Finite Elements in Analysis and Design 226 (2023) 104021

Contents lists available at ScienceDirect

Finite Elements in Analysis & Design

journal homepage: www.elsevier.com/locate/finel

Approximate deconvolution Leray reduced order model for convection-dominated flows

Anna Sanfilippo [a], Ian Moore [b], Francesco Ballarin [c,*], Traian Iliescu [b]

[a] Department of Mathematics, Università di Trento, via Sommarive 14, Povo, 38123, Italy
[b] Department of Mathematics, Virginia Tech, 225 Stanger Street, Blacksburg, 24061, VA, United States of America
[c] Department of Mathematics and Physics, Università Cattolica del Sacro Cuore, via Garzetta 48, Brescia, 25133, Italy

ARTICLE INFO

Keywords:
Reduced order models
Approximate deconvolution
Under-resolved regime
Spatial filter
Regularization
Leray model

ABSTRACT

In this paper, we propose a novel ROM stabilization strategy for under-resolved convection-dominated flows, the approximate deconvolution Leray ROM (ADL-ROM). The new ADL-ROM introduces AD as a new means to increase the accuracy of the classical Leray ROM (L-ROM) without degrading its numerical stability. We also introduce two new AD ROM strategies: the Tikhonov and van Cittert methods. Our numerical investigation for convection-dominated systems shows that, when the filter radius is relatively large, the new ADL-ROM is more accurate than the standard L-ROM. Furthermore, the new ADL-ROM is less sensitive with respect to model parameters than L-ROM.

1. Introduction

Galerkin ROMs (G-ROMs) are computational models that leverage data to dramatically reduce the dimension of full order models (FOMs), i.e., models obtained from classical numerical discretizations (e.g., the finite element method (FEM)). G-ROMs have been used to reduce the FOM computational cost in the numerical simulation of laminar fluid flows described by the Navier–Stokes equations (NSE) [1–5]. However, in the under-resolved regime (i.e., when the number of ROM degrees of freedom is not enough to accurately represent the flow dynamics), G-ROM yields inaccurate results, usually in the form of numerical oscillations. To alleviate this numerical inaccuracy, several types of ROM stabilization techniques have been developed [6–20].

One popular type of ROM stabilization for fluid flows is *regularized ROMs (Reg-ROMs)* [21–30]. The principle used to construct Reg-ROMs is to use a *ROM filter* to smooth out various terms in the underlying NSE. There are two ROM filters in current use: (i) The ROM projection is a ROM filter in which filtering takes place exclusively in the ROM space. (ii) The ROM differential filter is a ROM filter in which filtering takes place exclusively in the physical space. Higher-order DFs have been introduced in [31]. Both the ROM projection and the ROM differential filter have been used to construct Reg-ROMs.

One of the most popular Reg-ROMs is the *Leray ROM (L-ROM)* [21,22,25,27,29], which is a ROM stabilization for under-resolved convection-dominated flows. The L-ROM, inspired by the work of Jean Leray in the mathematical theory of the NSE [32], is based on a simple, yet powerful idea: Replace the nonlinear term in the NSE, $\boldsymbol{u} \cdot \nabla \boldsymbol{u}$, with $\overline{\boldsymbol{u}} \cdot \nabla \boldsymbol{u}$, where $\overline{\boldsymbol{u}}$ denotes the ROM filtered velocity. Because the filtered velocity, $\overline{\boldsymbol{u}}$, is smoother than the original, unfiltered velocity, $\boldsymbol{u}$, Leray was able to prove the existence of weak solutions to the NSE [32]. Filtering the velocity is also a good idea from the computational point of view: The Leray model was used in under-resolved simulations of turbulent flows with classical numerical discretizations [33]. The Leray model was extended to reduced order modeling of the NSE in [29] (see [27] for earlier work on the Kuramoto–Sivashinsky equations). In [29], it was

[*] Corresponding author.
E-mail address: francesco.ballarin@unicatt.it (F. Ballarin).

https://doi.org/10.1016/j.finel.2023.104021
Received 20 July 2023; Received in revised form 8 August 2023; Accepted 14 August 2023
Available online 28 August 2023

shown that the Leray ROM yields significantly more accurate results than the standard Galerkin ROM in the under-resolved numerical simulation of the 3D flow past a circular cylinder at Reynolds number $Re = 1000$. Since then, the L-ROM has been successfully used in the under-resolved simulation of various convection-dominated flows, e.g., further NSE applications [21,22,34,35], the stochastic NSE [31,36], and the quasi-geostrophic equations [37,38].

Despite the significant improvement over the standard Galerkin ROM, one drawback of the L-ROM observed in these numerical investigations is that it can be overdiffusive. For example, when the ROM filter radius, δ, is too large, the filter introduces too much dissipation in the L-ROM. Another drawback of the L-ROM observed in these numerical investigations is its sensitivity with respect to the ROM filter radius. Specifically, small perturbations of the optimal filter radius can decrease the L-ROM accuracy.

To address these L-ROM drawbacks, we propose a new Reg-ROM, the *approximate deconvolution Leray ROM (ADL-ROM)*, which is a significant improvement of the L-ROM. The innovation in the new ADL-ROM is the leverage of approximate deconvolution, a classical strategy used in the image processing and inverse problems communities, to increase the accuracy and decrease the sensitivity of the L-ROM. Specifically, in the new ADL-ROM, the filtered L-ROM velocity, $\overline{\boldsymbol{u}}$, is replaced with the AD velocity, $D(\overline{\boldsymbol{u}})$. Thus, the ADL-ROM nonlinear term is $D(\overline{\boldsymbol{u}}) \cdot \nabla \boldsymbol{u}$. Since $D(\overline{\boldsymbol{u}})$ approximation of $\boldsymbol{u}$ is both stable and accurate, using the AD operator in the ADL-ROM has the role of increasing the accuracy of the nonlinear term while preserving the extra stability introduced by the ROM filtering in $\overline{\boldsymbol{u}}$. Thus, the ADL-ROM can be thought of as a compromise between the G-ROM and the overdissipative L-ROM. To our knowledge, this is the first time the AD concept is used to construct ROM stabilizations, such as Reg-ROMs.

The rest of the paper is organized as follows: In Section 2, we outline the construction of the standard G-ROM for the NSE. In Section 3, we first present the differential filter and then introduce several AD strategies: the van Cittert, Tikhonov, and Lavrentiev approaches. In Section 4, we first present the classical L-ROM and then introduce the novel ADL-ROM. In Section 5, we perform a numerical investigation of the new ADL-ROM. Specifically, we compare the ADL-ROM with the classical L-ROM and the standard G-ROM in two numerical tests: the Burgers equation with a small diffusion coefficient and the convection-dominated flow past a backward-facing step. Finally, in Section 6, we present our conclusions and outline future research directions.

2. G-ROM

In this section, we briefly describe the construction of the standard G-ROM. As a mathematical model, we consider the incompressible NSE:

$$\frac{\partial \boldsymbol{u}}{\partial t} - Re^{-1}\Delta \boldsymbol{u} + \big(\boldsymbol{u}\cdot\nabla\big)\boldsymbol{u} + \nabla p \;=\; \boldsymbol{f}, \text{ in } \Omega \times (0,T] \tag{1}$$

$$\nabla \cdot \boldsymbol{u} \;=\; 0, \text{ in } \Omega \times (0,T], \tag{2}$$

where Ω is the spatial domain, T is the final time, $\boldsymbol{u} = [u_1, u_2, u_3]^\top : \Omega\times[0,T] \to \mathbb{R}^3$ is the velocity vector field, $p : \Omega\times[0,T] \to \mathbb{R}$ the pressure field, Re the Reynolds number, and $\boldsymbol{f} : \Omega\times[0,T] \to \mathbb{R}^3$ the forcing term. The NSE are equipped with appropriate boundary and initial conditions.

To build the G-ROM, we first collect the snapshots $\{\boldsymbol{u}_h^0, \boldsymbol{u}_h^1, \dots, \boldsymbol{u}_h^M\}$, which are FOM solutions at the time instances $t_0 \equiv 0, t_1, \dots, t_M \equiv T$, which are assumed to be equispaced for simplicity of presentation. Further details about the FOM are postponed to Section 5, as they are not essential for the presentation of the G-ROM. Next, we compute the centered snapshots $\{\boldsymbol{u}_h^0 - \boldsymbol{U}_h, \boldsymbol{u}_h^1 - \boldsymbol{U}_h, \dots, \boldsymbol{u}_h^M - \boldsymbol{U}_h\}$, where the centering trajectory $\boldsymbol{U}_h$ of the flow is defined as $\boldsymbol{U}_h(\boldsymbol{x}) = \frac{1}{M+1}\sum_{k=0}^{M} \boldsymbol{u}_h^k(\boldsymbol{x})$. Finally, we use the proper orthogonal decomposition (POD) [39,40] on the centered snapshots to construct the ROM basis functions $\boldsymbol{\varphi}_1, \dots, \boldsymbol{\varphi}_r$, where r is the ROM dimension. Although in our numerical experiments we use the POD to construct the ROM basis, we note that other ROM bases could be used instead [2,3,5,41]. In what follows, we assume that the ROM velocity approximation can be written as

$$\boldsymbol{u}_r(\boldsymbol{x},t) \;=\; \boldsymbol{U}_h(\boldsymbol{x}) + \sum_{j=1}^{r} c_j(t)\boldsymbol{\varphi}_j(\boldsymbol{x})\,, \tag{3}$$

where $\boldsymbol{c}(t) = [c_1(t), \dots, c_r(t)]^\top$ are the sought ROM coefficients. The next step in the G-ROM construction is to replace $\boldsymbol{u}$ with $\boldsymbol{u}_r$ in (1) and project the resulting equations onto the space spanned by the ROM basis, $\{\boldsymbol{\varphi}_j\}_{j=1}^r$. This yields the G-ROM:

$$\left(\frac{\partial \boldsymbol{u}_r}{\partial t}, \boldsymbol{\varphi}_i\right) + \big((\boldsymbol{u}_r\cdot\nabla)\boldsymbol{u}_r, \boldsymbol{\varphi}_i\big) + Re^{-1}\big(\nabla \boldsymbol{u}_r, \nabla \boldsymbol{\varphi}_i\big) = \big(\boldsymbol{f}, \boldsymbol{\varphi}_i\big)\,, \quad i = 1, \dots, r\,, \tag{4}$$

where $(\cdot,\cdot)$ denotes the L^2 inner product. We notice that the pressure does not appear anymore in (4) since the ROM basis functions $\boldsymbol{\varphi}_1, \dots, \boldsymbol{\varphi}_r$ are divergence free. Alternative ROMs which also allow to recover the pressure are discussed, e.g., in [1,4,42–44]. Since the Reg-ROMs introduced in Section 4 only differ in the treatment of the nonlinear term (which does not involve the pressure), dropping the pressure unknown from the ROM is beneficial in view of a simpler exposition of the novel Reg-ROMs. The G-ROM can be written as the following system of differential equations for the vector of time coefficients, $\boldsymbol{c} = \boldsymbol{c}(t)$:

$$\boldsymbol{c}' = \boldsymbol{b} + \boldsymbol{A}\boldsymbol{c} + \boldsymbol{c}^\top \boldsymbol{B}\boldsymbol{c}\,, \tag{5}$$

where the vector $\boldsymbol{b}$, the matrix $\boldsymbol{A}$, and the tensor $\boldsymbol{B}$ are defined as follows:

$$\boldsymbol{b}_i \;=\; \big(\boldsymbol{\varphi}_i, \boldsymbol{f}\big) - \big(\boldsymbol{\varphi}_i, \boldsymbol{U}_h\cdot\nabla \boldsymbol{U}_h\big) - Re^{-1}\big(\nabla\boldsymbol{\varphi}_i, \nabla \boldsymbol{U}_h\big)\,, \tag{6}$$

$$\boldsymbol{A}_{im} \;=\; -\big(\boldsymbol{\varphi}_i, \boldsymbol{U}_h\cdot\nabla\boldsymbol{\varphi}_m\big) - \big(\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_m\cdot\nabla \boldsymbol{U}_h\big) - Re^{-1}\big(\nabla\boldsymbol{\varphi}_i, \nabla\boldsymbol{\varphi}_m\big)\,, \tag{7}$$

$$\boldsymbol{B}_{imn} = -\left(\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_m \cdot \nabla \boldsymbol{\varphi}_n\right), \tag{8}$$

for $i, n, m = 1, \ldots, r$.

