients. Of course, for complex systems, one should use many more (e.g., hundreds and even thousands of) ROM basis functions to accurately approximate u^{FOM}. However, to graphically illustrate the need for closure modeling in our toy problem, we assume that three ROM basis functions are enough.

Next, we use the three ROM basis functions in the Galerkin framework to construct the *Galerkin ROM (G-ROM)*. Details regarding the G-ROM construction are given in Sect. 3. For the purpose of the toy problem illustration in this section, we just note that the resulting G-ROM is a three-dimensional dynamical system that can be written as follows:

$$\begin{bmatrix} \dot{a}_1 \\ \dot{a}_2 \\ \dot{a}_3 \end{bmatrix} = \begin{bmatrix} F_1(a_1, a_2, a_3) \\ F_2(a_1, a_2, a_3) \\ F_3(a_1, a_2, a_3) \end{bmatrix}, \qquad (2)$$

where F_1, F_2, and F_3 are the components of the ROM operators, e.g., vectors, matrices, and tensors, which are presented in Sect. 3. Since the three ROM basis
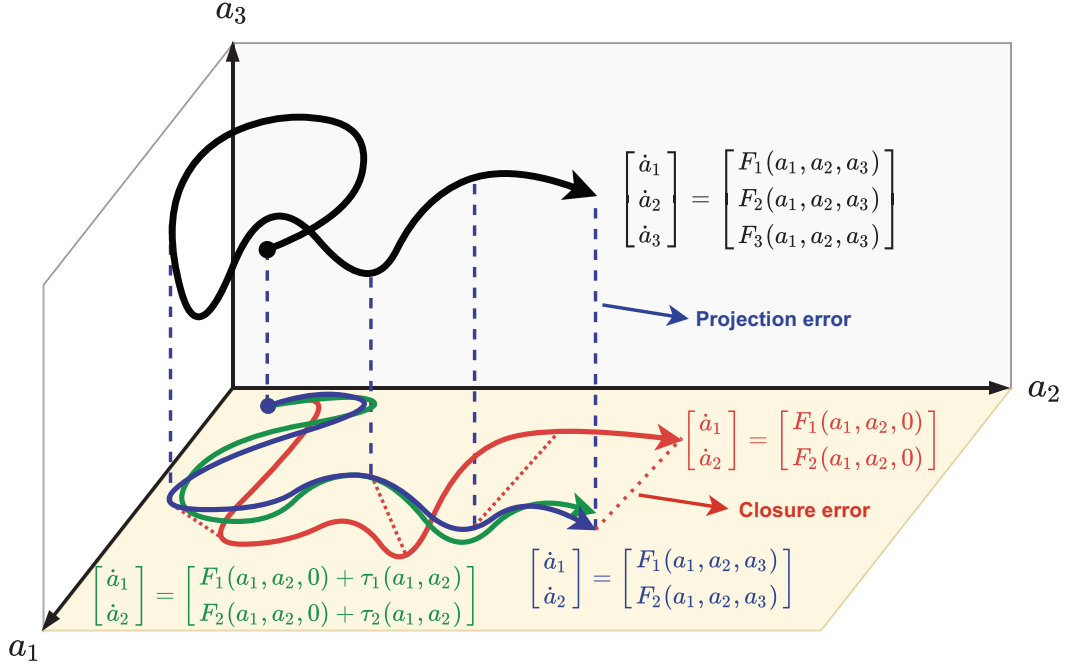
Fig. 1 A schematic representation of the ROM closure modeling for a three-dimensional toy problem. The goal is to reduce the three-dimensional G-ROM (2) (black curve and equations) to the most accurate two-dimensional ROM. The I-ROM (3) (blue curve and equations) is the most accurate ROM obtained in the Galerkin framework, but it is not closed (since it depends on a_3). The two-dimensional G-ROM (4) (red curve and equations) is closed, but it is not accurate (since we simply ignore the a_3 contribution). The two-dimensional G-ROM supplemented with a closure model (5) (green curve and equations) is closed and more accurate than the two-dimensional G-ROM (4) since the closure terms $\tau_1(a_1, a_2)$ and $\tau_2(a_1, a_2)$ aim at steering the green curve toward the blue curve

functions yield an accurate approximation of the FOM solution in (1), the three-dimensional G-ROM in (2) is expected to yield an accurate approximation to $\boldsymbol{u}^{FOM}$. That is, solving the three-dimensional G-ROM (2) for a_1, a_2, a_3, and then plugging these values back into (1) yields an accurate approximation to $\boldsymbol{u}^{FOM}$. In Fig. 1, the time evolution of the solution of the accurate three-dimensional G-ROM (2) is represented as the black curve.

At this point, we invoke the need to *reduce the computational cost* of the three-dimensional G-ROM (2). Specifically, we aim at constructing a *two-dimensional* ROM that is as accurate as possible (preferably, as accurate as the three-dimensional G-ROM (2)). For our toy problem (1), this amounts to constructing a dynamical system for a_1 and a_2 (assuming that the first two ROM basis functions dominate the third, as is often the case; see Sect. 3).

Of course, reducing the ROM dimension from three to two does not yield such a great reduction of computational time. We emphasize, however, that we consider this reduction only to illustrate the ROM closure modeling concept for our toy problem. In practical settings, ROMs reduce the FOM dimension by orders of magnitude.

The most natural way to construct an accurate two-dimensional ROM is to keep only the first two equations in (2) and discard the third equation:

$$\begin{bmatrix} \dot{a}_1 \\ \dot{a}_2 \end{bmatrix} = \begin{bmatrix} F_1(a_1, a_2, a_3) \\ F_2(a_1, a_2, a_3) \end{bmatrix}. \tag{3}$$

Mathematically, this amounts to first using a Galerkin expansion for all three ROM basis functions (i.e., using (1)), and then using a Galerkin projection onto only the first two basis functions (instead of projecting onto all three basis functions, as done in (2)).

In Fig. 1, the time evolution of the solution of the efficient, two-dimensional ROM (3) is represented as the blue curve. Of course, since we perform a Galerkin projection only onto the first two basis functions, we incur an error, which we denote as the (Galerkin) *projection error* (the blue dashed lines in Fig. 1). Nevertheless, it stands to reason that, in the Galerkin framework with the basis $\{\varphi_1, \varphi_2, \varphi_3\}$, the two-dimensional ROM (3) is the most accurate two-dimensional ROM we can hope to get. This is why we call the two-dimensional ROM (3) the *ideal ROM (I-ROM)*. However, the two-dimensional I-ROM (3) has a big problem: It is *not closed* since the equations for a_1 and a_2 depend on a_3. *This is the ROM closure problem.*

So how do we solve the ROM closure problem? *The easiest way to solve the ROM closure problem is to simply ignore it*. That is, we can simply ignore the a_3 contribution to the dynamics in (3):

$$\begin{bmatrix} \dot{a}_1 \\ \dot{a}_2 \end{bmatrix} = \begin{bmatrix} F_1(a_1, a_2, 0) \\ F_2(a_1, a_2, 0) \end{bmatrix}. \tag{4}$$

The ROM in (4) is two-dimensional and closed (since the equations depend only on a_1 and a_2). In Fig. 1, the time evolution of the solution of this two-dimensional ROM (4) is represented as the red curve. Of course, since in (4) we simply ignored the a_3 contribution to the correct dynamics of a_1, a_2 given by (3), we incur an error, which is generally called the *closure error* (the red dashed lines in Fig. 1).

Remark 1 (Galerkin Closure is a Relative Concept) We note that if we start with just two ROM basis functions φ_1 and φ_2, the Galerkin ROM framework (which is presented in Sect. 3 and outlined in Algorithm 1) yields a two-dimensional G-ROM that satisfies exactly the equations in (4). Thus, *the ROM closure concept is relative to the ROM space used in the Galerkin framework*:

- If we start with two basis functions, the Galerkin method yields the two-dimensional G-ROM (4), which is closed.
- If, however, we start with the larger (three-dimensional) ROM space spanned by φ_1, φ_2, and φ_3, the discussion in this section shows that the most accurate two-dimensional ROM obtained by a direct truncation of the three-dimensional G-ROM (2) (i.e., the I-ROM (3)) is not closed.

Remark 2 (Galerkin Closure is a General Concept) We emphasize that, although our discussion focuses exclusively on ROMs, *the Galerkin closure is a general concept that is associated with the classical Galerkin framework*. Thus, there is no surprise that, over half a century, closure has been addressed in different contexts: large eddy simulation (LES) [6], variational multiscale (VMS) methods [24], subgrid-scale (SGS) methods [21, 33], and nonlinear Galerkin (NG) methods [20].

At this point, it is probably a good idea to summarize our discussion. As illustrated in the schematic in Fig. 1, the reader interested in constructing the most accurate two-dimensional G-ROM has reached a *crossroads*:

- On the one hand, the I-ROM (3) is the most accurate two-dimensional ROM that we can get by using the Galerkin framework, but it is not closed.
- On the other hand, the G-ROM (4) is closed, but we are incurring the closure error.

This is as far as the classical Galerkin framework can take us. We're stuck. So what do we do next?

