Nuclear Engineering and Design 412 (2023) 112454

Contents lists available at ScienceDirect

Nuclear Engineering and Design

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

An energy-based lengthscale for reduced order models of turbulent flows

Changhong Mou [a], Elia Merzari [b], Omer San [c], Traian Iliescu [d,*]

[a] *Department of Mathematics, University of Wisconsin-Madison, Madison, WI 53706, USA*
[b] *Department of Nuclear Engineering, The Pennsylvania State University, University Park, PA 16802, USA*
[c] *School of Mechanical and Aerospace Engineering, Oklahoma State University, Stillwater, OK 74078, USA*
[d] *Department of Mathematics, Virginia Tech, Blacksburg, VA 24061, USA*

ARTICLE INFO

Keywords:
Reduced order model
Lengthscale
Mixing-length
Evolve-filter-relax
Turbulent channel flow

ABSTRACT

In this paper, we propose a novel reduced order model (ROM) lengthscale that is constructed by using energy distribution arguments. The new energy-based ROM lengthscale is fundamentally different from the current ROM lengthscales, which are built by using dimensional arguments. To assess the novel, energy-based ROM lengthscale, we compare it with a standard, dimensionality-based ROM lengthscale in two fundamentally different types of models: (i) the mixing-length ROM (ML-ROM), which is a ROM closure model; and (ii) the evolve-filter-relax ROM (EFR-ROM), which is a regularized ROM. We test the four combinations (i.e., ML-ROM and EFR-ROM equipped with the energy-based and dimensionality-based lengthscales) in the numerical simulation of the turbulent channel flow at $Re_\tau = 395$. The numerical investigation yields the following conclusions: (i) The new energy-based ROM lengthscale is significantly (almost two orders of magnitude) larger than the standard dimensionality-based ROM lengthscale. As a result, the energy-based lengthscale yields more stable ML-ROMs and EFR-ROMs than the dimensionality-based lengthscale. (ii) The energy-based lengthscale displays the correct asymptotic behavior with respect to the ROM dimension, whereas the dimensionality-based lengthscale does not. The energy-based lengthscale is intrinsically adaptive with respect to the ROM dimension, which is important in realistic settings where using the full order model data to determine an optimal ROM lengthscale may not be possible. (iii) The energy-based lengthscale yields ML-ROMs and (when significant filtering is effected) EFR-ROMs whose parameters are less sensitive (i.e., more robust) than the parameters of the ML-ROMs and EFR-ROMs based on the dimensionality-based lengthscale. The novel energy-based lengthscale could enable the development of better scale-aware ROM strategies for flow-specific applications and is expected to have long term applications in nuclear reactor thermal-hydraulics.

1. Introduction

Reduced order models (ROMs) are models whose dimensions are orders of magnitude lower than the dimensions of full order models (FOMs), i.e., computational models constructed by using classical numerical discretizations (e.g., finite element or finite volume methods). Because of their relatively low-dimensionality, ROMs can be used as efficient alternatives to FOMs in computationally intensive applications, e.g., flow control, shape optimization, and uncertainty quantification.

In recent years, ROMs have received a great deal of interest in nuclear engineering applications to generate models that can account for the fundamental physics of key phenomena while maintaining a low computational cost (Merzari et al., 2017; Fick et al., 2018). This is an important goal in nuclear engineering due to the size and complexity of nuclear systems, which make the detailed three-dimensional simulation of full systems impractical for design exploration purposes. ROMs can be used to predict a transient turbulent flow field in seconds to minutes on a single CPU, as opposed to turbulence-resolving methods such as large eddy simulation (LES) or direct numerical simulation (DNS), which require large computational resources (hours on clusters or supercomputers). Alternative lower-cost approaches in nuclear engineering include coarser LES, hybrid Reynolds-averaged Navier–Stokes (RANS)-LES, or even coarse RANS (Roelofs et al., 2012). Compared to these approaches, ROMs have the advantage that, if properly constructed, they can offer similar accuracy to LES (Fick et al., 2018), but they require expensive training data and they are not easily generalizable.