As illustrated in the numerical investigation in Section 5, although G-ROM (4) works well for laminar flows, for under-resolved convection-dominated flows it yields numerical oscillations.

3. Approximate deconvolution (AD) ROM

In this section, we introduce the AD strategy, which is a new concept in ROM stabilization. AD is central in image processing and inverse problems [45,46]. In a nutshell, the AD goal can be formulated as follows: Given an approximation of the filtered input signal, $\overline{\boldsymbol{u}} \doteq G\boldsymbol{u}$, where G is an invertible spatial filter, find an approximation of the unfiltered input signal, $\boldsymbol{u}$. Of course, the exact deconvolution, $G^{-1}\overline{\boldsymbol{u}}$, would seem a natural choice. Computationally, however, the exact deconvolution is a very bad idea since the noise in the high wavenumber components of $\overline{\boldsymbol{u}}$ is amplified by the inverse filter, G^{-1}. Thus, in practice, AD strategies are used instead [45,46]. In large eddy simulations of turbulent flows, the AD models have been pioneered by Adams and Stolz for classical numerical discretizations [47]. In reduced order modeling, the only AD model was proposed in [48], where AD was used to develop a ROM closure model.

In this paper, we leverage the AD strategy to develop a different type of ROM: the ADL-ROM (Section 4.2), which is a ROM stabilization. To construct the new ADL-ROM, we first present the ROM differential filter (Section 3.1). Then, we present three different AD strategies: the van Cittert AD approach (Section 3.2), the Tikhonov AD approach (Section 3.3), and the Lavrentiev AD approach (Section 3.4). We note that the Lavrentiev AD approach was used in [48] to develop a ROM closure model. To our knowledge, however, the van Cittert and Tikhonov approaches have never been used to develop ROM AD operators (see [49–51] for Tikhonov methods for ROM regularizations). We also note that, in Section 5.2, we perform a preliminary numerical investigation of the three AD strategies.

In what follows, we will use the following expansions:

$$\boldsymbol{u}_r(\boldsymbol{x}, t) = \boldsymbol{U}_h(\boldsymbol{x}) + \sum_{i=1}^{r} c_i(t)\boldsymbol{\varphi}_i(\boldsymbol{x}), \qquad \boldsymbol{c} \doteq [c_1, \ldots, c_r]^\top, \tag{9}$$

$$\overline{\boldsymbol{u}}_r(\boldsymbol{x}, t) = \boldsymbol{U}_h(\boldsymbol{x}) + \sum_{i=1}^{r} \overline{c}_i(t)\boldsymbol{\varphi}_i(\boldsymbol{x}), \qquad \overline{\boldsymbol{c}} \doteq [\overline{c}_1, \ldots, \overline{c}_r]^\top, \tag{10}$$

$$\boldsymbol{u}_r^{AD}(\boldsymbol{x}, t) = \boldsymbol{U}_h(\boldsymbol{x}) + \sum_{i=1}^{r} c_i^{AD}(t)\boldsymbol{\varphi}_i(\boldsymbol{x}), \quad \boldsymbol{c}_{AD} \doteq [c_1^{AD}, \ldots, c_r^{AD}]^\top. \tag{11}$$

That is, each of the different functions $\boldsymbol{u}_r$ (unfiltered ROM velocity), $\overline{\boldsymbol{u}}_r$ (filtered ROM velocity), and $\boldsymbol{u}_r^{AD}$ (ROM approximate deconvolution) is a linear combination of the same ROM basis functions $\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_r$ with different coefficients.

Remark 3.1. To simplify the presentation, in the rest of this section we assume that $\boldsymbol{U}_h = \boldsymbol{0}$ and that the NSE (1)–(2) are equipped with homogeneous Dirichlet boundary conditions. Thus, the centering trajectory, $\boldsymbol{U}_h$, will drop from (9)–(11). We note, however, that a nonzero centering trajectory will actually be required in Section 5.2 to handle the nonzero inlet conditions. The methodology introduced in this section readily extends to that case, which yields additional terms on the right-hand side of (14), (18), (21), and (23). For instance, following the same approach as that used in (6), we obtain the following modification of (14):

$$\left(\boldsymbol{M} + \delta^2 \boldsymbol{S}\right)\overline{\boldsymbol{c}} = \boldsymbol{M}\boldsymbol{c} + \boldsymbol{g},$$

where

$$\boldsymbol{g}_i = -\left(\boldsymbol{\varphi}_i, \boldsymbol{U}_h\right) - \delta^2\left(\nabla\boldsymbol{\varphi}_i, \nabla\boldsymbol{U}_h\right).$$

3.1. The ROM differential filter

To develop the ADL-ROM, we will use the *ROM differential filter (ROM-DF)*:

$$\overline{\boldsymbol{u}} = G\boldsymbol{u} \doteq (I - \delta^2 \Delta)^{-1}\boldsymbol{u}. \tag{12}$$

That is, given a function $\boldsymbol{u}$, we filter it with the ROM-DF $G = (I - \delta^2\Delta)^{-1}$ to obtain a filtered function $\overline{\boldsymbol{u}}$. A filter radius, δ, appears in the definition of G, and is set by the user. The input $\boldsymbol{u}$ will typically be the velocity of the NSE (1)–(2). We note that ROM-DF (12) is equivalent to solving the PDE

$$\begin{aligned} (I - \delta^2\Delta)\overline{\boldsymbol{u}} &= \boldsymbol{u} \qquad \text{in } \Omega, \\ \overline{\boldsymbol{u}} &= \boldsymbol{0} \qquad \text{on } \partial\Omega, \end{aligned} \tag{13}$$

where $\partial\Omega$ is the boundary of the spatial domain, Ω.

We notice that Eqs. (12)–(13) are reported in strong form, and for any input function $\boldsymbol{u}$. In practice, however, we apply this filter to the ROM velocity, $\boldsymbol{u}_r$, and do so by solving the ROM linear system corresponding to the application of the Galerkin method for (13) on the space spanned by the ROM basis functions $\{\boldsymbol{\varphi}_1, \dots, \boldsymbol{\varphi}_r\}$:

$$\left(\boldsymbol{M} + \delta^2 \boldsymbol{S}\right) \overline{\boldsymbol{c}} = \boldsymbol{M} \boldsymbol{c}, \tag{14}$$

where $\boldsymbol{M}$ is the POD mass matrix (which is the identity matrix since the POD basis functions are orthonormal) and $\boldsymbol{S}$ is the POD stiffness matrix, whose entries are defined as follows:

$$\boldsymbol{M}_{im} = \left(\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_m\right), \qquad \boldsymbol{S}_{im} = \left(\nabla\boldsymbol{\varphi}_i, \nabla\boldsymbol{\varphi}_m\right), \qquad i, m = 1, \dots, r.$$

The goal is now to obtain $\boldsymbol{u}_r^{AD} \approx G^{-1}\overline{\boldsymbol{u}}_r$ without resorting to the inverse operator G^{-1}, but applying instead an AD operator. Indeed, applying directly the filter inversion would result in an algorithm characterized by poor conditioning and ill-posedness [45,46,52]. Three AD operators are introduced below.

3.2. The van Cittert AD

Given an integer N, which represents the order of the AD operator, the van Cittert approximate deconvolution operator is given by

$$\boldsymbol{u}_{AD} = D_N \overline{\boldsymbol{u}} = \sum_{n=0}^{N} (I - G)^n \overline{\boldsymbol{u}}, \tag{15}$$

presented in [53, section 3.1.1]. To solve this, we use a Richardson iteration, described in [53, section 3.3.4]

$$\boldsymbol{u}_{AD}^{(n+1)} = \boldsymbol{u}_{AD}^{(n)} + \{\overline{\boldsymbol{u}} - G\boldsymbol{u}_{AD}^{(n)}\}, \qquad n = 0, \dots, N-1, \tag{16}$$

where the initial $\boldsymbol{u}_{AD}^{(0)}$ is set as $\overline{\boldsymbol{u}}$. By running the iteration (16) N times, we obtain $\boldsymbol{u}_{AD} \doteq \boldsymbol{u}_{AD}^{(N)}$ as the Nth order van Cittert operator applied to $\overline{\boldsymbol{u}}$. However, to make this iteration practical, we need to recall that each application of the filter G requires the inversion of a differential operator. Therefore, denoting by $\widetilde{\boldsymbol{u}}^{(n+1)}$ the quantity $\widetilde{\boldsymbol{u}}^{(n+1)} \doteq G\boldsymbol{u}_{AD}^{(n)}$, and substituting in our filter from (12), we observe that $\widetilde{\boldsymbol{u}}^{(n+1)}$ is found by solving

$$(I - \delta^2 \Delta)^{-1} \boldsymbol{u}_{AD}^{(n)} = \widetilde{\boldsymbol{u}}^{(n+1)} \Longleftrightarrow (I - \delta^2 \Delta)\widetilde{\boldsymbol{u}}^{(n+1)} = \boldsymbol{u}_{AD}^{(n)}. \tag{17}$$

Eq. (17) requires to solve a linear system at each iteration $n = 0, \dots, N-1$. As discussed above, in order to employ the van Cittert AD in a ROM setting, by multiplying (17) by each test function in our ROM space and expanding our prospective solution as a linear combination of ROM basis functions, we obtain the linear system

$$\left(\boldsymbol{M} + \delta^2 \boldsymbol{S}\right) \widetilde{\boldsymbol{c}}^{(n+1)} = \boldsymbol{M} \boldsymbol{c}_{AD}^{n}. \tag{18}$$

Thus, the iterative process (16) amounts to setting $\boldsymbol{c}_{AD}^{(0)} = \overline{\boldsymbol{c}}$, updating the coefficients of the ROM AD velocity as

$$\boldsymbol{c}_{AD}^{(n+1)} = \boldsymbol{c}_{AD}^{(n)} + \{\overline{\boldsymbol{c}} - \widetilde{\boldsymbol{c}}^{(n+1)}\} \qquad n = 0, \dots, N-1,$$

and finally defining $\boldsymbol{c}_{AD} \doteq \boldsymbol{c}_{AD}^{(N)}$.