The answer, as many times in numerical methods, is to take a middle of the road approach. Specifically, we construct a *ROM closure model* and add it to the G-ROM (4):

$$
\begin{bmatrix} \dot{a}_1 \\ \dot{a}_2 \end{bmatrix} = \begin{bmatrix} F_1(a_1, a_2, 0) + \tau_1(a_1, a_2) \\ F_2(a_1, a_2, 0) + \tau_2(a_1, a_2) \end{bmatrix},
\tag{5}
$$

where $\tau_1(a_1, a_2)$, $\tau_2(a_1, a_2)$ are the components of the ROM closure model, i.e., correction terms that aim at steering the inaccurate G-ROM (4) as close as possible to the accurate (but not closed) I-ROM (3). In Fig. 1, the time evolution of the solution of the closed ROM (5) is represented as the green curve.

How do we construct the ROM closure model in (5)? We answer this question in Sect. 5. But first, in Sect. 3, we present the main steps in the G-ROM construction.

3 Galerkin ROM (G-ROM)

Over the past four decades, projection ROMs have been used in the numerical simulation of fluid flows [8, 22, 23, 39, 42, 49]. In this tutorial, we exclusively consider projection ROMs that use numerical or experimental data to find the "best" basis, which is then used together with the Galerkin method to construct the ROM. In this section, we present the main steps in the construction of the Galerkin ROM.

To illustrate the Galerkin ROM construction, we start with a generic PDE for the dynamics of a variable of interest, $\boldsymbol{u}$:

$$
\boldsymbol{u}_t = \boldsymbol{f}(\boldsymbol{u}),
\tag{6}
$$

Algorithm 1 Galerkin ROM (G-ROM) algorithm

1: Use numerical or experimental data to construct modes $\{\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_R\}$, which represent the recurrent spatial structures in the system (6).
2: Choose the dominant modes $\{\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_r\}$, $r \leq R$, as ROM basis functions.
3: Use a Galerkin expansion $\boldsymbol{u}_r(\boldsymbol{x}, t) = \sum_{j=1}^r a_j(t)\,\boldsymbol{\varphi}_j(\boldsymbol{x})$.
4: Replace $\boldsymbol{u}$ with $\boldsymbol{u}_r$ in (6), and then on both sides of (6) take the inner product with each mode $\boldsymbol{\varphi}_i$, $i = 1, \ldots, r$. That is, perform a Galerkin projection of the PDE (6) onto the ROM space $\mathbf{X}^r := \text{span}\{\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_r\}$. The obtained *Galerkin ROM (G-ROM)* is of the form

$$\dot{\boldsymbol{a}} = \boldsymbol{F}(\boldsymbol{a}), \tag{7}$$

where $\boldsymbol{a}(t) = (a_i(t))_{i=1,\ldots,r}$ is the vector of coefficients in the Galerkin expansion in step 3 and $\boldsymbol{F}$ comprises the ROM operators.
5: In the offline stage, compute the ROM operators (e.g., vectors, matrices, and tensors), which are preassembled from the ROM basis.
6: In the online stage, repeatedly use the G-ROM (7) for longer time intervals.

---

equipped with appropriate boundary conditions and initial conditions. In Algorithm 1, we list the main steps in the Galerkin ROM construction.

Remark 3 (ROM=d2G) The main steps in the G-ROM (7) construction presented in Algorithm 1 are straightforward. In principle, they are the same steps as those used to construct classical Galerkin methods, e.g., the finite element method (FEM). The fundamental difference between the G-ROM and the FEM is that the former uses a *data-driven basis*, whereas the latter uses a universal basis (i.e., piecewise polynomials). Thus, one could think of the projection ROMs that we discuss in this tutorial as *data-driven Galerkin (d2G) methods*.

Next, we explain some of the steps in Algorithm 1.

ROM Basis (Step 1)
To construct the ROM basis, we first collect *snapshots* from the simulation of the FOM. If we are interested in time prediction (as in the numerical illustration in Sect. 6), the snapshots can be FEM approximations of (6) at the time instances $t_1, \ldots, t_M$, i.e., $\boldsymbol{u}_h^1, \ldots, \boldsymbol{u}_h^M$, respectively. (If (6) depends on parameters, we can also build a ROM basis for parameter prediction [22, 42].) Next, we use these snapshots to construct the modes $\{\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_R\}$, which represent the recurrent spatial structures in the system described by (6). Different approaches can be used to construct the ROM basis functions, e.g., (i) the proper orthogonal decomposition (POD) [8, 23, 31, 49, 51]; (ii) the reduced basis method (RBM) [22, 42]; (iii) the proper generalized decomposition (PGD) [15]; and (iv) clustering [9]. In this tutorial, to fix ideas, we exclusively use the POD to generate the ROM basis.

For a careful presentation of the POD basis, the reader is referred to, e.g., [23] (for a physical presentation) and to [51] (for a mathematical presentation). In this paper, however, we only briefly discuss the *qualitative* properties of the POD basis functions, which we will later use in our numerical illustration in Sect. 6. The reason for our brief qualitative discussion of the POD basis is that *ROM closure modeling*

does not depend on the particular type of ROM basis functions used. That is, our presentation of ROM closure modeling remains the same for any type of ROM basis used in a Galerkin framework, whether it is POD, RBM, or PGD.

The main principle used to construct the G-ROM basis can be stated as follows: *Use the available snapshots to find the ROM basis that "best" represents the system's dynamics.* Since this is the "best" basis, for certain problems, one can hope to use much fewer basis functions to construct the G-ROM than to construct, e.g., FEM models. For example, instead of using millions or even billions of basis functions as in FEM simulations, one can hope to use tens or hundreds basis functions in the G-ROM construction. This choice of "best" basis yields computational models (i.e., ROMs) whose dimension can be orders of magnitude lower than the dimension of FEM models. (This also explains the term "reduced" in the ROM terminology.)

Of course, a natural question is what the "best" ROM basis means. In fact, there are many proposals for the "best" ROM basis, and each proposal yields a different class of ROMs (e.g., POD, RBM, or PGD, to name just a few). For example, given a set of snapshots, the POD basis is the orthonormal basis that yields the minimum projection error with respect to a chosen norm (e.g., the L^2 norm) [51].

However, independent of the approach used to construct them, the ROM basis functions generally share several qualitative features. To illustrate this, in Fig. 2 we plot the Euclidian norm of two POD basis functions, $\boldsymbol{\varphi}_1$ and $\boldsymbol{\varphi}_{10}$, and two FEM basis functions, $\boldsymbol{\phi}_1^h$ and $\boldsymbol{\phi}_{10}^h$, for a 2D flow past a circular cylinder [37]. One can clearly see the significant differences between the POD basis functions (top two plots) and the FEM basis functions (bottom two plots). Indeed, the POD basis functions have global support (i.e., they can be nonzero over the entire computational domain), whereas the FEM basis functions have local support (i.e., they are one at one mesh point and zero everywhere else). To further illustrate the different characteristics of the POD basis, in Fig. 3 we plot the Euclidian norm of two POD basis functions, $\boldsymbol{\varphi}_1$ and $\boldsymbol{\varphi}_{10}$, for soft tissue modeling [48]. Comparing these two POD basis functions with the POD basis functions in the top two plots of Fig. 2, we can clearly see that different physical systems (i.e., the soft tissue in Fig. 3 and the flow in Fig. 2) yield fundamentally different POD basis functions. We emphasize that this is in complete contrast with classical numerical methods, such as the FEM. Indeed, the FEM basis functions are *universal* basis functions, i.e., they have the same shape (piecewise polynomials and local support) for all the problems. In contrast, the POD basis functions (and ROM basis functions in general) change their shape when we change the problem. This can be clearly seen by comparing the top two plots of Fig. 2 with the plots of Fig. 3.