ROMs have received attention for a variety of specific nuclear engineering applications. Examples are efforts to develop modal and machine-learning based ROMs for thermal stratification (Liu et al., 2022), which is recognized as critical for the licensing of liquid metal

---

* Corresponding author.
E-mail address: iliescu@vt.edu (T. Iliescu).
URL: https://sites.google.com/view/iliescu/ (T. Iliescu).

https://doi.org/10.1016/j.nucengdes.2023.112454
Received 8 November 2022; Received in revised form 25 April 2023; Accepted 23 June 2023

reactors. Another important application is the use of ROMs to develop closure models (Fiore et al., 2022) as a complement to existing DNS, LES, or experimental data. In fact, ROMs allow a broader exploration of the parameter space than DNS or LES alone, due to the high computational cost of the latter.

ROMs have also found applications in the modeling of parameterized coupled thermal-hydraulics and reactor physics problems (Vergari et al., 2020, 2021). Similarly, GeN-ROM, which was recently introduced in German et al. (2022), is a data-driven framework for reducing multiphysics problems in nuclear systems using a proper orthogonal decomposition aided reduced basis strategy. GeN-ROM claims significant computational speedup for multi-query applications and has been successfully tested on a two-dimensional multiphysics model of a molten salt fast reactor (MSFR). Finally, another notable effort is the use of ROMs to develop flow acceleration for the advection–diffusion equation (Merzari et al., 2011).

The Galerkin ROM (G-ROM) framework has been often used in the numerical simulation of fluid flows (Holmes et al., 1996; Ahmed et al., 2021). The G-ROM is constructed as follows: First, in an offline phase, the FOM is used to produce snapshots, which are then utilized to construct a low-dimensional (i.e., $r \ll N$) ROM basis $\{\varphi_1, \ldots, \varphi_r\}$, where r is the ROM dimension and N is the FOM dimension. There are several strategies for constructing the ROM basis, e.g., the proper orthogonal decomposition (POD) (Holmes et al., 1996; Volkwein, 2013) and the reduced basis method (Hesthaven et al., 2015; Quarteroni et al., 2015). In this paper, we exclusively use the POD to construct the ROM basis. Next, the ROM basis is used together with a Galerkin projection to build the G-ROM, which can be written as follows:

$$\dot{\boldsymbol{a}} = \mathbf{F}(\boldsymbol{a}), \tag{1}$$

where $\boldsymbol{a}$ is the vector of coefficients in the ROM approximation $\sum_{i=1}^{r} a_i(t)\boldsymbol{\varphi}_i(\boldsymbol{x})$ of the variable of interest, $\dot{\boldsymbol{a}}$ denotes the vector of time derivatives of $\boldsymbol{a}$, and the vector $\mathbf{F}$ comprises the ROM operators that are preassembled in the offline phase. In the online phase, the G-ROM (1) is employed for new parameter values and/or time intervals i.e., parameter values and time intervals that are different from those used in the training stage. Thus, the G-ROM can be utilized in practical applications to efficiently explore new parameter regimes without having to run the costly FOM.

We emphasize that, when FOMs (i.e., classical numerical methods, e.g., finite element, finite volume, or finite difference methods) are used for the spatial discretization of the fluid flow equations (e.g., the Navier–Stokes equations), the FOM lengthscale is generally defined as the mesh size, h, of the spatial mesh. In contrast, the following *natural question* is, to our knowledge, *still open*:

What is the ROM lengthscale, δ?

To formulate the above question mathematically, we first assume that the following FOM and ROM variables are given, as is generally the case when ROMs are applied in practical settings:

- ROM variables, e.g., the ROM dimension, r, the total number of available ROM basis functions, R, the ROM basis functions, $\boldsymbol{\varphi}_i$, and the corresponding eigenvalues, λ_i, in the eigenvalue problem solved to construct the POD basis.
- FOM variables, e.g., the FOM mesh size, h, the FOM solution, $\boldsymbol{u}^{FOM}$, and the computational domain characteristic lengthscale, L.

Given these input FOM and ROM variables, we then try to answer the above question, i.e., to express the ROM lengthscale, δ, as a function of the given FOM and ROM variables.

To motivate the need for a ROM lengthscale, we point out that the lengthscale is a fundamental notion in engineering, geophysical, and biomedical applications, where it is used to characterize the resolved spatial scales (i.e., to determine the size of the spatial scales approximated by the computational model) in flows around cars or airplanes, ocean or atmospheric flows, or blood flow in an artery, respectively.