3.3. The Tikhonov AD

The Tikhonov method of approximate deconvolution is defined as

$$\boldsymbol{u}_{AD} = D_\mu^T \overline{\boldsymbol{u}} = (G^* G + \mu I)^{-1} G^* \overline{\boldsymbol{u}}, \tag{19}$$

where G^* denotes the adjoint of the operator G, and $\mu \in \mathbb{R}^+$ is a positive constant; we refer, e.g., to [53, section 3.3.1] for further details on the Tikhonov AD. When plugging in the specific filter from (12) and proceeding formally, we can write

$$\begin{aligned}
[G^*G + \mu I]\boldsymbol{u}_{AD} &= G^*\overline{\boldsymbol{u}}, \\
[(I - \delta^2\Delta)^{-*}(I - \delta^2\Delta)^{-1} + \mu I]\boldsymbol{u}_{AD} &= (I - \delta^2\Delta)^{-*}\overline{\boldsymbol{u}}], \\
[(I - \delta^2\Delta)^{-1} + \mu(I - \delta^2\Delta)^*]\boldsymbol{u}_{AD} &= \overline{\boldsymbol{u}}, \\
[I + \mu(I - \delta^2\Delta)(I - \delta^2\Delta)^*]\boldsymbol{u}_{AD} &= (I - \delta^2\Delta)\overline{\boldsymbol{u}}, \\
[I + \mu(I - \delta^2\Delta^* - \delta^2\Delta + \delta^2\Delta\delta^2\Delta^*)]\boldsymbol{u}_{AD} &= (I - \delta^2\Delta)\overline{\boldsymbol{u}}, \\
\left[I + \mu(I - 2\delta^2\Delta + \delta^4\Delta\Delta)\right]\boldsymbol{u}_{AD} &= (I - \delta^2\Delta)\overline{\boldsymbol{u}}.
\end{aligned} \tag{20}$$

In (20), we have used the fact that the Laplacian is self-adjoint. In the numerical comparison of the three AD approaches in Section 5.2, for ease of implementation in our FE setting, we neglect the term $\delta^4\Delta^2$. Thus, expanding our desired solution as linear combinations of ROM basis functions, we solve for the vector of coefficients $\boldsymbol{c}_{AD}$ satisfying

$$[(1 + \mu)\boldsymbol{M} + 2\mu\delta^2\boldsymbol{S}]\boldsymbol{c}_{AD} = (\boldsymbol{M} + \delta^2\boldsymbol{S})\overline{\boldsymbol{c}}. \tag{21}$$

3.4. The Lavrentiev AD

The Lavrentiev method [48], which is the AD strategy we use in our numerical investigation in Section 5, is a modification of the Tikhonov method suitable for self-adjoint, positive definite operators. The filter described in (12) has these properties, as shown in [48]. Given $\mu \in \mathbb{R}^+$, the Lavrentiev AD operator is given in [53, section 3.3.2] as

$$\boldsymbol{u}_{AD} = D^L_\mu \overline{\boldsymbol{u}} = (G + \mu I)^{-1} \overline{\boldsymbol{u}}. \tag{22}$$

We solve for $\boldsymbol{u}_{AD}$ by substituting in G

$$\begin{aligned}
\boldsymbol{u}_{AD} &= \left[(I - \delta^2 \Delta)^{-1} + \mu I\right]^{-1} \overline{\boldsymbol{u}},\\
\left[(I - \delta^2 \Delta)^{-1} + \mu I\right] \boldsymbol{u}_{AD} &= \overline{\boldsymbol{u}},\\
\left[I + \mu(I - \delta^2 \Delta)\right] \boldsymbol{u}_{AD} &= (I - \delta^2 \Delta)\overline{\boldsymbol{u}},\\
\left[(1+\mu)I - \mu\delta^2 \Delta\right] \boldsymbol{u}_{AD} &= (I - \delta^2 \Delta)\overline{\boldsymbol{u}}.
\end{aligned}$$

Expanding our solution as a linear combination of ROM basis functions, multiplying by a test function, and integrating by parts, we obtain the system

$$\left[(1+\mu)\boldsymbol{M} + \mu\delta^2 \boldsymbol{S}\right] \boldsymbol{c}_{AD} = \left(\boldsymbol{M} + \delta^2 \boldsymbol{S}\right) \overline{\boldsymbol{c}} \tag{23}$$

and solve for the vector of coefficients $\boldsymbol{c}_{AD}$.

4. Reg-ROMs

In this section, we present the Reg-ROMs that will be used in the numerical investigation in Section 5. First, in Section 4.1, we present the standard L-ROM. Then, in Section 4.2, we introduce the novel ADL-ROM.

4.1. L-ROM

The initial use of the Leray model in 1934 by Jean Leray [32] was aimed at providing a theoretical framework for establishing the existence of weak solutions for the NSE. More recently, researchers have employed the Leray model as a computational tool for simulating convection-dominated flows (e.g., turbulent flows), utilizing conventional numerical methods [33], such as the finite element method [53]. We note that, when the differential filter is used, the Leray model is similar to the NS-α model advocated by Foias, Holm, Titi, and their collaborators [54]. During the last decade, the Leray model was extended to the realm of reduced order modeling, first for the Kuramoto–Sivashinsky equations [27], and then for the NSE [29,55] and the quasi-geostrophic equations [37,38].

In our setting, the L-ROM time discretization reads as follows: Given $\boldsymbol{u}_r^n$ and $\boldsymbol{u}_r^{n-1}$, find $\boldsymbol{u}_r^{n+1}$ such that

$$\left(\frac{\boldsymbol{u}_r^{n+1} - \frac{4}{3}\boldsymbol{u}_r^n + \frac{1}{3}\boldsymbol{u}_r^{n-1}}{\Delta t}, \boldsymbol{\varphi}_i\right) + \frac{2}{3} Re^{-1} \left(\nabla \boldsymbol{u}_r^{n+1}, \nabla \boldsymbol{\varphi}_i\right) + \frac{2}{3}\left(\left(\overline{\boldsymbol{u}}_r^{n+1} \cdot \nabla\right) \boldsymbol{u}_r^{n+1}, \boldsymbol{\varphi}_i\right) = 0, \tag{24}$$

$\forall n = 1, \dots, M-1$ and $\forall i = 1, \dots, r$, where Δt is the time step. Utilizing the ROM-DF (12) outlined in Section 3.1, the filtered convective term in (24) is defined as follows:

$$\overline{\boldsymbol{u}}_r^{n+1}(\boldsymbol{x}, t) \doteq \sum_{j=1}^{r} \overline{c}_j^{n+1}(t)\boldsymbol{\varphi}_j(\boldsymbol{x}). \tag{25}$$

The coefficients $\overline{\boldsymbol{c}}^{n+1}$ in (25) are found by solving the following reduced linear system:

$$\left(\boldsymbol{M} + \delta^2 \boldsymbol{S}\right) \overline{\boldsymbol{c}}^{n+1} = \boldsymbol{M}\boldsymbol{c}^{n+1}, \tag{26}$$

where $\boldsymbol{M}$ and $\boldsymbol{S}$ are the POD mass and stiffness matrices, respectively, δ is the radius of the ROM-DF, and $\boldsymbol{c}^{n+1}$ is the vector of reduced coefficients of the input variable, $\boldsymbol{u}_r^{n+1}$. We note that linear system (26) is linear system (14) for input coefficients $\boldsymbol{c}^{n+1}$. We also note that a second-order backward differences (BDF2) time discretization was used in (24), but other time discretizations are possible.

Thus, at the time instance t^{n+1}, the L-ROM time discretization yields the following nonlinear system:

$$\frac{1}{\Delta t}\boldsymbol{M}\left(\boldsymbol{c}^{n+1} - \frac{4}{3}\boldsymbol{c}^n + \frac{1}{3}\boldsymbol{c}^{n-1}\right) + \frac{2}{3}Re^{-1}\boldsymbol{S}\boldsymbol{c}^{n+1} + \frac{2}{3}\boldsymbol{C}(\overline{\boldsymbol{c}}^{n+1})\boldsymbol{c}^{n+1} = \boldsymbol{0}, \tag{27}$$

where $\boldsymbol{c}^{n+1}$ is the unknown reduced coefficient vector of the unfiltered solution at time t^{n+1}, $\overline{\boldsymbol{c}}^{n+1}$ the filtered velocity coefficient vector at time t^{n+1}, and the matrix $\boldsymbol{C}(\overline{\boldsymbol{c}}^{n+1})$ is defined as

$$\boldsymbol{C}(\overline{\boldsymbol{c}}^{n+1})_{ij} = \left((\overline{\boldsymbol{u}}_r^{n+1} \cdot \nabla)\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_j\right). \tag{28}$$

System (27) is a nonlinear system of equations in the unknowns $\boldsymbol{c}^{n+1}$, which we solve by using the Newton method. In particular, at each iteration of the Newton method, given the current approximation to $\boldsymbol{c}^{n+1}$, we update the vector $\overline{\boldsymbol{c}}^{n+1}$ via (26), and evaluate the residual of (27). We notice that the additional linear system (26) to be solved at each nonlinear iteration is still of small dimension $r \times r$.

In Algorithm 1, we outline the main steps of the L-ROM discretization.

Algorithm 1 L-ROM Pseudocode

```
$\boldsymbol{u}_{-1}$, $\boldsymbol{u}_0$, $\boldsymbol{u}_{in}$, $r$                              ▷ Inputs needed
for $n \in \{1, \dots, M-1\}$ do                                          ▷ Time loop
    FOM simulation to compute $\boldsymbol{u}_h^{n+1}$                    ▷ Snapshot collection
end for
$\mathrm{U}^r \doteq \mathrm{POD}\left(\left\{\boldsymbol{u}_h^n\right\}_{n=1}^M ; r\right)$    ▷ POD for velocity space
for $n \in \{1, \dots, M-1\}$ do                                          ▷ Time loop
    Solve (27) to compute $\boldsymbol{c}^{n+1}$ which requires to         ▷ L-ROM
      solve (26) at each nonlinear iteration                              ▷ Leray filtering
end for
```

L-ROM is a significant improvement over the standard G-ROM (4) since spatial filtering alleviates the spurious numerical oscillations displayed by G-ROM in convection-dominated flows. However, when the ROM filter radius, δ, is too large (i.e., the ROM filtering is too aggressive), L-ROM can be overdiffusive and can yield inaccurate results, as shown in the numerical investigation in Section 5. Furthermore, numerical investigations have revealed that L-ROM is sensitive with respect to δ: small δ variations can significantly decrease the L-ROM accuracy.