Galerkin ROM Construction (Steps 2–6)
To illustrate the G-ROM construction, we use the Navier-Stokes equations (NSE) as a mathematical model:

$$\frac{\partial \boldsymbol{u}}{\partial t} - Re^{-1}\Delta \boldsymbol{u} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} + \nabla p = \boldsymbol{0}\,, \tag{8}$$
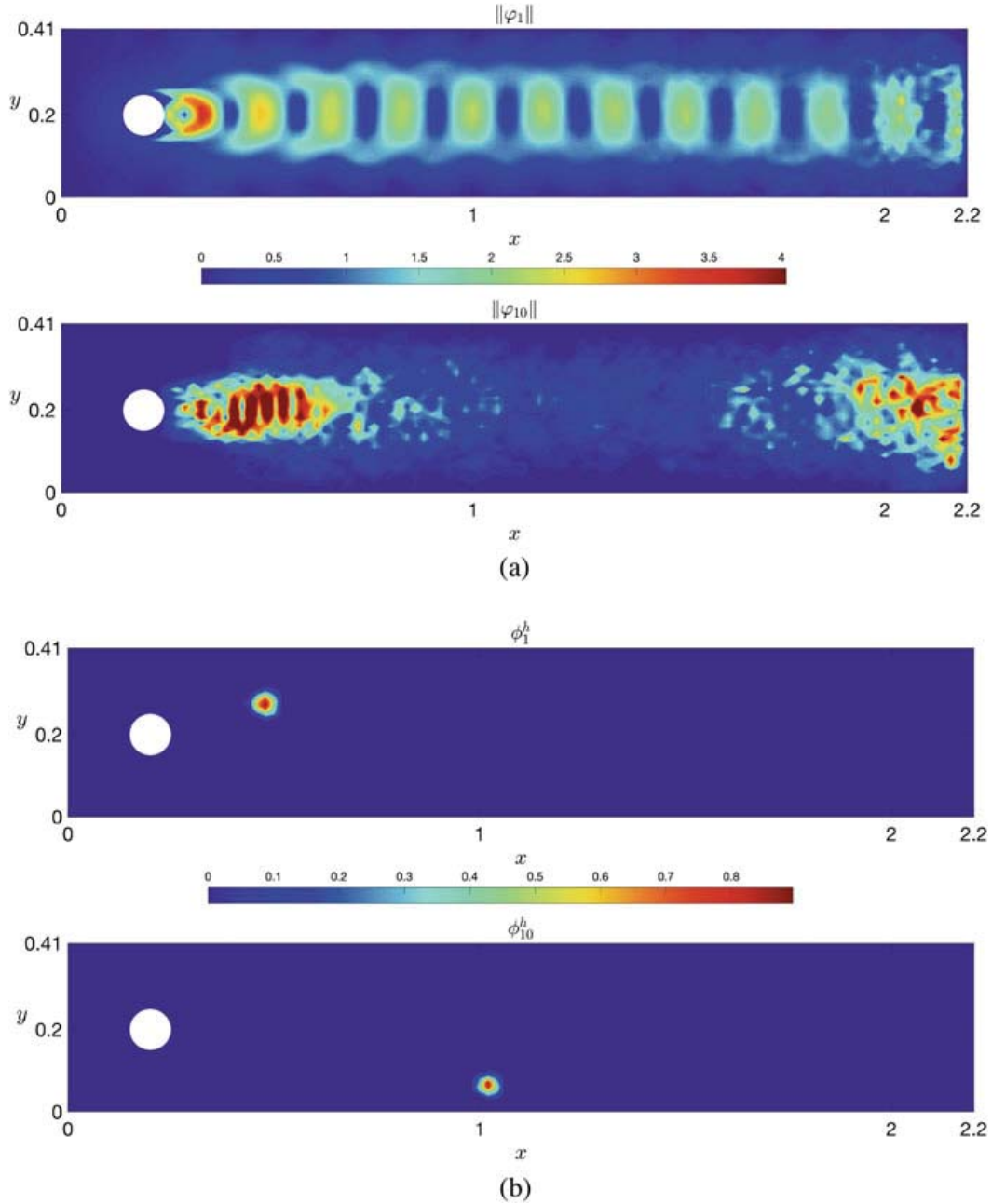
Fig. 2 2D flow past a circular cylinder: (**a**) Euclidian norm of ROM basis functions φ_1 and φ_{10} at mesh points. (**b**) Euclidian norm of FEM basis functions ϕ_1^h and ϕ_{10}^h at mesh points. Note that the ROM basis functions are fundamentally different from the FEM basis functions: The former have global support, whereas the latter have local support

$$\nabla \cdot \boldsymbol{u} = 0 , \tag{9}$$

where $\boldsymbol{u}$ is the velocity, p the pressure, and Re the Reynolds number. We consider the NSE posed on a bounded spatial domain in either $\mathbb{R}^2$ or $\mathbb{R}^3$, and supplemented with homogeneous Dirichlet boundary conditions and an appropriate
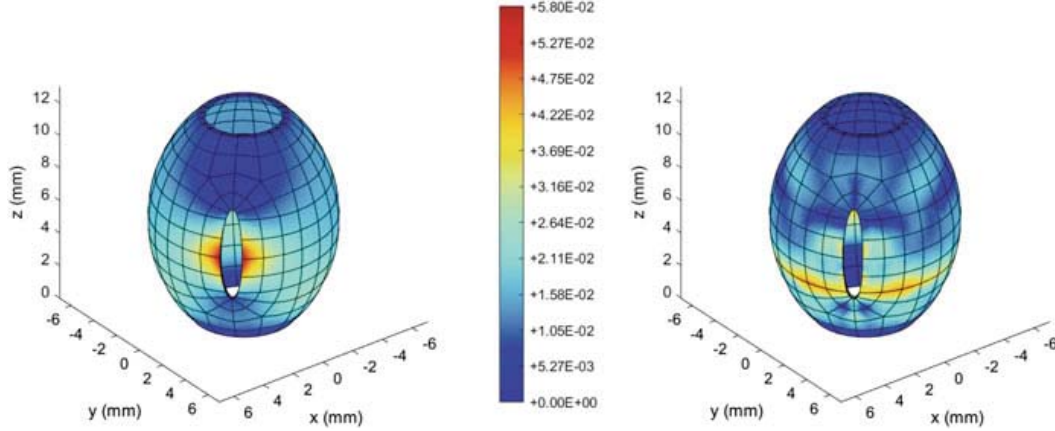
Fig. 3 Soft tissue modeling: Euclidian norm of ROM basis functions $\boldsymbol{\varphi}_1$ and $\boldsymbol{\varphi}_{10}$ at mesh points

initial condition. The NSE (8)–(9) can be cast in the general form (6) by choosing $\boldsymbol{f}(\boldsymbol{u}) = Re^{-1}\Delta\boldsymbol{u} - \boldsymbol{u} \cdot \nabla\boldsymbol{u}$ (after applying the Leray projection, which maps the vector field into the divergence-free subspace of the underlying state space) [50].

To construct the G-ROM for the NSE, we follow Steps 2–6 in Algorithm 1. That is, we choose the first r basis functions from the modes constructed in Step 1, use a Galerkin truncation

$$\boldsymbol{u}_r(\boldsymbol{x}, t) = \sum_{j=1}^{r} a_j(t)\,\boldsymbol{\varphi}_j(\boldsymbol{x}), \tag{10}$$

replace $\boldsymbol{u}$ with $\boldsymbol{u}_r$ in the NSE (8), and project the resulting PDE onto the ROM space, X^r. Furthermore, we apply the divergence theorem to the diffusion term and the pressure term. This yields the G-ROM [37]:

$$\dot{\boldsymbol{a}} = A\,\boldsymbol{a} + \boldsymbol{a}^\top B\,\boldsymbol{a}, \tag{11}$$

where $\boldsymbol{a}(t)$ is the vector of unknown coefficients $a_j(t)$, $1 \leq j \leq r$ in the Galerkin expansion (10). The ROM operator A in (11) is an $r \times r$ matrix that corresponds to the diffusion term in the NSE (i.e., $-Re^{-1}\Delta\boldsymbol{u}$) and has entries

$$A_{im} = -Re^{-1}\left(\nabla\boldsymbol{\varphi}_m, \nabla\boldsymbol{\varphi}_i\right), \quad 1 \leq i, m \leq r, \tag{12}$$

where $(\cdot, \cdot)$ denotes the L^2 inner product. The ROM operator B in (11) is an $r \times r \times r$ tensor that corresponds to the nonlinear term in the NSE (i.e., $\boldsymbol{u} \cdot \nabla\boldsymbol{u}$) and has entries

$$B_{imn} = -\left(\boldsymbol{\varphi}_m \cdot \nabla\boldsymbol{\varphi}_n, \boldsymbol{\varphi}_i\right), \quad 1 \leq i, m, n \leq r. \tag{13}$$

We note that the pressure term in the G-ROM (11) vanishes 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, 22].

Once the matrix A and tensor B are assembled in the offline stage, the G-ROM (11) is a relatively low-dimensional, efficient dynamical system that can be used in the online stage for longer time intervals (or more parameter values, e.g., Re [22, 42]).

4   The Closure Problem and Its Solution: The Closure Model

This section has two goals: In Sect. 4.1, we motivate the need for ROM closure modeling in the under-resolved regime, i.e., we describe the ROM closure problem. In Sect. 4.2, we show how to solve the ROM closure problem, i.e., we show how to construct a ROM closure model. To this end, we give the definition of the ROM closure model, show that using the exact closure model (i.e., using the ideal ROM) increases the ROM accuracy, and finally outline the main steps in the ROM closure model construction.

4.1   The Closure Problem

The G-ROM (11) constructed in Sect. 3 is appealing from the computational point of view: The G-ROM can significantly reduce the dimension (and, thus, the computational cost) of classical numerical discretization (e.g., FEM) models by orders of magnitude. So one can ask the following natural question:

? Q1

What is wrong with G-ROM?

The short answer to Q1 is: It depends on the resolution. Specifically:

> A1

It depends on whether we are in the *resolved* regime or the *under-resolved* regime.

- In the *resolved* regime (i.e., when there are enough ROM basis functions $\{\varphi_1, \ldots, \varphi_r\}$ to accurately represent the underlying dynamics), the G-ROM produces accurate results.