To further motivate the need for a ROM lengthscale, we emphasize that there are ROMs that use a lengthscale in their very definition. For example, in under-resolved simulations of turbulent flows, i.e., when the number of ROM basis functions is not enough to represent the turbulent flow dynamics, it is well known that the standard G-ROM (1) often yields spurious numerical oscillations and even blowup. Thus, ROM closures and stabilizations are often used (see Ahmed et al., 2021 for a review). The ROM closures modify the standard G-ROM (1) as follows:

$$\dot{\boldsymbol{a}} = \mathbf{F}(\boldsymbol{a}) + \boldsymbol{\tau}, \tag{2}$$

where $\boldsymbol{\tau}$ is the closure term. The energy cascade, which is central in Kolmogorov's statistical theory of turbulence (Frisch, 1995; Pope, 2000), states that, in the inertial range, energy is transferred from the large scales to the small scales. Since the ROM basis functions are often ordered in decreasing order of energy content, the first ROM basis functions generally correspond to the large, energy containing scales, and the remaining ROM basis functions correspond to the small scales. Thus, the role of the ROM closure term $\boldsymbol{\tau}$ in (2) is to dissipate energy from the system. The eddy viscosity ROM closures are a popular class of ROM closures in which the closure term is dissipative. Examples in this class include the mixing-length ROM (Aubry et al., 1988; Holmes et al., 1996; Wang et al., 2012), the Smagorinsky ROM (Wang et al., 2012), the dynamic SGS ROM (Wang et al., 2012), the eddy viscosity variational multiscale ROM (Wang et al., 2012), the RANS-ROM (Lorenzi et al., 2016), and data-driven ROM closures (Hijazi et al., 2020; Mohebujjaman et al., 2019).

The ROM stabilizations are a related class of models that aim at increasing the ROM stability. In this paper, we consider ROM stabilizations that address the numerical instability in the convection-dominated regime, which are consistent with the energy cascade concept. (ROM pressure stabilizations can be found, e.g., in Ballarin et al., 2015; DeCaria et al., 2020; Stabile et al., 2017; Stabile and Rozza, 2018.) The regularized ROMs are stabilization strategies that increase the ROM stability by using ROM spatial filtering of various terms of the underlying equations. Example of regularized ROMs include the evolve-filter-relax ROM (Wells et al., 2017; Gunzburger et al., 2019; Girfoglio et al., 2021a,b; Strazzullo et al., 2022) and the Leray ROM (Wells et al., 2017; Sabetghadam and Jafarpour, 2012; Kaneko et al., 2020; Gunzburger et al., 2020; Iliescu et al., 2018; Tsai and Fischer, 2022).

We emphasize that a ROM lengthscale is needed to construct both the ROM closures and the regularized ROMs mentioned above. We also note that a ROM lengthscale could also be useful in the preprocessing strategy advocated in Aradag et al. (2011), Farcas et al. (2022) as a means to filter out the noise in the input data.

In this paper, we propose a *novel ROM lengthscale*, which is constructed by using energy distribution arguments. To assess the new energy-based ROM lengthscale, we compare it with the classical ROM lengthscale used in Aubry et al. (1988), Holmes et al. (1996), Wang et al. (2012), which is based on fundamentally different, dimensional arguments. To compare the two ROM lengthscales, we utilize them to build two different types of ROMs for under-resolved simulations: (i) the mixing-length ROM (Aubry et al., 1988; Holmes et al., 1996; Wang et al., 2012), which is a ROM closure model that augments the standard G-ROM with a correction term; and (ii) the evolve-filter-relax ROM (Wells et al., 2017), which is a regularized ROM (Reg-ROM) that leverages ROM spatial filtering to increase the ROM stability and accuracy. We test both the ROM closure and the Reg-ROM equipped with both ROM lengthscales in the numerical simulation of the turbulent channel flow at $Re_\tau = 395$. We note that a preliminary numerical investigation of the new ROM lengthscale was performed in Mou et al. (2022).

The rest of the paper is organized as follows: In Section 2, we outline the standard G-ROM, ML-ROM, and EFR-ROM. In Section 3, we define the new, energy-based lengthscale and the standard dimensionality-based lengthscale. In Section 4, we present results for our investigation of the ML-ROM and EFR-ROM equipped with the two lengthscales in the numerical simulation of the turbulent channel flow at $Re_\tau = 395$. Finally, in Section 5, we draw conclusions and outline directions of future research.