4.2. ADL-ROM

To address these L-ROM drawbacks, in this section we propose a novel Reg-ROM, the *approximate deconvolution Leray ROM (ADL-ROM)*, which is inspired from earlier work in the finite element setting based on a clever idea of Adrian Dunca [53,56,57]. The new ADL-ROM is based on a simple yet powerful idea: Replace the filtered L-ROM velocity, $\overline{\boldsymbol{u}}$, with an approximate deconvolution velocity, $D(\overline{\boldsymbol{u}})$, where D is one of the AD operators defined in Section 3. The numerical investigation in Section 5 shows that using the AD operator in the ADL-ROM increases the accuracy of the L-ROM nonlinear term without compromising the L-ROM stability. Thus, in a sense, ADL-ROM can be regarded as a compromise between the numerically unstable G-ROM and the potentially overdiffusive L-ROM.

In our setting, the ADL-ROM time discretization reads as follows: Given $\boldsymbol{u}_r^n$ and $\boldsymbol{u}_r^{n-1}$, find $\boldsymbol{u}_r^{n+1}$ such that

$$\left(\frac{\boldsymbol{u}_r^{n+1} - \frac{4}{3}\boldsymbol{u}_r^n + \frac{1}{3}\boldsymbol{u}_r^{n-1}}{\Delta t}, \boldsymbol{\varphi}_i\right) + \frac{2}{3} Re^{-1} \left(\nabla \boldsymbol{u}_r^{n+1}, \nabla \boldsymbol{\varphi}_i\right) + \frac{2}{3}\left(\left(\mathrm{D}_\mu^L(\overline{\boldsymbol{u}}_r^{n+1}) \cdot \nabla\right) \boldsymbol{u}_r^{n+1}, \boldsymbol{\varphi}_i\right) = 0, \tag{29}$$

$\forall n = 1, \dots, M-1$ and $\forall i = 1, \dots, r$.

Utilizing the ROM differential filter (12) outlined in Section 3.1 and the Lavrentiev AD operator (22) in Section 3, the AD convective term in (29) is defined as follows:

$$\boldsymbol{u}_{AD,\ r}^{n+1}(\boldsymbol{x}, t) \doteq \mathrm{D}_\mu^L(\overline{\boldsymbol{u}}_r^{n+1}) \doteq \sum_{j=1}^{r} c_{AD,\ j}^{n+1}(t) \boldsymbol{\varphi}_j(\boldsymbol{x}). \tag{30}$$

The coefficients $\boldsymbol{c}_{AD}^{n+1}$ in (30) are found by solving the reduced linear system in (23), upon providing the current approximation of the filtered coefficients $\overline{\boldsymbol{c}}^{n+1}$, which in turn are computed by solving the linear system (26). Although we utilize the BDF2 method for the ADL-ROM time discretization (29), other time discretizations are possible.

Thus, at the time instance t^{n+1}, the ADL-ROM time discretization yields the following nonlinear system:

$$\frac{1}{\Delta t}\boldsymbol{M}\left(\boldsymbol{c}^{n+1} - \frac{4}{3}\boldsymbol{c}^n + \frac{1}{3}\boldsymbol{c}^{n-1}\right) + \frac{2}{3} Re^{-1} \boldsymbol{S}\boldsymbol{c}^{n+1} + \frac{2}{3}\boldsymbol{C}(\boldsymbol{c}_{AD}^{n+1})\boldsymbol{c}^{n+1} = \boldsymbol{0}, \tag{31}$$

where $\boldsymbol{c}^{n+1}$ are the unknown reduced coefficient vectors of the unfiltered velocity field, and $\boldsymbol{c}_{AD}^{n+1}$ the vectors of the corresponding AD ROM velocity field obtained by solving (23) and (26). Therefore, each iteration of the Newton solver requires to solve two small $r \times r$ linear systems before the evaluation of the residual.

In Algorithm 2, we outline the main steps of the ADL-ROM discretization.

5. Numerical results

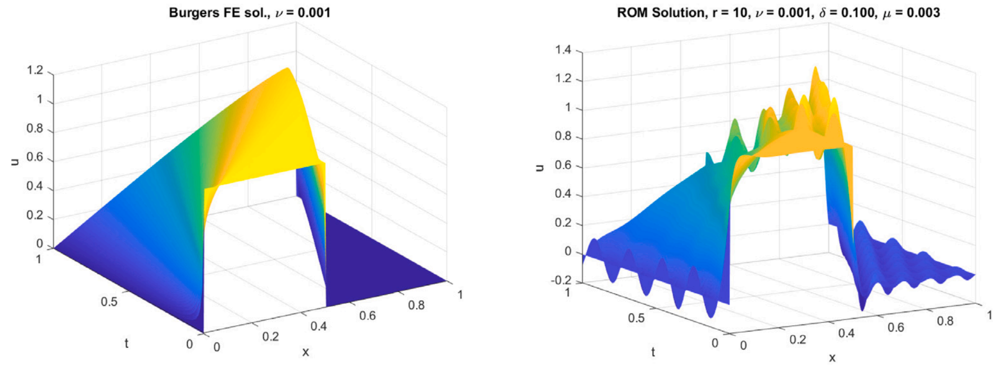
To maintain consistency with the only previous time AD methods were used in ROMs [48] and for reasons discussed in Section 5.2, the rest of this paper will focus on the Lavrentiev AD method described in Section 3.4. Further references to the ADL-ROM in our numerical results should be understood to refer to ADL-ROMs constructed with the Lavrentiev regularization method.

5.1. Burgers equation

Goals. The following is an investigation of ADL-ROM performance in the simpler setting of Burgers equation, which is a common test problem for new ROM methods, see, e.g., [58–61].

Algorithm 2 ADL-ROM Pseudocode

```
$\boldsymbol{u}_{-1}$, $\boldsymbol{u}_0$, $\boldsymbol{u}_{in}$, $r$                              ▷ Inputs needed
for $n \in \{1, \dots, M-1\}$ do                                        ▷ Time loop
    FOM simulation to compute $\boldsymbol{u}_h^{n+1}$                         ▷ Snapshot collection
end for
$\mathbf{U}^r \doteq \mathrm{POD}\left(\left\{\boldsymbol{u}_h^n\right\}_{n=1}^{M}; r\right)$                  ▷ POD for velocity space
for $n \in \{1, \dots, M-1\}$ do                                        ▷ Time loop
    Solve (31) to compute $\boldsymbol{c}^{n+1}$ which requires to                 ▷ L-ROM
      solve (26) at each nonlinear iteration, and to                 ▷ Leray filtering
      solve (23) at each nonlinear iteration                         ▷ AD
end for
```


Burgers FE sol., ν = 0.001

ROM Solution, r = 10, ν = 0.001, δ = 0.100, μ = 0.003

(a) FEM Solution

(b) G-ROM, 10 basis functions

Fig. 1. Burgers equation [62]: (a) FEM solution; (b) G-ROM solution.

Computational setting. Burgers equation is a nonlinear problem of one spatial dimension given as

$$u_t - \nu u_{xx} + u u_x = 0. \tag{32}$$

Here, $u \doteq u(x,t)$ is defined for $x \in \Omega = [0,1]$, $t \in (0,1]$, and we consider the low viscosity setting where ν is small. A typical form of Burgers equation to make it more challenging in a computational setting is to use a discontinuous initial condition:

$$u(x,0) = \begin{cases} 1, & 0 < x < 0.5, \\ 0, & x = 0,\ 0.5 \le x \le 1. \end{cases} \tag{33}$$

Because of the similarity in structure between Burgers equation and the NSE, it may be solved at a FE and ROM level using methods similar to those discussed in Section 2, upon neglecting the pressure term in (1) and the divergence constraint (2).

Criteria. We will consider the errors between a well resolved FE solution $u(t)$ and a ROM solution $u_r(t)$ as

$$E_u^a(t) \doteq \left\| u(t) - u_r(t) \right\|_{L^2(\Omega)}. \tag{34}$$

As a comparison between the ADL-ROM and the standard G-ROM, we will also employ a relative reduction in error defined as

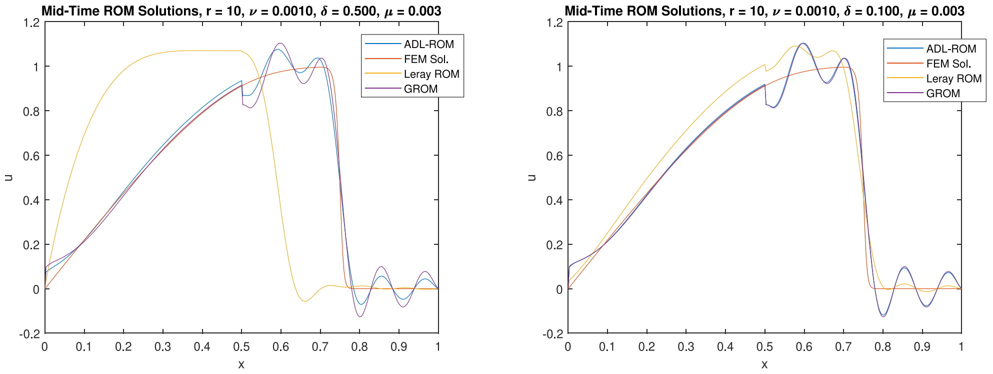
$$RE(t) = -100 \cdot \frac{\left(E_u^a(t)_{G-ROM} - E_u^a(t)_{ADL-ROM}\right)}{E_u^a(t)_{G-ROM}}. \tag{35}$$

With this formula, negative percent values indicate that the ADL-ROM is performing better than the G-ROM.

Numerical results. The plots in Figs. 1(a)–1(b) show that, even with a well resolved FE solution, the nonlinear nature of the problem as well as the challenging initial condition present problems for the standard G-ROM.

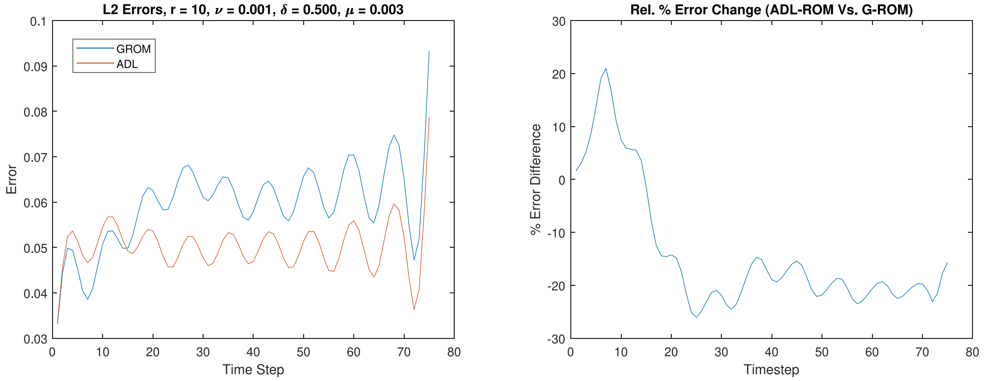
The FE solution in Fig. 1(a) was computed with 350 evenly spaced elements between 0 and 1 and piecewise linear finite element functions. Both the FE solution in Fig. 1(a) and the G-ROM solution in Fig. 1(b) were computed with an implicit Euler time discretization and using a Newton solver, 75 even time steps between 0 and 1, and $\nu = 10^{-3}$. Even though the FE solution is completely resolved, the ROM displays severe oscillations. As shown below, using instead an ADL-ROM constructed with the Lavrentiev regularization method, we are able to lessen the oscillations.