- In the *under-resolved* regime (i.e., when there are not enough ROM basis functions $\{\varphi_1, \ldots, \varphi_r\}$ to accurately represent the underlying dynamics), the G-ROM produces inaccurate results.

But then one can ask the following questions:

? Q2

Why is the under-resolved regime important? Why do we need to worry about it?

> A2

Many important applications (e.g., atmospheric boundary layer flows, digital twins of wind farms, and anisotropic and heterogeneous biological tissues) are centered around *multiscale* systems that require a large number of ROM basis functions. However, to ensure a low computational cost in these applications, under-resolved G-ROMs are generally used.

4.2 The Closure Model

In Sect. 4.1, we defined the ROM closure problem, and we explained why it is important. In this section, we present the solution to the ROM closure problem. That is, we answer the following question:

? Q3

What is the solution to the closure problem?

> A3

The solution to the closure problem is the closure model. That is, replace the G-ROM (11) with

$$\dot{a} = F(a) + \tau(a), \tag{14}$$

where $\tau(a)$ is the closure model, which represents the effect of the discarded ROM modes $\{\varphi_{r+1}, \ldots, \varphi_R\}$ on the ROM dynamics.

Note that A3 is a vague definition, which begs the following questions: What exactly does "model the effect" mean? What exactly does $\tau(a)$ in (14) actually model?

Answering these natural questions is not straightforward. To do so, we need to *extend the Galerkin framework*. This sounds like a daunting task, but it turns out to be relatively simple. The "trick" is to *rethink the space* we use in the Galerkin framework:

In the resolved regime, the ROM space $X^r := \text{span}\{\varphi_1, \ldots, \varphi_r\}$ is the only space we will ever need, since everything happens in X^r. Thus, in the resolved regime, G-ROM should (and generally does) work just fine.

However, in the under-resolved regime we need *two* spaces: (i) the *resolved space* X^r, and (ii) the *unresolved space* $X^{r'} := \text{span}\{\varphi_{r+1}, \ldots, \varphi_R\}$. To keep the ROM dimension (and, therefore, its computational cost) low, we want to work in the resolved space, X^r. However, to increase the ROM accuracy, we should do our best to model the contribution to the ROM dynamics made by the dynamics in the unresolved space, $X^{r'}$. But this sounds like a lot of work (both in terms of modeling and computation). So the following is a natural question:

? Q4

Does $X^{r'}$ have a significant effect on the ROM dynamics?

> A4

Yes.

The answer A4 is simple. In Sect. 4.2.1, we introduce the ideal ROM, which adds the *exact* closure term to the classical G-ROM. The ideal ROM results clearly show why the effect of $X^{r'}$ should be modeled. Specifically, we show that the ideal ROM results are dramatically more accurate than the G-ROM results. Thus, we conclude that modeling the exact ROM closure term is beneficial to ROM accuracy.

4.2.1   The Ideal ROM (I-ROM)

To present the ideal ROM, we first need to define the spaces of resolved ROM scales (i.e., X^r) and unresolved ROM scales (i.e., $X^{r'}$). To this end, we extend the variational multiscale (VMS) framework proposed by Hughes and his group two decades ago in the FEM context. We note, however, that there are other ways of defining the spaces of resolved and unresolved ROM scales, e.g., spatial filtering [37].

First, we leverage the orthonormality of the ROM basis functions and construct the two orthogonal spaces, X^r and $X^{r'}$, as follows:

$$X^r := \mathrm{span}\{\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_r\} \quad \text{and} \quad X^{r'} := \mathrm{span}\{\boldsymbol{\varphi}_{r+1}, \ldots, \boldsymbol{\varphi}_R\}. \quad (15)$$

The space X^r represents the space of the *resolved ROM scales*, i.e., the spatial scales that are explicitly approximated by a given r-dimensional ROM. In contrast, the space $X^{r'}$ represents the space of the *unresolved ROM scales*, i.e., the spatial scales that are not explicitly approximated by the chosen ROM. We note that since the ROM basis functions are generally ordered from the most important to the least important (with respect to a physical criterion, e.g., kinetic energy [23]), the decomposition in (15) is natural. We also note that since we are concerned with the under-resolved regime that often occurs in practical applications, we consider the case when $r \ll R$.

The next step in the construction of the ideal ROM is to extend the Galerkin framework to the space $X^R := X^r \oplus X^{r'}$, which is the *maximal ROM space* (i.e., the space spanned by all the snapshots). Thus, we use the ROM approximation of both resolved and unresolved scales, i.e., we utilize $\boldsymbol{u}_R \in X^R$ defined as

$$\boldsymbol{u}_R = \sum_{j=1}^{R} a_j \boldsymbol{\varphi}_j = \sum_{j=1}^{r} a_j \boldsymbol{\varphi}_j + \sum_{j=r+1}^{R} a_j \boldsymbol{\varphi}_j = \boldsymbol{u}_r + \boldsymbol{u}', \quad (16)$$

where $\boldsymbol{u}_r \in X^r$ represents the *resolved ROM component* of $\boldsymbol{u}$, and $\boldsymbol{u}' \in X^{r'}$ represents the *unresolved ROM component* of $\boldsymbol{u}$. Next, we plug $\boldsymbol{u}_R$ in the generic equation (6), project the resulting equation onto X^r, and use the ROM basis orthogonality to show that $(\boldsymbol{u}_{R,t}, \boldsymbol{\varphi}_i) = (\boldsymbol{u}_{r,t}, \boldsymbol{\varphi}_i)$, $\forall i = 1, \ldots, r$, where $\boldsymbol{u}_{R,t}$ and $\boldsymbol{u}_{r,t}$ are the time derivatives of $\boldsymbol{u}_R$ and $\boldsymbol{u}_r$, respectively. Following these steps, we obtain the *ideal ROM (I-ROM)*:

$$(\boldsymbol{u}_{r,t}, \boldsymbol{\varphi}_i) = (\boldsymbol{f}(\boldsymbol{u}_r), \boldsymbol{\varphi}_i) + \underbrace{(\boldsymbol{f}(\boldsymbol{u}_R), \boldsymbol{\varphi}_i) - (\boldsymbol{f}(\boldsymbol{u}_r), \boldsymbol{\varphi}_i)}_{\boldsymbol{\tau}^{I-ROM} = \text{ideal ROM closure term}}, \quad \forall i = 1, \ldots, r. \quad (17)$$

The last two terms in (17) yield the *ideal ROM closure term*, $\boldsymbol{\tau}^{I-ROM}$, which represents the effect of the discarded ROM modes $\{\boldsymbol{\varphi}_{r+1}, \ldots, \boldsymbol{\varphi}_R\}$ onto the dynamics of the resolved ROM scales, $\boldsymbol{u}_r$. Using the expansion (16), the I-ROM (17) can be written as the following dynamical system for the vector of ROM coefficients of the resolved scales:

$$\dot{\boldsymbol{a}} = \boldsymbol{F}(\boldsymbol{a}) + \boldsymbol{\tau}^{I-ROM}(a_1, \ldots, a_r, a_{r+1}, \ldots, a_R). \quad (18)$$

The above discussion clearly shows that, from a mathematical point of view, the correct equations satisfied by the coefficients of the resolved ROM scales are the I-ROM equations (18) instead of the G-ROM equations (11). However, we need to
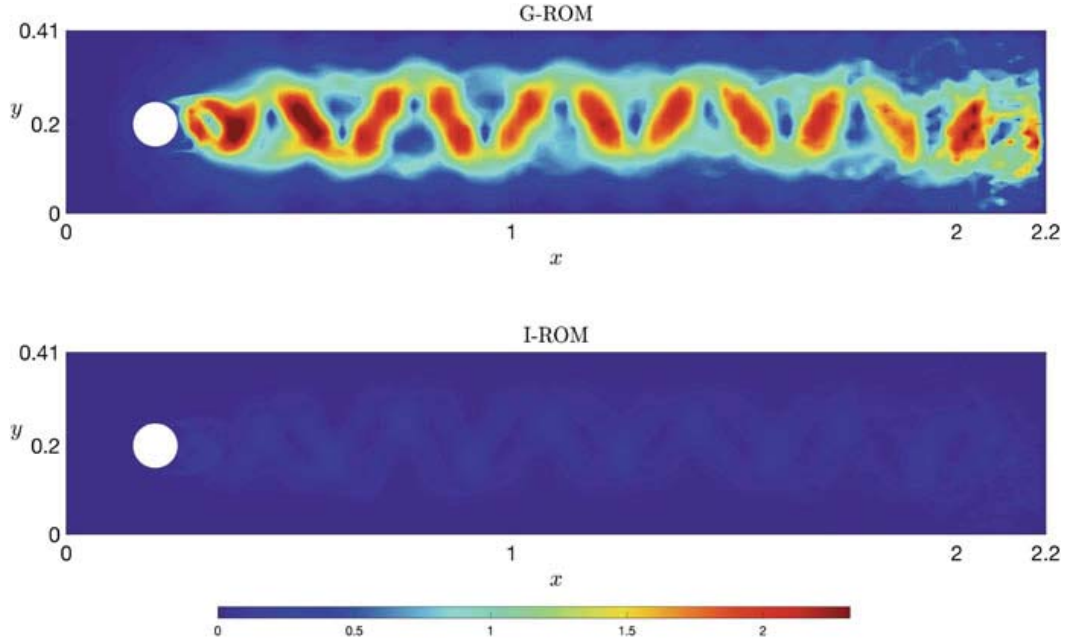
Fig. 4 2D flow past a circular cylinder. The Euclidian norm of the error, $\boldsymbol{u}^{FOM} - \boldsymbol{u}^{ROM}$, at mesh points for G-ROM (11) (top) and I-ROM (17) (bottom). The I-ROM error is significantly lower than the G-ROM error, which illustrates the potential benefit of ROM closure modeling

ask ourselves whether this mathematical framework has a *practical impact* (i.e., we need to ask question Q4). Specifically, we need to check whether the I-ROM results are better than the G-ROM results.