2. Reduced order models

In this section, we outline the construction of the standard G-ROM, mixing-length ROM (ML-ROM), and evolve-filter-relax ROM (EFR-ROM). As a mathematical model, we consider the incompressible Navier–Stokes equation (NSE):

$$\frac{\partial \boldsymbol{u}}{\partial t} - Re^{-1}\Delta \boldsymbol{u} + \left(\boldsymbol{u}\cdot\nabla\right)\boldsymbol{u} + \nabla p \ = \ \boldsymbol{f}, \tag{3}$$

$$\nabla\cdot\boldsymbol{u} \ = \ 0, \tag{4}$$

where $\boldsymbol{u} = [u_1, u_2, u_3]^\top$ is the velocity vector field, p the pressure field, Re the Reynolds number, and $\boldsymbol{f}$ the forcing term. The NSE are equipped with appropriate boundary and initial conditions.

2.1. Galerkin ROM (G-ROM)

To build the G-ROM, we consider the centering trajectory of the flow,

$$\boldsymbol{U}(\boldsymbol{x}) = \frac{1}{T}\int_{t_0}^{t_0+T} \boldsymbol{u}(\boldsymbol{x},t)dt, \tag{5}$$

and assume that the ROM velocity approximation can be written as follows:

$$\boldsymbol{u}_r(\boldsymbol{x},t) \ = \ \boldsymbol{U}(\boldsymbol{x}) + \sum_{j=1}^{r} a_j(t)\boldsymbol{\varphi}_j(\boldsymbol{x})\,, \tag{6}$$

where $\{\boldsymbol{\varphi}_j\}_{j=1}^r$ are the ROM basis functions and $\boldsymbol{a} = [a_1,\ldots,a_r]^\top$ are the sought ROM coefficients. In our numerical experiments, we use the POD (Holmes et al., 1996) to construct the ROM basis, but other ROM bases could be used (Brunton and Kutz, 2019; Hesthaven et al., 2015; Quarteroni et al., 2015). The next step in the G-ROM construction is to replace $\boldsymbol{u}$ with $\boldsymbol{u}_r$ in (3) and project the resulting equations onto the space spanned by the ROM basis, $\{\boldsymbol{\varphi}_j\}_{j=1}^r$. This yields the G-ROM:

$$\left(\frac{\partial \boldsymbol{u}_r}{\partial t},\boldsymbol{\varphi}_i\right) + \left((\boldsymbol{u}_r\cdot\nabla)\boldsymbol{u}_r,\boldsymbol{\varphi}_i\right) + Re^{-1}\left(\nabla\boldsymbol{u}_r,\nabla\boldsymbol{\varphi}_i\right) = \left(\boldsymbol{f},\boldsymbol{\varphi}_i\right), \quad i = 1,\ldots,r, \tag{7}$$

where $(\cdot,\cdot)$ denotes the L^2 inner product. The G-ROM can be written as the following dynamical system for the vector of time coefficients, $\boldsymbol{a}(t)$:

$$\dot{\boldsymbol{a}} = \boldsymbol{b} + A\boldsymbol{a} + \boldsymbol{a}^\top B\boldsymbol{a}\,, \tag{8}$$

where

$$\boldsymbol{b}_i \ = \ \left(\boldsymbol{\varphi}_i,\boldsymbol{f}\right) - \left(\boldsymbol{\varphi}_i,\boldsymbol{U}\cdot\nabla\boldsymbol{U}\right) - Re^{-1}\left(\nabla\boldsymbol{\varphi}_i,\nabla\boldsymbol{U}\right), \tag{9}$$

$$\boldsymbol{A}_{im} \ = \ -\left(\boldsymbol{\varphi}_i,\boldsymbol{U}\cdot\nabla\boldsymbol{\varphi}_m\right) - \left(\boldsymbol{\varphi}_i,\boldsymbol{\varphi}_m\cdot\nabla\boldsymbol{U}\right) - Re^{-1}\left(\nabla\boldsymbol{\varphi}_i,\nabla\boldsymbol{\varphi}_m\right), \tag{10}$$

$$\boldsymbol{B}_{imn} \ = \ -\left(\boldsymbol{\varphi}_i,\boldsymbol{\varphi}_m\cdot\nabla\boldsymbol{\varphi}_n\right). \tag{11}$$