We emphasize that, even though we are using a smoothing process, it is important to avoid over smoothing. The objective is to minimize non-physical phenomena, not replace one (oscillations) with another (excessive smoothing). In Fig. 2(a), the Leray



(a) ROM solutions for $\delta = 0.5$. (b) ROM solutions for $\delta = 0.1$.

Fig. 2. Comparison of ROM solutions at $t = 0.5$ with two δ values.



(a) L^2 errors per time step (b) Relative % error change (Eqn. (35))

Fig. 3. ADL-ROM errors [62]: $\delta = 0.5$, $\mu = 0.003$.

Table 1
Mean absolute L^2 errors across all time steps.

Model	G-ROM	L-ROM	ADL-ROM
Mean Error	5.94×10^{-2}	4.24×10^{-1}	5.03×10^{-2}

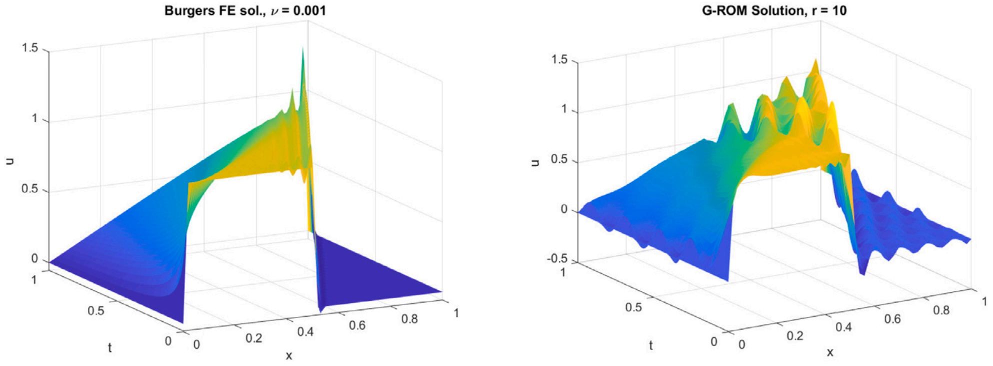
ROM with a large filter radius of $\delta = 0.5$ produces a very smooth solution which is incompatible with the FE solution, while the ADL-ROM tapers the oscillations present in the G-ROM. In Fig. 2(b), a more moderate value of $\delta = 0.1$ produces a L-ROM which is still over-smoothed, though the ADL-ROM is now nearly identical to the G-ROM.

The plots in Fig. 2 demonstrate that the ADL-ROM is able to accommodate a larger filter radius than the L-ROM. Our investigation suggests that the larger values of δ, near $\delta = 0.5$, are more suitable for minimizing our definition of error given in (34). Additionally, the dramatic differences between the L-ROM in Figs. 2(a) and 2(b) as compared to the ADL-ROM solutions show that the addition of the parameter μ causes the ADL-ROM solutions to vary less dramatically with changes in δ.

While the effects of the ADL-ROM shown in Fig. 2(a) are mild at each time step, the errors in Figs. 3(a) and 3(b) show that a reasonable reduction in error can be obtained from the ADL-ROM, which is only a single additional linear solve from the L-ROM. In Table 1, the mean error of the L-ROM are of order 10^{-1}, while the other two methods are of order 10^{-2}, so the L-ROM errors for each time step are not plotted in Fig. 3(a) to avoid distorting the scale.

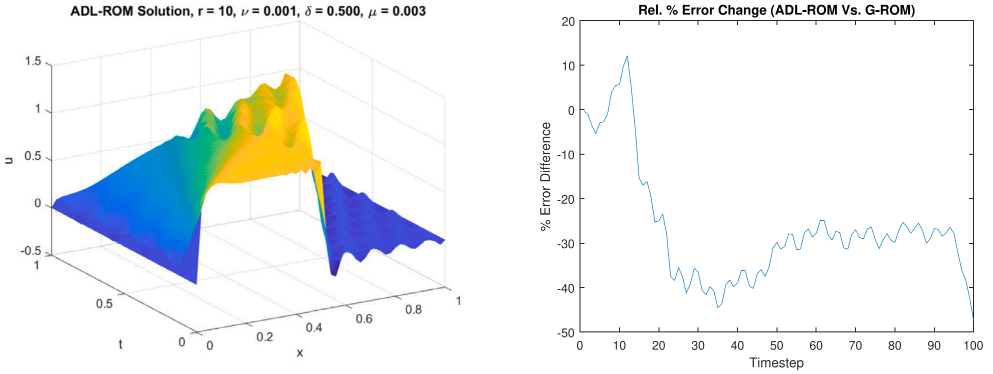
These tests should not be taken to suggest that the L-ROM cannot be tuned provide acceptable results, but that the ADL-ROM is a minimal cost adaption to the L-ROM, the main benefit of which is the ability to choose a larger radius filter than is possible with the L-ROM. The second parameter μ is an additional complexity in choice, but at the same time the solutions generated by the ADL-ROM are less variable than those generated by the L-ROM.

Additional test. Following on the observation that the ADL-ROM allows larger filter values of δ, the following test aims to increase the numerical oscillations in the solution and assess the ADL-ROM's performance in a more challenging setting. If we take the same initial condition as in but decrease the number of evenly spaced elements from 350 to 50, the shock will fail to be properly resolved



(a) FE solution (b) G-ROM, 10 basis functions

Fig. 4. The G-ROM propagates the oscillations present in the FE solution.



(a) ADL-ROM solution (b) Relative % error change (Eqn. (35))

Fig. 5. The ADL-ROM damps the oscillations.

Table 2
Mean absolute L^2 errors across all time steps.

Model	G-ROM	L-ROM	ADL-ROM
Mean Error	8.05×10^{-2}	4.24×10^{-1}	5.68×10^{-2}

in the FE simulation and cause cascading oscillations in Fig. 4(a), which are propagated into the G-ROM of Fig. 4(b). The images in Fig. 4 were generated with $\nu = 10^{-3}$, 100 evenly spaced timesteps between 0 and 1, but only 50 evenly spaced elements.

Because the input FE data is poor, this necessitates a change in how the errors in Fig. 5(b) are calculated. Instead of comparing against the poor input FE data, we compute a higher resolution FE solution. In Fig. 5(b), the reduction in error is computed by using a FE solution with 200 evenly spaced elements and 100 timesteps.

With ADL parameters chosen as $\delta = 0.5$, $\mu = 0.003$, the ADL-ROM shown in Fig. 5 is able to substantially quell the oscillations inherent in the FE and G-ROM solutions. Comparing between Tables 1 and 2, the ADL-ROM is able to maintain its accuracy significantly better than the G-ROM is when downgrading the input data. The leading digits of error for the L-ROM are the same in both cases because the L-ROM fails in the same way in both cases by severely over smoothing the computed solution.

These results suggest that the ADL-ROM performs best when it is applied as a treatment for significant oscillations in the numerical solution. In particular, this suggests the applicability of the method for use in convection-dominated flow simulations, such as the ones we propose for the NSE.

5.2. Backward facing step

Goals. In this section, we analyze and compare the performances of G-ROM (4), L-ROM (24), and ADL-ROM (29) for the NSE (1)–(2). The goal is to investigate ADL-ROM accuracy. We consider the two-dimensional velocity flow over the backward-facing step at Reynolds number $Re = 1429$.

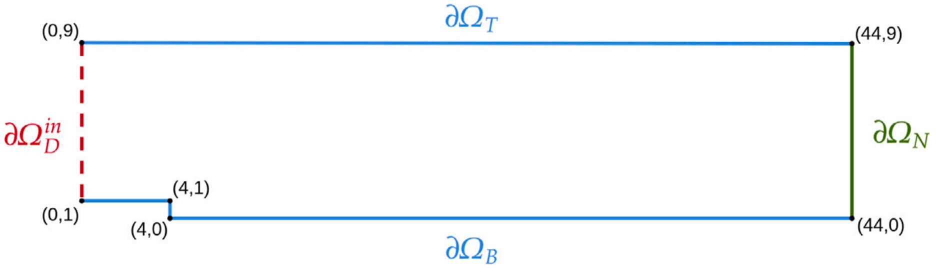


Fig. 6. The computational domain, Ω. $\partial\Omega_D \doteq \partial\Omega_D^{in} \cup \partial\Omega_D^{\text{wall}}$, where the inlet boundary $\partial\Omega_D^{in}$ is represented by a dashed red line and the no-slip boundaries by a solid blue line.

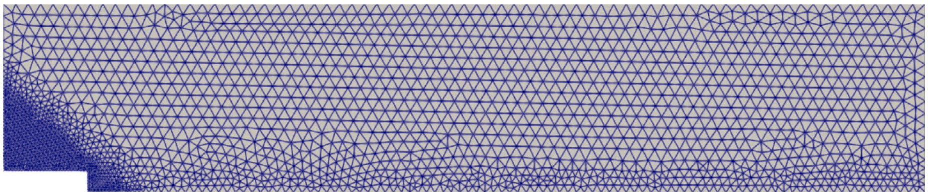


Fig. 7. The FE mesh.

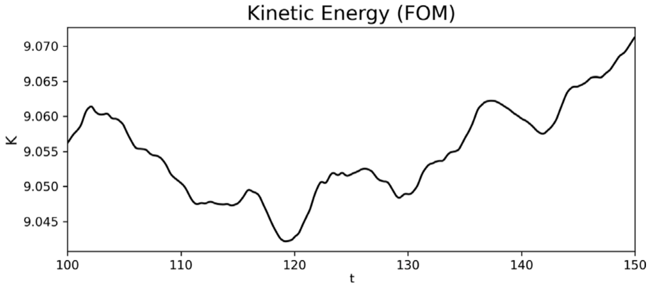


Fig. 8. Time evolution of FOM kinetic energy.

Computational setting. We consider the motion of an incompressible flow in the same domain as that used in [63, section 4.4] and in [64, section 3.5], i.e, $\Omega \doteq \{[0,44]\times[0,9]\}\backslash\{[0,4]\times[0,1]\}$, which is depicted in Fig. 6. We set a kinematic viscosity of 7×10^{-4} and use no-slip boundary conditions on $\partial\Omega_D^{\text{wall}} \doteq \partial\Omega_{\text{B}}\cup\partial\Omega_{\text{T}}$, representing the union of the bottom $\partial\Omega_{\text{B}}$ and top walls $\partial\Omega_{\text{T}}$ of the channel (solid blue boundary in Fig. 6), with a constant inlet velocity profile $\boldsymbol{u}_{in} = (1,0)$ on $\partial\Omega_D^{in}$ (red dashed line in Fig. 6). Furthermore, we employ homogeneous Neumann boundary conditions on $\partial\Omega_N$ (green line in Fig. 6).