In Fig. 4, we present results for the I-ROM (18) and the G-ROM (11) in the numerical simulation of a two-dimensional flow past a circular cylinder. These plots clearly show that the I-ROM performs significantly better than the classical G-ROM. Thus, these results suggest that including a model for the I-ROM closure term, $\boldsymbol{\tau}^{I-ROM}$, could increase the ROM accuracy.

Remark 4 (The Closure Model Increases Accuracy) There is a lot of confusion in the ROM community (and not only) regarding the role of the closure model. In this section, we tried to emphasize that the main role of the ROM closure model is to increase the *accuracy* of the G-ROM. Indeed, in Eq. (14), adding the closure term, $\boldsymbol{\tau}(\boldsymbol{a})$, to the classical G-ROM yields a more accurate model (in the extended Galerkin framework).

That being said, in many important practical applications (e.g., convection-dominated flows), the G-ROM's inaccuracy often manifests itself in the form of *spurious numerical oscillations*. Thus, a popular misconception (at least in computational fluid dynamics) is that the only role of the ROM closure model is to eliminate/alleviate these numerical oscillations, i.e., to increase the numerical stability of the G-ROM.

However, we emphasize that, while numerical stability of the model is *necessary* (indeed, if the model is accurate, then it has to be stable), it is *not sufficient*. For

example, we can add a very large stabilization term to the classical G-ROM. This, most likely, will stabilize the model, but will also degrade its accuracy.

To summarize, we emphasize that ROM closure modeling is not simply about adding numerical stabilization. Instead, ROM closure modeling is about adding the *"right"* amount of numerical stabilization (i.e., the amount of stabilization that makes the model accurate).

4.2.2 Closure Model Construction

The I-ROM results in Sect. 4.2.1 clearly show that the effect of $X^{r'}$ should be modeled. We emphasize, however, that the I-ROM itself does *not* represent a practical solution since it depends on the coefficients of the discarded ROM modes, $a_{r+1}, \ldots, a_R$, which we do not model in our ROM (since we work in X^r).

? Q5

How do we make the I-ROM (18) practical?

> A5

We construct a closure model, τ, which is an approximation in X^r of the I-ROM closure term, τ^{I-ROM}:

$$\tau^{I-ROM}(a_1, \ldots, a_r, a_{r+1}, \ldots, a_R) \approx \tau(a_1, \ldots, a_r). \tag{19}$$

Since τ in (19) lives in X^r, it can be computed with the available ROM data, and, thus, can be used in practical computations.

Remark 5 (Closure=Correction) Equation (14) shows that the closure model, τ, in (19) can be interpreted as a correction term that is added to the G-ROM (11) to correct its dynamics in X^R. So do we really need I-ROM in order to construct the closure model? In Sect. 5, we will show that the I-ROM is needed when we construct data-driven ROM closures. Furthermore, we note that the I-ROM derivation explains the closure model terminology. Indeed, $\tau^{I-ROM}(a_1, \ldots, a_r, a_{r+1}, \ldots, a_R)$ shows that the I-ROM (17) is closed in X^R, but not in X^r.

ROM closure models are of three types: (i) *Functional*, which use physical insight to construct the closure model; (ii) Structural, which use mathematical tools; and (iii) *Data-driven*, which use available data. The three types of ROM closure models are surveyed in [1]. In this tutorial, we take a different approach and, for clarity of presentation, focus on data-driven approaches, which have experienced a

tremendous development over the last few years. Specifically, in the next section, we present the data-driven variational multiscale ROM closure model.

5   The Data-Driven Variational Multiscale ROM (D2-VMS-ROM)

In this section, we illustrate how data-driven modeling can be leveraged to construct the ROM closure model. Specifically, we outline the main steps in the construction of one data-driven ROM closure model, i.e., the data-driven variational multiscale ROM (D2-VMS-ROM) that was proposed in [37] (see also [52]). To this end, we follow the presentation in Section 2.3 in [37] to construct the two-scale D2-VMS-ROM. (We note that a three-scale D2-VMS-ROM was also proposed and tested in [37].)

To build the D2-VMS-ROM, we start with the I-ROM (18). As explained in answer A5, to construct the ROM closure model we need to find an approximation $\tau(a_1, \ldots, a_r)$ for the I-ROM closure term in (18), $\tau^{I-ROM}(a_1, \ldots, a_r, a_{r+1}, \ldots, a_R)$. The construction of the data-driven ROM closure model consists of two steps: (i) postulating a model form ansatz; and (ii) solving a least squares problem to determine the coefficients of the model form. Next, we outline these two steps.

5.1   Model Form Ansatz

The first step in the construction of the data-driven ROM closure model is to *postulate a model form (ansatz)*. Specifically, we approximate the I-ROM closure term τ^{I-ROM} with $g(u_r)$, where g is a *generic* function whose coefficients/parameters still need to be determined:

$$\boxed{\tau_i^{I-ROM} \stackrel{(17)}{=} \left(f(u_R), \varphi_i\right) - \left(f(u_r), \varphi_i\right) \approx \left(g(u_r), \varphi_i\right), \quad i = 1, \ldots, r.} \quad (20)$$

5.2   Least Squares Problem

To determine the coefficients/parameters in g used in (20), in the offline stage, we solve the following low-dimensional *least squares problem*:

$$\min_{\boldsymbol{g} \text{ parameters}} \sum_{j=1}^{M} \left\| \left[\left(\boldsymbol{f}(\boldsymbol{u}_R^{FOM}(t_j)), \boldsymbol{\varphi}_i \right) - \left(\boldsymbol{f}(\boldsymbol{u}_r^{FOM}(t_j)), \boldsymbol{\varphi}_i \right) \right] \right.$$
$$\left. - \left(\boldsymbol{g}(\boldsymbol{u}_r^{FOM}(t_j)), \boldsymbol{\varphi}_i \right) \right\|^2, \tag{21}$$

where $\boldsymbol{u}_R^{FOM}$ and $\boldsymbol{u}_r^{FOM}$ are obtained from the FOM data, and M is the number of snapshots. Once $\boldsymbol{g}$ is determined, the I-ROM (17) with the I-ROM closure term replaced by $\boldsymbol{g}$ yields the *data-driven VMS-ROM (D2-VMS-ROM)*:

$$\left(\boldsymbol{u}_{r,t}, \boldsymbol{\varphi}_i \right) = \left(\boldsymbol{f}(\boldsymbol{u}_r), \boldsymbol{\varphi}_i \right) + \left(\boldsymbol{g}(\boldsymbol{u}_r), \boldsymbol{\varphi}_i \right), \qquad i = 1, \dots, r. \tag{22}$$

We emphasize that we have a lot of *flexibility* in choosing the model form ansatz (20) in the D2-VMS-ROM. For example, for the NSE, we can choose the following model form: $\forall i = 1, \dots, r,$

$$\left(\boldsymbol{g}(\boldsymbol{u}_r), \boldsymbol{\varphi}_i \right) = \left(\tilde{A} \boldsymbol{a} + \boldsymbol{a}^\top \tilde{B} \boldsymbol{a} \right)_i, \tag{23}$$

where, for computational efficiency, we assume that the structures of $\boldsymbol{g}$ and $\boldsymbol{f}$ are similar. Thus, in the least squares problem (21), we solve for all the entries in the $r \times r$ matrix $\tilde{A}$ and the $r \times r \times r$ tensor $\tilde{B}$.

The least squares problem (21) is *low-dimensional* since there are only $(r^2 + r^3)$ entries in $\tilde{A}$ and $\tilde{B}$ to be optimized, and r is small. Thus, (21) can be efficiently solved in the offline stage. For the NSE, the D2-VMS-ROM (22) takes the form

$$\dot{\boldsymbol{a}} = (A + \tilde{A})\boldsymbol{a} + \boldsymbol{a}^\top (B + \tilde{B})\boldsymbol{a}, \tag{24}$$

where A and B are the G-ROM operators in (11), and $\tilde{A}$ and $\tilde{B}$ are the VMS-ROM closure operators in (23).

Remark 6 (Physical Constraints) To improve the D2-VMS-ROM accuracy, one can use physical constraints when solving the least squares problem (21) to find the entries of the VMS-ROM closure operators $\tilde{A}$ and $\tilde{B}$. Numerical experiments have shown that imposing physical constraints can indeed increase the D2-VMS-ROM accuracy [35].