2.2. Mixing-length ROM (ML-ROM)

The G-ROM (8) is computationally efficient and relatively accurate in the numerical simulation of laminar flows. However, as mentioned in Section 1, in under-resolved simulations of turbulent flows, the standard G-ROM (1) often yields spurious numerical oscillations and even blowup. The reason is that, according to the energy cascade concept and Kolmogorov's statistical theory of turbulence (Frisch, 1995; Pope, 2000), the model should include a mechanism to transfer energy from the large scales (i.e., the ROM scales) to the small scales. Thus, the G-ROM is generally equipped with a ROM closure model, which models the effect of the discarded ROM modes $\{\boldsymbol{\varphi}_{r+1},\ldots\}$ on the G-ROM dynamics:

$$\dot{\boldsymbol{a}} = \boldsymbol{b} + A\boldsymbol{a} + \boldsymbol{a}^\top B\boldsymbol{a} + \boldsymbol{\tau}\,, \tag{12}$$

where $\boldsymbol{\tau}$ is the ROM closure model. The current ROM closure models are carefully surveyed in Ahmed et al. (2021). Some of these ROM closure models are inspired from classical large eddy simulation (LES) closure modeling (Berselli et al., 2006; Sagaut, 2006). To construct these LES-ROM closure models, one needs to define a ROM lengthscale, which represents the size of the spatial scales modeled in the LES-ROM. There are only a few ROM lengthscales in current use. In Section 3, we define a novel ROM lengthscale. To assess this new ROM lengthscale, we consider one of the simplest ROM closure models, the ML-ROM (Holmes et al., 1996; Wang et al., 2012), in which the ROM closure term $\boldsymbol{\tau}$ in (12) is written as

$$\boldsymbol{\tau} = -\left(\alpha\, U_{ML}\,\delta\right) S_r\,\boldsymbol{a}\,, \tag{13}$$

where δ is one of the two ROM lengthscales defined in Section 3, U_{ML} is a characteristic velocity scale, α is a constant, and S_r is the ROM stiffness matrix with entries $(S_r)_{ij} = \left(\nabla\boldsymbol{\varphi}_i,\nabla\boldsymbol{\varphi}_j\right),\ i,j = 1,\ldots,r$. The ML-ROM model (13) is a functional closure model, which aims at increasing the ROM viscosity in order to dissipate energy and mimic the effect of the discarded modes (Couplet et al., 2003). The ML-ROM (13) was first used in Aubry et al. (1988), Holmes et al. (1996) and was further investigated in Wang et al. (2012).

2.3. Evolve-filter-relax ROM (EFR-ROM)

Regularized ROMs (Reg-ROMs) (Kaneko et al., 2020; Wells et al., 2017) represent an alternative to ROM closures (e.g., the ML-ROM outlined in Section 2.2) in under-resolved simulations of turbulent flows. Instead of adding a closure term, $\boldsymbol{\tau}$, as in ROM closure modeling (see (12)), Reg-ROMs are constructed by using ROM spatial filtering of various terms in the NSE to increase the ROM numerical stability. Although regularized models have been used for decades in classical CFD (Fischer and Mullen, 2001; Layton and Rebholz, 2012; Mullen and Fischer, 1999), Reg-ROMs have only been recently developed (Wells et al., 2017).

The evolve-filter-relax ROM (EFR-ROM) is one of the most popular Reg-ROMs. EFR-ROM is a modular ROM stabilization strategy that consists of three steps: In the first step, which is called the evolve step, the standard G-ROM is used to advance the current EFR-ROM time iteration, $\boldsymbol{a}^n$, to an intermediate approximation, $\boldsymbol{w}^{n+1}$. In the second step, which is called the filter step, the intermediate approximation, $\boldsymbol{w}^{n+1}$, is filtered with the ROM differential filter (Wells et al., 2017), which yields the filtered intermediate approximation, $\overline{\boldsymbol{w}}^{n+1}$. In the third step, which is called the relax step, the EFR-ROM approximation at the next time step, $\boldsymbol{a}^{n+1}$, is calculated as the convex combination of the intermediate approximation, $\boldsymbol{w}^{n+1}$, and the filtered intermediate approximation, $\overline{\boldsymbol{w}}^{n+1}$. The EFR-ROM is summarized in the following algorithm:

$$\begin{array}{lll} \textbf{(I)}\ \ \textit{Evolve}: & \boldsymbol{a}^n \overset{G-ROM\ (8)}{\longmapsto} \boldsymbol{w}^{n+1} \\ \textbf{(II)}\ \ \textit{Filter}: & \left(M_r + \gamma\,\delta^2\,S_r\right)\overline{\boldsymbol{w}}^{n+1} = \boldsymbol{w}^{n+1} \\ \textbf{(III)}\ \textit{Relax}: & \boldsymbol{a}^{n+1} = (1-\chi)\,\boldsymbol{w}^{n+1} + \chi\,\overline{\boldsymbol{w}}^{n+1}\,, \end{array}$$

where $\chi \in [0,1]$ is a relaxation parameter and M_r is the ROM mass matrix with entries $(M_r)_{ij} = \left(\boldsymbol{\varphi}_i,\boldsymbol{\varphi}_j\right),\ i,j = 1,\ldots,r$. In Step (II), we use the ROM differential filter (DF) with an explicit ROM lengthscale, δ, which represents the filtering radius. The DF acts as a spatial filter by eliminating the small scales (i.e., high frequencies) from the input data (Berselli et al., 2006). We note that, in Step (II), we modify the classical DF (Berselli et al., 2006) by introducing a new parameter

γ. This new parameter γ has a role similar to that of the parameter α used in the ML-ROM (13): It controls the amount of filtering used in the DF. Step (III) is a relaxation step in which the EFR-ROM velocity approximation at the new time step is defined as a convex combination of the approximations obtained in Step (I) and Step (II). The relaxation parameter χ diminishes the magnitude of the numerical diffusion (Ervin et al., 2012; Fischer and Mullen, 2001; Mullen and Fischer, 1999) and increases the accuracy (Bertagna et al., 2016; Ervin et al., 2012). Although there is no universal χ scaling, the choices $\chi \sim \Delta t$ (Ervin et al., 2012), where Δt is the time step size, and higher χ values (Bertagna et al., 2016) have been used in the literature. To our knowledge, the EFR-ROM was first used in Wells et al. (2017) without the relaxation step, and in Gunzburger et al. (2019) with the relaxation step (see also Strazzullo et al., 2022 and references therein). EFR-ROM was also investigated in Wells et al. (2017), Gunzburger et al. (2019), Girfoglio et al. (2021a,b), Strazzullo et al. (2022).

3. ROM lengthscales

In this section, we present two different ROM lengthscales: In Section 3.1, we present a standard ROM lengthscale, denoted δ_1, which is constructed by using dimensional analysis arguments. In Section 3.2, we propose a novel ROM lengthscale, denoted δ_2, which is constructed by using energy balance arguments. As explained in the introduction, both definitions aim at expressing the ROM lengthscale as a function of the following two types of input variables: (i) ROM variables (e.g., the ROM dimension, r, the total number of ROM basis functions, R, the eigenvalues, λ_i, and the ROM basis functions, $\boldsymbol{\varphi}_i$). (ii) FOM variables (e.g., the fine FOM mesh size, h, the FOM solution, $\boldsymbol{u}^{FOM}$, and the computational domain characteristic lengthscale, L). Given these input FOM and ROM variables, we then try to answer the following natural question: *What is the ROM lengthscale, δ?*

3.1. Standard dimensionality-based ROM lengthscale δ_1

In this section, we use dimensional analysis to construct the first ROM lengthscale, δ_1. To this end, we follow the approach used in Section 3.2 in Wang et al. (2012), which, in turn, is based on the pioneering ML-ROM proposed in Aubry et al. (1988) for a turbulent pipe flow.

To construct the ROM lengthscale δ_1, we first define the componentwise FOM velocity fluctuations, i.e., the unresolved component of the velocity, which is computed by using FOM data:

$$u_i'^{FOM} = \sum_{j=r+1}^{R} a_j^{FOM} \varphi_j^i, \qquad i = 1, 2, 3, \tag{14}$$

where R is the total number of ROM modes, φ_j^i are the componentwise ROM basis functions, and $a_j^{FOM} = (\boldsymbol{u}^{FOM}, \boldsymbol{\varphi}_j)$ are the ROM coefficients computed by using FOM data. Using the componentwise FOM velocity fluctuations $u_1'^{FOM}, u_2'^{FOM}$, and $u_3'^{FOM}$ in the x, y, and z directions, respectively, we build the FOM velocity fluctuation vector field $\boldsymbol{u}'^{FOM} = [u_1'^{FOM}, u_2'^{FOM}, u_3'^{FOM}]$. Since $\boldsymbol{u}'^{FOM}$ varies with time, we calculate the time averaged value of $\boldsymbol{u}'^{FOM}$, i.e.,