Snapshot generation. We perform our tests on a triangular mesh with $h_{\text{min}} = 0.15$ and $h_{\text{max}} = 0.6$ (Fig. 7). For the spatial discretization, we employ the inf-sup stable Taylor-Hood $\mathbb{P}^2 - \mathbb{P}^1$ FE pair for velocity and pressure, respectively, and this leads to a FE space of dimension $N_h \doteq N_h^u + N_h^p = 18404 + 2370 = 20774$. A second-order BDF2 scheme is used with the time step $\Delta t = 0.05$ for both FOM and ROM time discretizations. The time interval on which FOM is performed is $[T_0, T] = [0, 150]$. The value of the initial conditions $\boldsymbol{u}_{-1}$ and $\boldsymbol{u}_0$ is $(\boldsymbol{0},\boldsymbol{0})$.

At the FOM level, since we work in the under-resolved regime, we apply a regularization strategy. Specifically, we employ the evolve-filter-relax (EFR) strategy (for details, see [65]). The EFR stabilization strategy allows us to obtain accurate results in the convection-dominated regime on the coarse mesh in Fig. 7. We use EFR as a stabilization strategy for the following reasons: EFR has been widely used at the FOM level, is easy to implement, is effective, and has been used successfully in our numerical investigations [29,31,55,62,65,66]. We emphasize, however, that other regularization strategies (e.g., Leray or even ADL, which would ensure FOM-ROM consistency [65]) or stabilization approaches could be investigated. This could be subject of future work. In Fig. 8, we plot the time evolution of the FOM kinetic energy on the time interval $[100, 150]$. This plot shows that the flow is not periodic or periodic-like.

ROM construction. To build the ROM basis functions, we collect 1000 snapshots of the velocity field over the time interval $[100.05, 150]$, as in [64,67]. The first $r = 10$ POD modes capture about 80% of the flow's energy. We limit ourselves to a small number of POD basis functions for the purpose of showing the effect of regularization when the ROM cannot accurately describe all

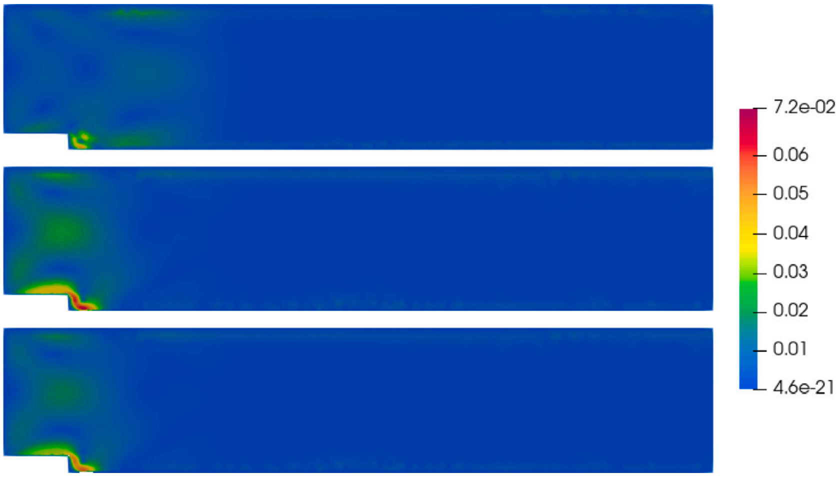


Fig. 9. Pointwise difference between AD applied to G-ROM and G-ROM, $\delta = 0.6$. 1st: Lavrentiev with $\mu = 0.1$. 2nd: Tikhonov with $\mu = 0.1$. 3rd: van Cittert with $N = 10$.

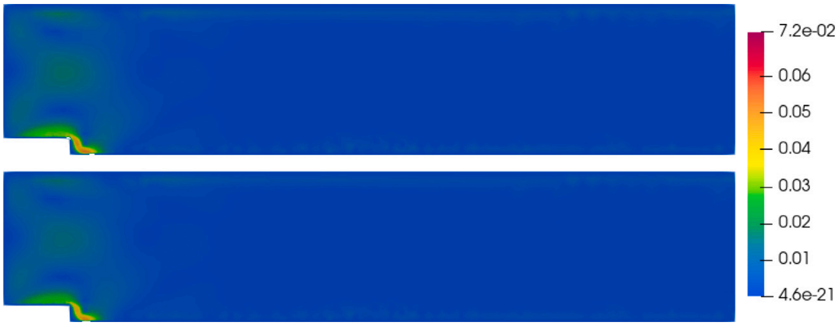


Fig. 10. Pointwise difference between AD applied to G-ROM and G-ROM, $\delta = 0.6$. 1st: Tikhonov with $\mu = 0.05$. 2nd: van Cittert with $N = 12$.

scales of motion, which is the case in many realistic applications. These modes are used to construct all ROMs. All the ROMs are investigated on the time interval $[100.05, 150]$.

AD comparison. Before proceeding to the discussion of ADL-ROM results, we report here on the choice of AD to use given the options discussed in Section 3. To do this, we will briefly examine the effects that these AD methods have on a standard G-ROM solution. The G-ROM solution will be calculated following the methods discussed in Section 2 and in the previous paragraph to a final time of 20 time units with a fixed step length of 0.01 time units.

In Figs. 9 and 10, we plot the pointwise difference between the computed G-ROM solution at $t = 20$ and each of the AD methods applied to the G-ROM solution. Fig. 9 also shows that, with AD parameters chosen as labeled, the Lavrentiev AD method produces results which are the least similar to the Tikhonov or van Cittert methods. However, this is mostly a matter of tuning: Fig. 10 shows that altering the AD parameters of μ for the Tikhonov or N for the van Cittert method will make those methods produce very similar results as to the Lavrentiev method in the first plot of Fig. 9.

Each of the AD methods shown in Figs. 9 and 10 affects the computed G-ROM solution most strongly near the backward facing step, which is the primary place where vortices are formed and is where we want the smoothing to be focused. Additionally, by varying either δ in the original filter or one of the AD parameters μ or N as in Fig. 10, each of the AD regularization methods can produce similar results to the other methods. For these reasons, the numerical results of this paper focus on the Lavrentiev AD method described in Section 3.4.

Parameters. As mentioned before, the inlet velocity and the size of the step lead to a Reynolds number $Re = 1429$. For the EFR FOM stabilization strategy, there are two parameters: the filtering radius of the differential filter, δ_{EFR}, employed in the *Filter* step, and the relaxation parameter, $\chi \in [0, 1]$, employed in the *Relax* step. For the rest of the article, we present results for $\delta_{EFR} = 0.01$ and $\chi = 0.01$. This choice limits the amount of dissipation introduced by the differential filter in the EFR algorithm.

One of the most important parameters of L-ROM and ADL-ROM is the radius of the ROM differential filter introduced in Section 3.1, δ. To avoid confusion between the filtering radius of the EFR step δ_{EFR}, and the filtering radius of the ROM-DF, in the discussion we will denote by δ_{ROM} the radius of the ROM-DF, instead of δ. To illustrate the case of an aggressive choice of the

Table 3
Average absolute and relative L^2 errors for G-ROM, L-ROM, and ADL-ROM.

	Average L^2 errors	
Model	$\mathcal{E}^a(L^2)$	$\mathcal{E}^r(L^2)$
G-ROM	1.2668	7.25e−02
L-ROM	1.6061	8.88e−02
ADL-ROM	1.0144	5.60e−02

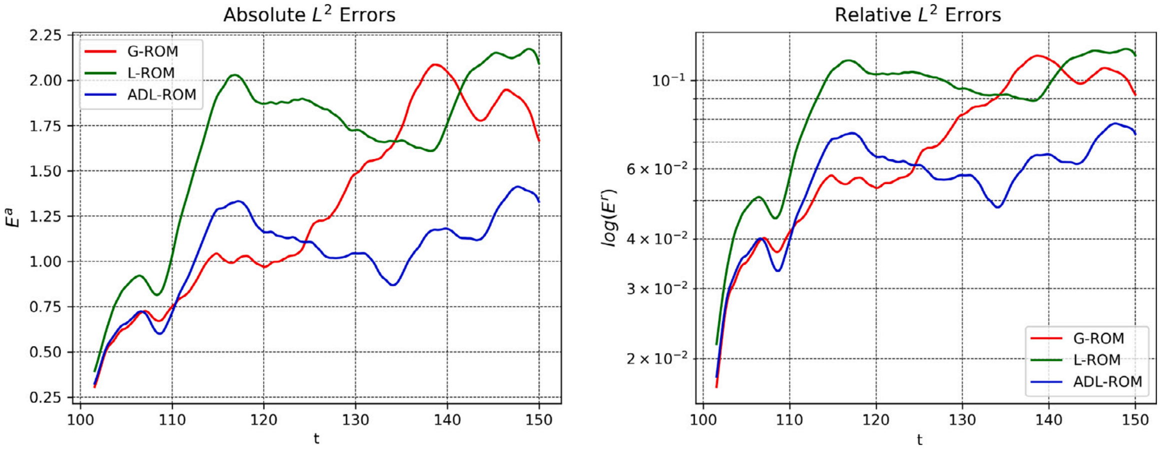


Fig. 11. Time evolution of the absolute and relative L^2 errors for G-ROM (red), L-ROM (green), and ADL-ROM (blue).

L-ROM filter radius, in our numerical investigation we used $\delta_{ROM} = 0.6$. Another important ADL-ROM parameter is the Lavrentiev regularization parameter, μ. In the image processing and inverse problem communities (see [45,46]), numerous approaches are proposed to determine the regularization parameters. In our numerical investigation, for the fixed $\delta_{ROM} = 0.6$ value, we choose the μ value that ensures that the ADL-ROM solution is as close as possible to the FOM solution. This approach yields the value $\mu = 0.06$.

Criteria. We test the ROMs accuracy by using the velocity field L^2 absolute and relative errors, and the kinetic energy. The errors are defined as

$$E^a_{\boldsymbol{u}}(t) \doteq \left\|\boldsymbol{u}(t) - \boldsymbol{u}_r(t)\right\|_{L^2(\Omega)} \quad \text{and} \quad E^r_{\boldsymbol{u}}(t) \doteq \frac{E^a_{\boldsymbol{u}}(t)}{\|\boldsymbol{u}(t)\|_{L^2(\Omega)}}, \tag{36}$$

while the kinetic energy is defined as

$$K_{\boldsymbol{u}} \doteq \frac{1}{2}\|\boldsymbol{u}(t)\|^2_{L^2(\Omega)}. \tag{37}$$

To compare the ROMs' performance, we also use the following error metric

$$\text{time-average } \ L^2 \ \text{ norm}: \quad \mathcal{E}^{a/r}(L^2) \doteq \frac{1}{M}\sum_{j=1}^{M} E^{a/r}_{\boldsymbol{u}}(t_j), \tag{38}$$

where the a and r superscripts refer to the absolute and relative errors, respectively.