In Algorithm 2, we list the main steps in the construction of ROMs equipped with data-driven closure models.

6   ROM Closures in Action: Numerical Results

In the previous sections, we tried to convince the reader that ROM closures are important since they significantly increase the ROM accuracy in the under-

---

Algorithm 2 Data-driven ROM closure algorithm

1: Use numerical or experimental data to construct modes $\{\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_R\}$, which represent the recurrent spatial structures in the system.
2: Choose the dominant modes $\{\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_r\}$, $r \leq R$, as ROM basis functions.
3: Use a Galerkin expansion $\boldsymbol{u}_R(\boldsymbol{x}, t) = \sum_{j=1}^{R} a_j(t)\, \boldsymbol{\varphi}_j(\boldsymbol{x})$.
4: Replace $\boldsymbol{u}$ with $\boldsymbol{u}_R$ in (6).
5: Use a Galerkin projection of the PDE obtained in step 4 onto the space of resolved ROM scales $\mathbf{X}^r := \text{span}\{\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_r\}$ to obtain the *ideal ROM (I-ROM)*:

$$\dot{\boldsymbol{a}} = \boldsymbol{F}(\boldsymbol{a}) + \boldsymbol{\tau}^{I-ROM}, \tag{25}$$

where $\boldsymbol{a}(t) = (a_i(t))_{i=1,\ldots,r}$ is the vector of coefficients in the Galerkin expansion in step 3, $\boldsymbol{F}$ comprises the G-ROM operators, and $\boldsymbol{\tau}^{I-ROM}$ is the ideal ROM closure term defined in (17).

6: In the offline stage:

- Compute the G-ROM operators (e.g., vectors, matrices, and tensors), which are preassembled from the ROM basis.
- Choose a model form $\boldsymbol{g}$ for $\boldsymbol{\tau}^{I-ROM}$ in (25).
- Solve the least squares problem (21) to find the parameters in the model form.
- Compute $\boldsymbol{G}(\boldsymbol{a})$, which comprises the ROM closure operators corresponding to the model form $\boldsymbol{g}$ for $\boldsymbol{\tau}^{I-ROM}$.
- Replace the I-ROM (25) with the data-driven ROM closure model

$$\dot{\boldsymbol{a}} = \boldsymbol{F}(\boldsymbol{a}) + \boldsymbol{G}(\boldsymbol{a}). \tag{26}$$

7: In the online stage, repeatedly use the data-driven ROM closure (26) for various parameter settings and/or longer time intervals.

---

resolved regime. We note, however, that all our arguments have been *mathematical* arguments. Thus, we can ask the following natural question:

? Q6

Do ROM closures work in practice?

The answer to Q6 is simple:

> A6

Yes!

Table 1 2D flow past a circular cylinder. L^2 norm of errors for G-ROM, D2-VMS-ROM, and I-ROM for different r values

r	G-ROM	I-ROM	D2-VMS-ROM
2	1.509e+00	5.987e−02	1.504e−02
3	8.595e−01	5.072e−01	8.024e−02
4	6.583e−01	3.415e−02	2.538e−02
5	7.095e−01	4.197e−01	5.156e−01
6	5.562e−01	2.371e−01	3.132e−02
7	4.760e−01	2.324e−01	6.482e−02
8	2.692e−01	2.122e−01	1.691e−02

The answer A6 is elaborated in the survey in [1], which presents a plethora of examples of under-resolved ROM simulations of complex dynamics (e.g., turbulent flows) in which ROM closures significantly increase the accuracy at a modest computational overhead.

In this section, for clarity of presentation, we illustrate how a specific ROM closure model (i.e., the D2-VMS-ROM outlined in Sect. 5) increases the ROM accuracy for the 2D flow past a circular cylinder [37], which is a simple test problem commonly used in the ROM community. (We note, however, that the D2-VMS-ROM was successfully used for challenging test problems, e.g., turbulent channel flow [36] and the quasi-geostrophic equations [38].) In our numerical investigation, we use a Reynolds number $Re = 1000$ and four ROM basis functions (i.e., $r = 4$). Details of the computational setting can be found in [37].

In Table 1, we list the L^2 norm of the error, $\boldsymbol{u}^{FOM} - \boldsymbol{u}^{ROM}$, for G-ROM (11) (second column), I-ROM (17) (third column), and D2-VMS-ROM (22) (fourth column). We note that the G-ROM error is relatively large, whereas both the D2-VMS-ROM and I-ROM error are much smaller than G-ROM. In particular, the D2-VMS-ROM error is one and even two orders of magnitude smaller than the G-ROM error for some r values. In Fig. 5, we present plots of the Euclidian norm of the error at each mesh point at the final time, for G-ROM (11) (top), I-ROM (17) (middle), and D2-VMS-ROM (22) (bottom). We note that the G-ROM error is relatively large, whereas the D2-VMS-ROM error is almost negligible. These two plots clearly show that adding the data-driven closure model to the classical G-ROM (i.e., using the D2-VMS-ROM) significantly increases the G-ROM accuracy. Although the I-ROM cannot be used in practical computations (since it is not closed), we included I-ROM results for comparison purposes. Table 1 and Fig. 5 show that the D2-VMS-ROM is not only more accurate than the standard G-ROM, but it is almost as accurate as the I-ROM (which includes an ideal closure model). Thus, for this test problem, the D2-VMS-ROM error almost reaches the theoretical lower bound given by the I-ROM error. Overall, Fig. 5 clearly shows that closure models can significantly increase the ROM accuracy in under-resolved simulations.
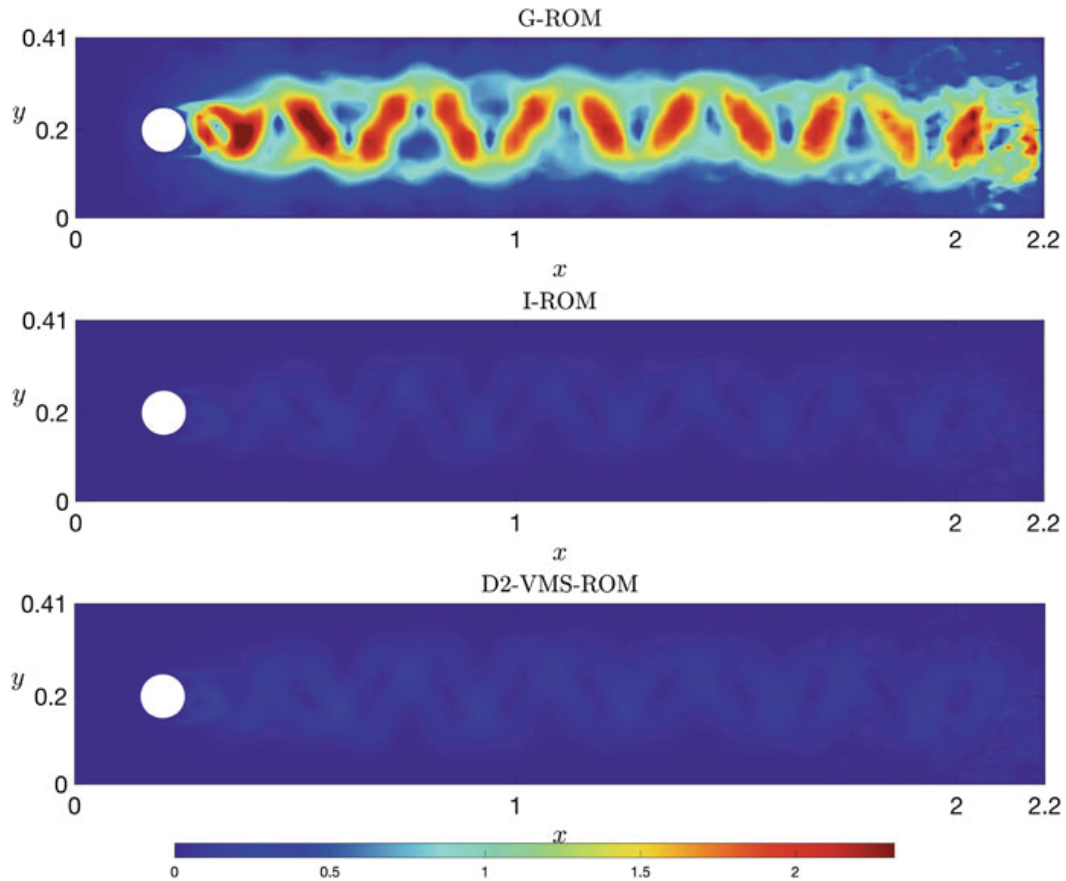
Fig. 5 2D flow past a circular cylinder. The Euclidian norm of the error, $\boldsymbol{u}^{FOM} - \boldsymbol{u}^{ROM}$, at mesh points for G-ROM (11) (top), I-ROM (17) (middle), and D2-VMS-ROM (22) (bottom). The D2-VMS-ROM error is significantly lower than the G-ROM error, which illustrates the benefit of ROM closure modeling. Also note that, in this case, the D2-VMS-ROM error almost reaches the theoretical lower bound given by the I-ROM error

7   Mathematical Foundations of ROM Closures

In Sects. 4 and 5, we discussed the *mathematical modeling* of ROM closures. In Sect. 6, we discussed the *numerical simulation* of ROM closures. The following is a natural question:

⍰ Q7

What can we prove about ROM closures?