$$\langle \boldsymbol{u}'^{FOM} \rangle_t(\boldsymbol{x}) = \frac{1}{M} \sum_{k=1}^{M} \boldsymbol{u}'^{FOM}(\boldsymbol{x}, t_k) = \frac{1}{M} \sum_{k=1}^{M} \sum_{l=r+1}^{R} \left(\boldsymbol{u}^{FOM}(\cdot, t_k), \boldsymbol{\varphi}_l(\cdot) \right) \boldsymbol{\varphi}_l(\boldsymbol{x}), \tag{15}$$

where M is the number of snapshots.

To construct the ROM lengthscale δ_1, we adapt equation (22) used in Wang et al. (2012) to our computational setting (i.e., the turbulent channel flow in Section 4):

$$\delta_1 := \left(\frac{\int_0^{L_1} \int_0^{L_2} \int_0^{L_3} \sum_{i=1}^{3} u_i'^{FOM} u_i'^{FOM} \, dx_1 \, dx_2 \, dx_3}{\int_0^{L_1} \int_0^{L_2} \int_0^{L_3} \sum_{i=1}^{3} \sum_{j=1}^{3} \frac{\partial u_i'^{FOM}}{\partial x_j} \frac{\partial u_i'^{FOM}}{\partial x_j} \, dx_1 \, dx_2 \, dx_3} \right)^{1/2}, \tag{16}$$

where L_1, L_2, and L_3 are the streamwise, wall-normal, and spanwise dimensions of the computational domain of the turbulent channel flow test problem, respectively.

Note that a quick dimensional analysis shows that the quantity defined in (16) has the units of a lengthscale: $[\delta_1] = \left((\mathrm{m/s})(\mathrm{m/s})\,\mathrm{m}^3 \right) / \left((1/\mathrm{s})(1/\mathrm{s})\,\mathrm{m}^3 \right)^{1/2} = \mathrm{m}$. We also note that the ROM lengthscale, δ_1, defined in (16), depends on the FOM velocity fluctuation vector field, $\boldsymbol{u}'^{FOM}$.

An alternative lengthscale was defined in equation (23) in Wang et al. (2012). Since this alternative lengthscale was not used in the numerical investigation in Wang et al. (2012) (because it was harder to implement), we do not consider it in this study.

3.2. Novel energy-based ROM lengthscale δ_2

In this section, we use energy balancing arguments and propose a new ROM lengthscale, δ_2. Noticing that the ROM truncation level, r, has the role of dividing the kinetic energy of the system, we can require that the new ROM lengthscale, δ_2, do the same. Specifically, we require that the ratio of kinetic energy contained in the first r ROM modes, $\sum_{i=1}^{r} \lambda_i$, to the kinetic energy contained in the total number of ROM modes, $\sum_{i=1}^{R} \lambda_i$, be equal to the ratio of the kinetic energy that can be represented on an imaginary mesh of size δ_2, $KE(\delta_2)$, to the kinetic energy that can be represented on the FOM mesh, $KE(h)$:

$$\frac{\sum_{i=1}^{r} \lambda_i}{\sum_{i=1}^{R} \lambda_i} = \frac{KE(\delta_2)}{KE(h)} . \tag{17}$$

Remark 3.1. We emphasize that the mesh of size δ_2 is not used in the actual ROM construction. Instead, this imaginary mesh is used to highlight the fundamental difference between the physical lengthscales of the space generated by the first r ROM modes, $\boldsymbol{X}^r = \mathrm{span}\{\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_r\}$, and the physical lengthscales of the space generated by all R ROM modes, $\boldsymbol{X}^R = \mathrm{span}\{\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_r, \boldsymbol{\varphi}_{r+1}, \ldots, \boldsymbol{\varphi}_R\}$. Indeed, since the FOM data was used to build all the ROM modes, all the lengthscales of the space $\boldsymbol{X}^R$ can be represented on the FOM mesh. However, the imaginary mesh of size δ_2 can represent only the lengthscales of the space $\boldsymbol{X}^r$; it cannot be expected to represent the physical lengthscales of the space spanned by the higher index ROM modes $\{\boldsymbol{\varphi}_{r+1}, \ldots, \boldsymbol{\varphi}_R\}$, which are generally associated with the small scales.