Numerical results. In Table 3, we list the average absolute and relative L^2 errors (38) for the G-ROM, L-ROM, and ADL-ROM. These results show that both the G-ROM and the ADL-ROM are more accurate than the L-ROM, with the G-ROM being about 30% more accurate than the L-ROM, and the ADL-ROM about 60% more accurate than the L-ROM. Furthermore, the ADL-ROM is consistently more accurate than the G-ROM with a difference of about 30%. Also the temporal trend of the absolute and relative L^2 errors (36), displayed in Fig. 11, confirms this conclusion. It shows that ADL-ROM always performs better than L-ROM and starts to perform better than G-ROM after a while. On the other hand, G-ROM performs better than L-ROM except for the time interval $[134, 142]$. In Fig. 12, we plot the time evolution of the kinetic energy (37) of the FOM projection, G-ROM, L-ROM, and ADL-ROM. These plots support the conclusions in Table 3 and in Fig. 11. Specifically, the G-ROM and L-ROM results are relatively inaccurate, while the ADL-ROM results are significantly more accurate.

We show representative solutions of the velocity field for $T_f = 150$ in Fig. 13. It is clear that the L-ROM is not able to reconstruct the solution provided by the FOM, while the G-ROM leads to more accurate results and the ADL-ROM leads to the best results. As additional proof, we exhibit the plot of pointwise error functions, i.e., the pointwise difference between the FOM and the

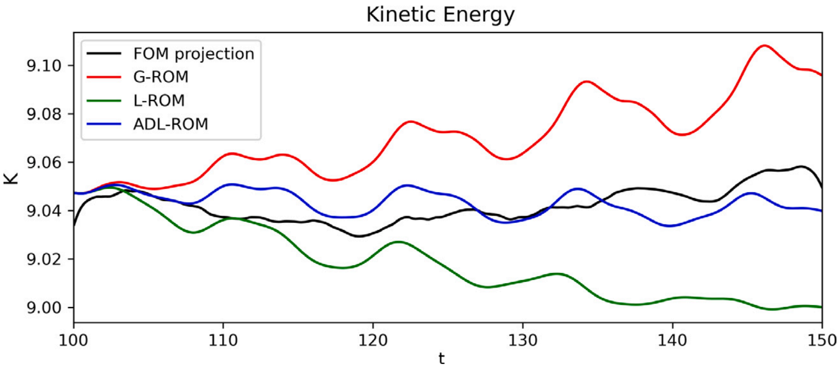


Fig. 12. Time evolution of the kinetic energy for FOM projection (black), G-ROM (red), L-ROM (green), and ADL-ROM (blue).

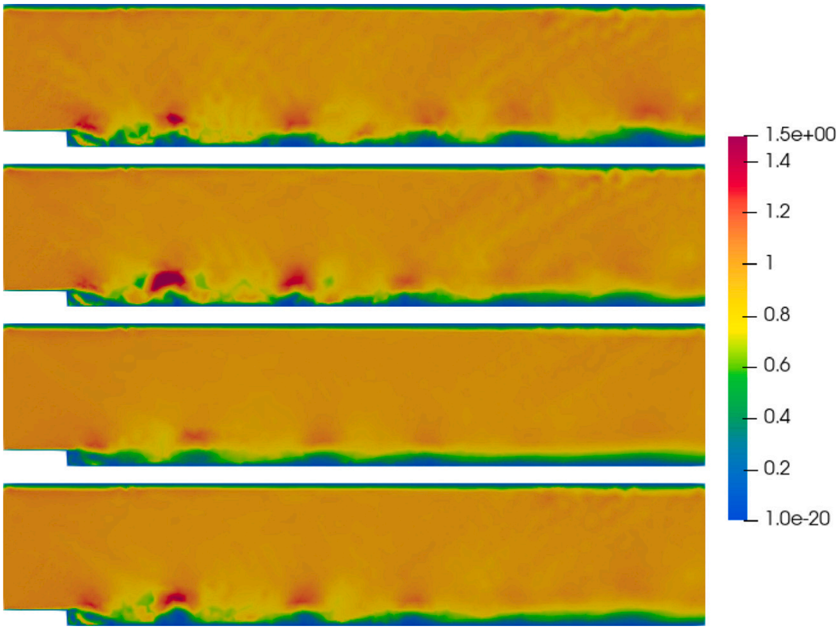


Fig. 13. At $T_f = 150$, 1st: FOM solution, 2nd: G-ROM solution, 3rd: L-ROM solution, and 4th: ADL-ROM solution.

ROMs solutions in Fig. 14. The previously observed difference in the accuracy of the ROMs solutions is even clearer in this plot by considering the maximum pointwise value in each of the three plots.

The errors listed in Table 3 and all the plots show that both the G-ROM and the ADL-ROM are more accurate than the L-ROM. Furthermore, the ADL-ROM is more accurate than the G-ROM.

Computational performance. The FE solution has 20774 degrees of freedom. The CPU time required by the 3000 time steps of the FE simulation is 9900 s, and the CPU time required by the last 1000 final time steps is 3000 s. The final 1000 time steps of the ROM solutions plotted in Fig. 13 require 2.20 s (G-ROM), 2.35 s (L-ROM), and 2.47 s (ADL-ROM). The mild increase in CPU time from G-ROM to L-ROM is due to line 8 in Algorithm 1, which is required only by L-ROM. Similarly, the mild increase in CPU time from L-ROM to ADL-ROM is due to line 9 in Algorithm 2, which is required only by ADL-ROM. We emphasize, however, that all the ROMs yield a significant speedup: 1363.63 (G-ROM), 1276.59 (L-ROM) and 1214.58 (ADL-ROM).

6. Conclusions and future work

L-ROM is a popular ROM stabilization for convection-dominated flows [21–25,27,29,30]. One of L-ROM's drawbacks is that its accuracy highly depends on δ, the radius of the spatial filter used to smooth out the spurious numerical oscillations in the standard G-ROM. Indeed, L-ROM numerical investigations have shown that small variations around the optimal δ values (i.e., filter radius values that yield the most accurate L-ROM results) can result in inaccurate predictions. For example, an aggressive δ choice, such

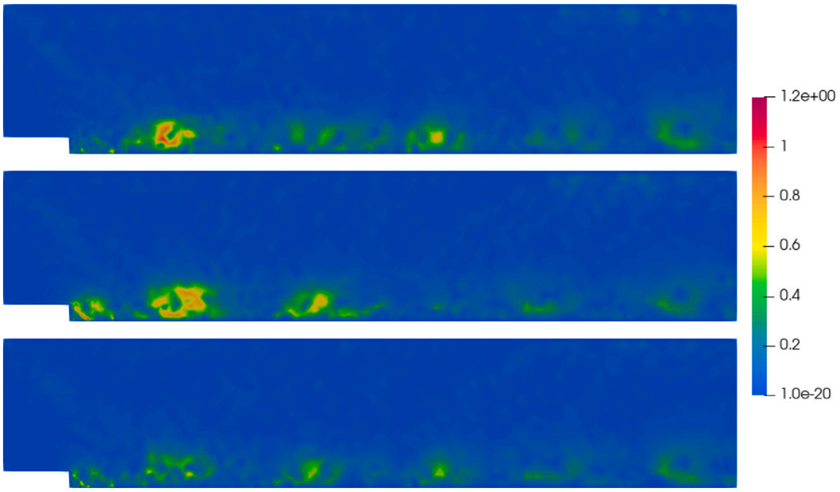


Fig. 14. At $T_f = 150$, 1st: pointwise error between FOM and G-ROM solutions, 2nd: pointwise error between FOM and L-ROM solutions, and 3rd: pointwise error between FOM and ADL-ROM solutions.

as using a large δ value intended to alleviate as much as possible the G-ROM oscillations, generally yields inaccurate L-ROM results since in that case L-ROM is overdiffusive.

To alleviate L-ROM's overdiffusive character in these settings, we propose a new regularized ROM, the ADL-ROM, which leverages approximate deconvolution [45,53] to increase the L-ROM's accuracy without compromising its numerical stability. ADL-ROM's main novelty is the replacement of L-ROM's nonlinear term $\overline{\boldsymbol{u}} \cdot \nabla \boldsymbol{u}$ with $D(\overline{\boldsymbol{u}}) \cdot \nabla \boldsymbol{u}$, where $D(\overline{\boldsymbol{u}})$ is the approximate deconvolution of the filtered ROM velocity, $\overline{\boldsymbol{u}}$. Another novel contribution of the ADL-ROM is the use of two new types of approximate deconvolution ROM operators: the van Cittert AD and the Tikhonov AD.

To assess the new ADL-ROM, we compare it with the standard L-ROM and G-ROM in the numerical investigations of two test problems: the Burgers equation with a small diffusion coefficient and the convection-dominated flow past a backward-facing step. Our numerical investigation yielded the following conclusions: First, for a large filter radius, ADL-ROM yielded more accurate results than the standard L-ROM. Specifically, ADL-ROM added a limited amount of numerical diffusion, just enough to stabilize the ROM simulation. L-ROM, on the other hand, was overdiffusive, yielding inaccurate results. The second conclusion yielded by our numerical investigation was that ADL-ROM was less sensitive with respect to parameter variations than L-ROM. Specifically, finding the optimal ADL-ROM parameters was significantly easier than finding the optimal L-ROM parameters. The third conclusion of our numerical investigation was that, for carefully chosen parameters, the three ROM approximate deconvolution strategies investigated (i.e., Lavrentiev AD, Tikhonov AD, and van Cittert AD) yielded similar results.

This first application of approximate deconvolution to construct novel regularized ROMs such as the new ADL-ROM has yielded encouraging results. There are, however, several research directions that could be further investigated. For example, the novel ADL-ROM can be assessed in the numerical simulation of more challenging convection-dominated flows, such as the turbulent channel flow [66]. One could also further investigate the role of the AD strategy in the ADL-ROM construction, both in terms of accuracy and parameter sensitivity. In particular, a thorough investigation of the accuracy of the new AD ROM strategies (Tikhonov and van Cittert) with ROM-DF and other filters should be performed. Finally, providing mathematical support (such as numerical analysis and parameter scalings) for the novel ADL-ROM is an open problem.

CRediT authorship contribution statement

Anna Sanfilippo: Conceptualization, Methodology, Software, Validation, Writing – original draft, Visualization. **Ian Moore:** Conceptualization, Methodology, Software, Validation, Writing – original draft, Visualization. **Francesco Ballarin:** Conceptualization, Methodology, Software, Validation, Writing – review & editing, Supervision, Funding acquisition. **Traian Iliescu:** Conceptualization, Methodology, Validation, Writing – review & editing, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

We acknowledge the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie Actions, grant agreement 872442 (ARIA). FB acknowledges the INdAM-GNCS project "Metodi numerici per lo studio di strutture geometriche parametriche complesse" (CUP E53C22001930001), the MIUR PRIN 2022 PNRR project "ROMEU: Reduced Order Models for Environmental and Urban flows", and the project "Reduced order modeling for numerical simulation of partial differential equations" funded by Università Cattolica del Sacro Cuore, Italy. TI acknowledges support through National Science Foundation, United States grants DMS-2012253 and CDS&E-MSS-1953113.

References

[1] F. Ballarin, A. Manzoni, A. Quarteroni, G. Rozza, Supremizer stabilization of POD–Galerkin approximation of parametrized steady incompressible Navier–Stokes equations, Internat. J. Numer. Methods Engrg. 102 (2015) 1136–1161.