The answer to Q7 is simple:

> **A7**

Not so much. Yet.

In this section, we briefly summarize some relevant theoretical aspects associated with ROM closure modeling. Compared with the analysis of classical numerical schemes [6, 27, 44], the theoretical foundations for ROM closures are much less developed. We emphasize, however, that recently there have been significant advancements in this exciting and important research area.

The theoretical investigations of ROM closure modeling generally aim at proving error bounds for ROM closures of the form

$$\|u^{FOM} - u^{ROM}\| \leq C \, (\text{space error} + \text{time error} + \text{ROM error}) , \qquad (27)$$

where u^{FOM} is the FOM solution, u^{ROM} is the ROM solution, $\|\cdot\|$ is a given norm, the space error is the error that results from the spatial approximation, the time error is the error that results from the time approximation, the ROM error is the error that results from the ROM approximation, and C is a generic constant that does not depend on the discretization parameters. We note that the first two terms on the right-hand side of (27) appear in error bounds for classical numerical discretizations, e.g., the FEM [27]. The third term, however, does not appear in these bounds.

The main purpose of the error bound (27) is to show the *convergence* of the ROM solution to the FOM solution. For example, as the spatial mesh size and the time step go to zero, the space error and time error in (27), respectively, are expected to go to zero (at a rate that depends on the particular spatial and time discretizations used). Furthermore, as the number of ROM basis functions goes to the rank of the snapshot matrix, the ROM error in (27) is also expected to go to zero. Thus, as the right-hand side of (27) goes to zero, so does the error on the left-hand side of (27), which proves the convergence of the ROM solution to the FOM solution.

For the G-ROM (11), the numerical analysis started two decades ago with the pioneering work of Kunisch and Volkwein, who proved the first error bounds for the POD of parabolic equations, e.g., the heat equation [31] and the Navier-Stokes equations [32]. More than a decade later, Singler improved Kunisch and Volkwein's results, by proving sharper error bounds [47]. Recently, optimal pointwise in time error bounds were proved in [30]. These results finally bring the G-ROM numerical analysis to a level comparable to (although not as developed as) the level of the numerical analysis of the FEM.

For the ROM closure models, the numerical analysis is relatively scarce. The numerical analysis for ROM closures aims at proving a modified form of the G-ROM error bound (27):

$$\|\boldsymbol{u}^{FOM} - \boldsymbol{u}^{ROM}\| \leq C \text{ (space error + time error + ROM error + closure error) ,}$$
(28)

where the closure error is the error that results from the approximation of the closure term $\boldsymbol{\tau}^{I-ROM}$ in the I-ROM (17) with a closure model.

As mentioned in [1], the first numerical analysis of ROM closures was performed in [7], where error bounds for the time discretization of the Smagorinsky model (i.e., a ROM closure model developed on phenomenological arguments) were proven. Error bounds for the time and space discretizations of the Smagorinsky model were later proven in [43] in an RBM context. Error bounds for VMS closure models were proved in [19, 25, 26, 45] (see also [4, 46] for related work). Finally, error bounds for the D2-VMS-ROM (22) were proved in [29] (see also [28] for related work).

8   Conclusions and Outlook

In this paper, we presented a brief tutorial for reduced order model (ROM) closures. In the first part of our tutorial, we motivated the ROM closures. We note that ROM closure modeling is often misunderstood in the ROM community. Thus, we started our tutorial by explaining the need for ROM closure modeling (i.e., the ROM closure problem) in realistic applications, and then we carefully described the ROM closure model. Specifically, we first outlined the main steps used to construct the Galerkin ROM (G-ROM), which is based on leveraging a data-driven basis in the classical Galerkin framework. Next, we noted that, although G-ROM can decrease the computational cost of standard numerical discretizations by orders of magnitude, it yields inaccurate results in under-resolved ROM simulations, i.e., when the number of basis functions is not enough to capture the underlying system's dynamics. To address the G-ROM's inaccuracy in under-resolved simulations, we introduced the ROM closure model. We motivated the need for ROM closure by presenting a mathematical extension of the classical Galerkin framework to include not only the space of resolved scales, but also the space of unresolved scales. In this extended variational multiscale framework, we showed that the correct ROM dynamics include an additional term (i.e., the closure term), which represents the effect of the unresolved scales. Furthermore, we showed that this mathematical framework, which we named the ideal ROM (I-ROM), yields numerical results that are significantly more accurate than the G-ROM results. Thus, we concluded that a ROM closure model, which is a practical model for the I-ROM closure term, should be added to the G-ROM to increase its accuracy in realistic, under-resolved simulations.

In the second part of our tutorial, we outlined the main steps in the construction of ROM closure models. To simplify our presentation, we focused on one particular type of ROM closure modeling, i.e., data-driven modeling. Furthermore, we illustrated this construction for one specific data-driven ROM closure model, i.e., the data-driven variational multiscale ROM (D2-VMS-ROM). In our construction,

we started with the closure term in the I-ROM, and we simply posed the closure problem as leveraging the available FOM data to find the "best" ROM closure model. To this end, we first postulated a model form for the ROM closure model. Then, we solved a least squares problem to find the parameters in the model form that yield the ROM closure model that is the closest to the ideal ROM closure model. Finally, we also included numerical results for the two-dimensional flow past a circular cylinder, which showed that the D2-VMS-ROM was significantly more accurate than the standard G-ROM, and almost as accurate as the I-ROM. These numerical results illustrated the significant benefit of ROM closure modeling in under-resolved simulations.

We hope that this brief tutorial offers a glimpse into the exciting research field of ROM closure modeling, which has witnessed a significant development over the past two decades. This research area is currently experiencing a dynamic development in several directions. One of the most active research directions is the use of machine learning tools to construct more accurate and more efficient ROM closure models. Recently, deep learning models have been shown to be quite effective and computationally efficient in capturing the relationship between resolved and unresolved scales [2]. However, these models often need large amounts of training data and their generalization, expressivity, and analysis still remain mostly challenging.

Another important research direction is the development of ROM closures for problems in solid mechanics. Although most ROM closure modeling has been performed in computational fluid dynamics [1], there has been recent work done in solid mechanics. For example, approximations of the mechanical behavior of soft tissue showed substantial improvement in accuracy over G-ROM with the addition of ROM closure terms at a modest computational overhead [48]. The ability of ROM closure to capture the nonlinearities of soft tissue behavior is especially promising for its application in biomechanics.

Depending on the applications, one can also couple ROMs with additional parameterization schemes or surrogate models for some of the unresolved scales in order to recover more dynamical features of the original system, especially when the ROMs are constructed for under-resolved dynamical regimes. For instance, in the context of data assimilation, when observations are only available for the (large-scale) low-frequency modes, one can design computationally efficient strategies within the conditional Gaussian framework [12–14] to approximate the dynamics of the high-frequency (unresolved) modes with quantified uncertainties by a suitable dynamical model for the unresolved modes.

Finally, providing mathematical support for ROM closures is also an important research direction. We note that significant mathematical support has been provided for closures in classical computational fluid dynamics [6, 27, 44]. For ROM closures, however, only the first steps have been taken and much more remains to be done.

References

1. S.E. Ahmed, S. Pawar, O. San, A. Rasheed, T. Iliescu, B.R. Noack, On closures for reduced order models—a spectrum of first-principle to machine-learned avenues. Phys. Fluids **33**(9), 091301 (2021)
2. S.E. Ahmed, O. San, A. Rasheed, T. Iliescu, A long short-term memory embedding for hybrid uplifted reduced order models. Phys. D 132471 (2020)
3. S.E. Ahmed, O. San, A. Rasheed, T. Iliescu, A. Veneziani, Physics guided machine learning for variational multiscale reduced order modeling (2022). in preparation
4. M. Azaïez, T.C. Rebollo, S. Rubino, A cure for instabilities due to advection-dominance in POD solution to advection-diffusion-reaction equations. J. Comput. Phys. **425**, 109916 (2021)
5. J. Berner, U. Achatz, L. Batté, L. Bengtsson, A. de la Cámara, H.M. Christensen, M. Colangeli, D.R.B. Coleman, D. Crommelin, S.I. Dolaptchiev, C.L.E. Franzke, P. Friederichs, P. Imkeller, H. Järvinen, S. Juricke, V. Kitsios, F. Lott, V. Lucarini, S. Mahajan, . . . , and J.-I. Yano, Stochastic parameterization toward a new view of weather and climate models. Bull. Am. Meteorol. Soc. **98**(3), 565–588 (2017)
6. L.C. Berselli, T. Iliescu, W.J. Layton, *Mathematics of Large Eddy Simulation of Turbulent Flows*. Scientific Computation. (Springer, Berlin, 2006)
7. J. Borggaard, T. Iliescu, Z. Wang, Artificial viscosity proper orthogonal decomposition. Math. Comput. Model. **53**(1–2), 269–279 (2011)
8. S.L. Brunton, J.N. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control* (Cambridge University Press, Cambridge, 2019)
9. J. Burkardt, M. Gunzburger, H.C. Lee, POD and CVT-based reduced-order modeling of Navier–Stokes flows. Comput. Methods Appl. Mech. Eng. **196**(1–3), 337–355 (2006)
10. M.D. Chekroun, H. Liu, J.C. McWilliams, Variational approach to closure of nonlinear dynamical systems: autonomous case. J. Stat. Phys. **179**, 1073–1160 (2020)
11. M.D. Chekroun, H. Liu, S. Wang, *Stochastic parameterizing manifolds and non-Markovian reduced equations: stochastic manifolds for nonlinear SPDEs II*. Springer Briefs in Mathematics (Springer, Berlin, 2015)
12. N. Chen, Learning nonlinear turbulent dynamics from partial observations via analytically solvable conditional statistics. J. Comput. Phys. **418**, 109635 (2020)
13. N. Chen, Y. Li, H. Liu, Conditional Gaussian nonlinear system: a fast preconditioner and a cheap surrogate model for complex nonlinear systems (2021). arXiv preprint arXiv:2112.05226
14. N. Chen, A.J. Majda, Conditional Gaussian systems for multiscale nonlinear stochastic systems: Prediction, state estimation and uncertainty quantification. Entropy **20**(7), 509 (2018)
15. F. Chinesta, P. Ladeveze, E. Cueto, A short review on model order reduction based on proper generalized decomposition. Arch. Comput. Methods Eng. **18**(4), 395–404 (2011)
16. A.J. Chorin, F. Lu, Discrete approach to stochastic parametrization and dimension reduction in nonlinear dynamics. Proc. Natl. Acad. Sci. U.S.A. **112**(32), 9804–9809 (2015)
17. D. Crommelin, E. Vanden-Eijnden, Subgrid-scale parameterization with conditional Markov chains. J. Atmos. Sci. **65**(8), 2661–2675 (2008)
18. V. DeCaria, T. Iliescu, W. Layton, M. McLaughlin, M. Schneier, An artificial compression reduced order model. SIAM J. Numer. Anal. **58**(1), 565–589 (2020)
19. F.G. Eroglu, S. Kaya, L.G. Rebholz, A modular regularized variational multiscale proper orthogonal decomposition for incompressible flows. Comput. Methods Appl. Mech. Eng. **325**, 350–368 (2017)
20. C. Foiaş, O. Manley, R. Rosa, R. Temam, *Navier–Stokes Equations and Turbulence* (Cambridge University Press, Cambridge, 2001)
21. J.-L. Guermond, Stabilization of Galerkin approximations of transport equations by subgrid modeling. M2AN Math. Model. Numer. Anal. **33**(6), 1293–1316 (1999)
22. J.S. Hesthaven, G. Rozza, B. Stamm, *Certified Reduced Basis Methods for Parametrized Partial Differential Equations* (Springer, Berlin, 2015)

23. P. Holmes, J.L. Lumley, G. Berkooz, *Turbulence, Coherent Structures, Dynamical Systems and Symmetry* (Cambridge, 1996)
24. T.J.R. Hughes, Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods. Comput. Methods Appl. Mech. Eng. **127**(1–4), 387–401 (1995)
25. T. Iliescu, Z. Wang, Variational multiscale proper orthogonal decomposition: convection-dominated convection-diffusion-reaction equations. Math. Comput. **82**(283), 1357–1378 (2013)
26. T. Iliescu, Z. Wang, Variational multiscale proper orthogonal decomposition: Navier-Stokes equations. Num. Methods P.D.E.s **30**(2), 641–663 (2014)
27. V. John, *Finite Element Methods for Incompressible Low Problems* (Springer, Berlin, 2016)
28. B. Koc, M. Mohebujjaman, C. Mou, T. Iliescu, Commutation error in reduced order modeling of fluid flows. Adv. Comput. Math. **45**(5–6), 2587–2621 (2019)
29. B. Koc, C. Mou, H. Liu, Z. Wang, G. Rozza, T. Iliescu, Verifiability of the data-driven variational multiscale reduced order model (2021). http://arxiv.org/abs/2108.04982
30. B. Koc, S. Rubino, M. Schneier, J.R. Singler, T. Iliescu, On optimal pointwise in time error bounds and difference quotients for the proper orthogonal decomposition. SIAM J. Numer. Anal. **59**(4), 2163–2196 (2021)
31. K. Kunisch, S. Volkwein, Galerkin proper orthogonal decomposition methods for parabolic problems. Numer. Math. **90**(1), 117–148 (2001)
32. K. Kunisch, S. Volkwein, Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. SIAM J. Numer. Anal. **40**(2), 492–515 (electronic) (2002)
33. W.J. Layton, A connection between subgrid scale eddy viscosity and mixed methods. Appl. Math. Comput. **133**, 147–157 (2002)
34. A.J. Majda, I. Timofeyev, E. Vanden-Eijnden, A mathematical framework for stochastic climate models. Commun. Pure Appl. Math. **54**, 891–974 (2001)
35. M. Mohebujjaman, L.G. Rebholz, T. Iliescu, Physically-constrained data-driven correction for reduced order modeling of fluid flows. Int. J. Num. Meth. Fluids **89**(3), 103–122 (2019)
36. C. Mou, *Data-Driven Variational Multiscale Reduced Order Modeling of Turbulent Flows*. Ph.D. Thesis, Virginia Tech, 2021
37. C. Mou, B. Koc, O. San, L.G. Rebholz, T. Iliescu, Data-driven variational multiscale reduced order models. Comput. Methods Appl. Mech. Eng. **373**, 113470 (2021)
38. C. Mou, H. Liu, D.R. Wells, T. Iliescu, Data-driven correction reduced order models for the quasi-geostrophic equations: a numerical investigation. Int. J. Comput. Fluid Dyn. **34**, 147–159 (2020)
39. B.R. Noack, M. Morzynski, G. Tadmor, *Reduced-Order Modelling for Flow Control*, vol. 528 (Springer, Berlin, 2011)
40. S. Pawar, S.E. Ahmed, O. San, A. Rasheed, Data-driven recovery of hidden physics in reduced order modeling of fluid flows. Phys. Fluids **32**(3), 036602 (2020)
41. S. Pawar, S.E. Ahmed, O. San, A. Rasheed, An evolve-then-correct reduced order model for hidden fluid dynamics. Mathematics **8**(4), 570 (2020)
42. A. Quarteroni, A. Manzoni, F. Negri, *Reduced Basis Methods for Partial Differential Equations: An Introduction*, vol. 92 (Springer, Berlin, 2015)
43. T.C. Rebollo, E.D. Ávila, M.G. Mármol, F. Ballarin, G. Rozza, On a certified Smagorinsky reduced basis turbulence model. SIAM J. Numer. Anal. **55**(6), 3047–3067 (2017)
44. T. Chacón Rebollo, R. Lewandowski, *Mathematical and Numerical Foundations of Turbulence Models and Applications* (Springer, Berlin, 2014)
45. J.P. Roop, A proper-orthogonal decomposition variational multiscale approximation method for a generalized Oseen problem. Adv. Numer. Anal. (2013)
46. S. Rubino, Numerical analysis of a projection-based stabilized POD-ROM for incompressible flows. SIAM J. Numer. Anal. **58**(4), 2019–2058 (2020)
47. J.R. Singler, New POD error expressions, error bounds, and asymptotic results for reduced order models of parabolic PDEs. SIAM J. Numer. Anal. **52**(2), 852–876 (2014)

48. W. Snyder, J.A. McGuire, C. Mou, D. A. Dillard, T. Iliescu, R. De Vita, Data-driven variational multiscale reduced order modeling of vaginal tissue (2022). in preparation
49. K. Taira, M.S. Hemati, S.L. Brunton, Y. Sun, K. Duraisamy, S. Bagheri, S.T. M. Dawson, C.-A. Yeh, Modal analysis of fluid flows: applications and outlook. AIAA J. **58**(3), 998–1022 (2020)
50. R. Temam, *Navier-Stokes equations: Theory and Numerical Analysis*, vol. 2 (American Mathematical Society, Providence, 2001)
51. S. Volkwein, Proper orthogonal decomposition: theory and reduced-order modelling. Lecture Notes, University of Konstanz (2013). http://www.math.uni-konstanz.de/numerik/personen/volkwein/teaching/POD-Book.pdf
52. X. Xie, M. Mohebujjaman, L.G. Rebholz, T. Iliescu, Data-driven filtered reduced order modeling of fluid flows. SIAM J. Sci. Comput. **40**(3), B834–B857 (2018)
53. L. Zanna, P. Porta Mana, J. Anstey, T. David, T. Bolton, Scale-aware deterministic and stochastic parametrizations of eddy-mean flow interaction. Ocean Model. **111**, 66–80 (2017)