[2] S.L. Brunton, J.N. Kutz, Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control, Cambridge University Press, 2019.

[3] J.S. Hesthaven, G. Rozza, B. Stamm, Certified Reduced Basis Methods for Parametrized Partial Differential Equations, in: SpringerBriefs in Mathematics, Springer, 2015.

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

[5] A. Quarteroni, A. Manzoni, F. Negri, Reduced Basis Methods for Partial Differential Equations: An Introduction, Vol. 92, Springer, 2015.

[6] S. Ali, F. Ballarin, G. Rozza, Stabilized reduced basis methods for parametrized steady Stokes and Navier-Stokes equations, Comput. Math. Appl. 80 (11) (2020) 2399–2416.

[7] S. Ali, F. Ballarin, G. Rozza, A reduced basis stabilization for the unsteady Stokes and Navier-Stokes equations, Adv. Comput. Sci. Eng. 1 (2) (2023) 180–201.

[8] M. Balajewicz, E.H. Dowell, Stabilization of projection-based reduced order models of the Navier–Stokes, Nonlinear Dynam. 70 (2012) 1619–1632.

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

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

[11] K. Carlberg, C. Bou-Mosleh, C. Farhat, Efficient non-linear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximations, Internat. J. Numer. Methods Engrg. 86 (2) (2011) 155–181.

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

[13] S. Grimberg, C. Farhat, N. Youkilis, On the stability of projection-based model order reduction for convection-dominated laminar and turbulent flows, J. Comput. Phys. 419 (2020) 109681.

[14] C. Huang, C.R. Wentland, K. Duraisamy, C. Merkle, Model reduction for multi-scale transport problems using model-form preserving least-squares projections with variable transformation, J. Comput. Phys. 448 (2022) 110742.

[15] I. Kalashnikova, M.F. Barone, On the stability and convergence of a Galerkin reduced order model (ROM) of compressible flow with solid wall and far-field boundary treatment, Internat. J. Numer. Methods Engrg. 83 (10) (2010) 1345–1375.

[16] P. Pacciarini, G. Rozza, Stabilized reduced basis method for parametrized advection–diffusion PDEs, Comput. Methods Appl. Mech. Engrg. 274 (2014) 1–18.

[17] E.J. Parish, C. Wentland, K. Duraisamy, The adjoint Petrov–Galerkin method for non-linear model reduction, Comput. Methods Appl. Mech. Engrg. 365 (2020) 112991.

[18] E.J. Parish, M. Yano, I. Tezaur, T. Iliescu, Residual-based stabilized reduced-order models of the transient convection-diffusion-reaction equation obtained through discrete and continuous projection, 2023, arXiv preprint, http://arxiv.org/abs/2302.09355.

[19] D. Torlo, F. Ballarin, G. Rozza, Stabilized weighted reduced basis methods for parametrized advection dominated problems with random inputs, SIAM/ASA J. Uncertain. Quantif. 6 (4) (2018) 1475–1502.

[20] F. Zoccolan, M. Strazzullo, G. Rozza, A streamline upwind Petrov-Galerkin reduced order method for advection-dominated partial differential equations under optimal control, 2023, arXiv preprint, http://arxiv.org/abs/arXiv:2301.01973.

[21] M. Girfoglio, A. Quaini, G. Rozza, A finite volume approximation of the Navier-Stokes equations with nonlinear filtering stabilization, Comput. & Fluids 187 (2019) 27–45.

[22] M. Girfoglio, A. Quaini, G. Rozza, A POD-Galerkin reduced order model for a LES filtering approach, J. Comput. Phys. 436 (2021) 110260.

[23] M. Gunzburger, T. Iliescu, M. Schneier, A leray regularized ensemble-proper orthogonal decomposition method for parameterized convection-dominated flows, IMA J. Numer. Anal. 40 (2) (2020) 886–913.

[24] T. Iliescu, H. Liu, X. Xie, Regularized reduced order models for a stochastic Burgers equation, Int. J. Numer. Anal. Model. 15 (4–5) (2018) 594–607.

[25] K. Kaneko, P.-H. Tsai, P. Fischer, Towards model order reduction for fluid-thermal analysis, Nucl. Eng. Des. 370 (2020) 110866.

[26] E. Rezaian, K. Duraisamy, Predictive modeling of complex flows using regularized conditionally parameterized graph neural networks, in: AIAA SCITECH 2023 Forum, 2023, p. 1284.

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

[28] P.H. Tsai, P. Fischer, Parametric model-order-reduction development for unsteady convection, Front. Phys. (2022) 711.

[29] D. Wells, Z. Wang, X. Xie, T. Iliescu, An evolve-then-filter regularized reduced order model for convection-dominated flows, Internat. J. Numer. Methods Fluids 84 (2017) 598—615.

[30] X. Xie, D. Wells, Z. Wang, T. Iliescu, Numerical analysis of the leray reduced order model, J. Comput. Appl. Math. 328 (2018) 12–29.

[31] M. Gunzburger, T. Iliescu, M. Mohebujjaman, M. Schneier, An evolve-filter-relax stabilized reduced order stochastic collocation method for the time-dependent Navier-Stokes equations, SIAM / ASA J. Uncertain. Quantif. 7 (4) (2019) 1162–1184.

[32] J. Leray, Sur le mouvement d'un fluide visqueux emplissant l'espace, Acta Math. 63 (1934) 193–248.

[33] B.J. Geurts, D.D. Holm, Regularization modeling for large-eddy simulation, Phys. Fluids 15 (1) (2003) L13–L16.

[34] M. Girfoglio, A. Quaini, G. Rozza, Pressure stabilization strategies for a LES filtering reduced order model, Fluids 6 (9) (2021).

[35] M. Girfoglio, A. Quaini, G. Rozza, A hybrid projection/data-driven reduced order model for the Navier-Stokes equations with nonlinear filtering stabilization, J. Comput. Phys. 486 (2023) 112127.

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

[37] M. Girfoglio, A. Quaini, G. Rozza, A linear filter regularization for POD-based reduced-order models of the quasi-geostrophic equations, Comptes Rendus. Mécanique (2023).

[38] M. Girfoglio, A. Quaini, G. Rozza, A novel Large Eddy Simulation model for the Quasi-Geostrophic equations in a finite volume setting, J. Comput. Appl. Math. 418 (2023) 114656.

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

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

[41] G. Rozza, G. Stabile, F. Ballarin, Advanced Reduced Order Methods and Applications in Computational Fluid Dynamics, in: Computational science and engineering, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2022.

[42] K. Kean, M. Schneier, Error analysis of supremizer pressure recovery for POD based reduced-order models of the time-dependent Navier–Stokes equations, SIAM J. Numer. Anal. 58 (4) (2020) 2235–2264.

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

[44] G. Stabile, G. Rozza, Finite volume POD-Galerkin stabilised reduced order methods for the parametrised incompressible Navier-Stokes equations, Comput. & Fluids 173 (2018) 273–284.

[45] M. Bertero, P. Boccacci, Introduction to Inverse Problems in Imaging, Institute of Physics Publishing, Bristol, 1998, p. xii+351.

[46] P.C. Hansen, Discrete Inverse Problems: Insight and Algorithms, Vol. 7, Society for Industrial and Applied Mathematics, 2010.

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

[48] X. Xie, D. Wells, Z. Wang, T. Iliescu, Approximate deconvolution reduced order modeling, Comput. Methods Appl. Mech. Engrg. 313 (2017) 512–534.

[49] L. Cordier, B. Abou El Majd, J. Favier, Calibration of POD reduced-order models using Tikhonov regularization, Internat. J. Numer. Methods Fluids 63 (2) (2010) 269–296.

[50] Y. Wang, I.M. Navon, X. Wang, Y. Cheng, 2D Burgers equation with large Reynolds number using POD/DEIM and calibration, Internat. J. Numer. Methods Fluids 82 (12) (2016) 909–931.

[51] J. Weller, E. Lombardi, A. Iollo, Robust model identification of actuated vortex wakes, Physica D 238 (4) (2009) 416–427.

[52] X. Xie, Approximate Deconvolution Reduced Order Modeling (Master's Thesis), Virginia Tech, 2015, http://theses.lib.vt.edu/theses/available/etd-11232015-161835/unrestricted/Xie_X_T_2015.pdf.

[53] W.J. Layton, L.G. Rebholz, Approximate Deconvolution Models of Turbulence: Analysis, Phenomenology and Numerical Analysis, Vol. 2042, Springer Berlin Heidelberg, 2012.

[54] C. Foias, D.D. Holm, E.S. Titi, The Navier–Stokes-alpha model of fluid turbulence, Physica D 152 (2001) 505–519.

[55] D. Wells, Stabilization of POD-ROMs (Ph.D. thesis), Virginia Tech, 2015, http://vtechworks.lib.vt.edu/bitstream/handle/10919/52960/Wells_DR_D_2015.pdf?sequence=1&isAllowed=y.

[56] W. Layton, C.C. Manica, M. Neda, L.G. Rebholz, Numerical analysis and computational testing of a high accuracy leray-deconvolution model of turbulence, Num. Meth. P.D.E.s 24 (2) (2008) 555–582.

[57] W.J. Layton, R. Lewandowski, A high accuracy leray-deconvolution model of turbulence and its limiting behavior, Anal. Appl. 6 (01) (2008) 23–49.

[58] M. Ahmed, O. San, Stabilized principal interval decomposition method for model reduction of nonlinear convective systems with moving shocks, Comput. Appl. Math. 37 (5) (2018) 6870–6902.

[59] B. Koc, T. Chacón Rebollo, S. Rubino, Uniform bounds with difference quotients for proper orthogonal decomposition reduced order models of the Burgers equation, J. Sci. Comput. 95 (2) (2023) 43.

[60] K. Kunisch, S. Volkwein, Control of the Burgers equation by a reduced-order approach using proper orthogonal decomposition, J. Optim. Theory Appl. 102 (2) (1999) 345–371.

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

[62] I. Moore, Filter Based Stabilization Methods for Reduced Order Models of Convection-Dominated Systems (Master's thesis), Virginia Tech, 2023.

[63] J. Baiges, R. Codina, S. Idelsohn, Reduced-order subscales for POD models, Comput. Methods Appl. Mech. Engrg. 291 (2015) 173–196.

[64] C. Mou, B. Koc, O. San, L.G. Rebholz, T. Iliescu, Data-driven variational multiscale reduced order models, Comput. Methods Appl. Mech. Engrg. 373 (2021) 113470.

[65] M. Strazzullo, M. Girfoglio, F. Ballarin, T. Iliescu, G. Rozza, Consistency of the full and reduced order models for evolve-filter-relax regularization of convection-dominated, marginally-resolved flows, Internat. J. Numer. Methods Engrg. 123 (14) (2022) 3148–3178.

[66] C. Mou, E. Merzari, O. San, T. Iliescu, An energy-based lengthscale for reduced order models of turbulent flows, Nucl. Eng. Des. 412 (2023) 112454.

[67] R. Reyes, R. Codina, Projection-based reduced order models for flow problems: A variational multiscale approach, Comput. Methods Appl. Mech. Engrg. 363 (2020) 112844.