To compute the ratio $\frac{KE(\delta_2)}{KE(h)}$ in (17), we transfer the problem to the usual Fourier space. To this end, we first notice that δ_2 defines a cutoff wavenumber:

$$k_{\delta_2} := \frac{2\pi}{\delta_2} . \tag{18}$$

We then notice that the kinetic energy in the system can be written in terms of the energy spectrum, $E(\cdot)$:

$$KE(k) = \int_{k_0}^{k} E(k') \, dk' , \tag{19}$$

where $k_0 = \frac{2\pi}{L}$ is the Fourier wavenumber that corresponds to the computational domain characteristic lengthscale, L. In the case of isotropic, homogeneous turbulence, we have the usual energy spectrum given by Kolmogorov's theory (Sagaut, 2006; Pope, 2000)

$$E(k) \sim C \, \varepsilon^{2/3} \, k^{-5/3} . \tag{20}$$

Thus, the condition imposed in (17) can be written as

$$\frac{\int_{k_0}^{k_{\delta_2}} E(k) \, dk}{\int_{k_0}^{k_h} E(k) \, dk} = \frac{\sum_{i=1}^{r} \lambda_i}{\sum_{i=1}^{R} \lambda_i} \overset{\text{notation}}{=} \Lambda , \tag{21}$$

where $k_h = \frac{2\pi}{h}$ is the highest Fourier wavenumber that can be resolved on the given FOM mesh size, h. The LHS of (21) can be evaluated by using (20):

$$\int_{k_0}^{k_{\delta_2}} E(k)\, dk = C\, \varepsilon^{2/3} \int_{k_0}^{k_{\delta_2}} k^{-5/3}\, dk = C\, \varepsilon^{2/3}\, \frac{k_{\delta_2}^{-2/3} - k_0^{-2/3}}{-2/3}\,, \tag{22}$$

and, similarly,

$$\int_{k_0}^{k_h} E(k)\, dk = C\, \varepsilon^{2/3} \int_{k_0}^{k_h} k^{-5/3}\, dk = C\, \varepsilon^{2/3}\, \frac{k_h^{-2/3} - k_0^{-2/3}}{-2/3}. \tag{23}$$

Plugging (22) and (23) back into (21), simplifying, and rearranging, we obtain

$$k_{\delta_2}^{-2/3} = \Lambda\, k_h^{-2/3} + (1-\Lambda)\, k_0^{-2/3}\,. \tag{24}$$

Since $1 \leq r \leq R$, Λ satisfies the inequality $0 < \Lambda \leq 1$. Thus, (24) implies that $k_{\delta_2}^{-2/3}$ is a convex combination of $k_h^{-2/3}$ and $k_0^{-2/3}$. Furthermore, as expected, k_{δ_2} satisfies the following asymptotic relations:

$$\Big[(r \to R) \Longrightarrow (k_{\delta_2} \longrightarrow k_h) \Big] \quad \text{and} \quad \Big[(r \to 0) \Longrightarrow (k_{\delta_2} \longrightarrow k_0) \Big]. \tag{25}$$

Using (24) together with (18), gives us a formula for δ_2:

$$\begin{aligned} \delta_2 = \frac{2\,\pi}{k_{\delta_2}} &= 2\,\pi \left[\Lambda \left(\frac{2\,\pi}{h} \right)^{-2/3} + (1-\Lambda) \left(\frac{2\,\pi}{L} \right)^{-2/3} \right]^{3/2} \\ &= \left[\Lambda\, h^{2/3} + (1-\Lambda)\, L^{2/3} \right]^{3/2}\,. \end{aligned} \tag{26}$$

The new ROM lengthscale, δ_2, defined in (26), depends on the FOM mesh size, h, the ROM dimension, r, the total number of ROM basis functions, R, the eigenvalues, λ_i, and the computational domain characteristic lengthscale, L.

Remark 3.2 (*Asymptotic Behavior*)**.** We note that the novel ROM lengthscale, δ_2, satisfies the following two natural asymptotic relations:

- As r approaches R, δ_2 approaches h.
  This simply says that, as expected, when the ROM dimension, r, approaches the maximal ROM dimension, R, the ROM lengthscale, δ_2, approaches the minimal FOM lengthscale, h (i.e., the spatial mesh size).
- As r approaches 1, δ_2 approaches L.
  This simply says that, as expected, when the ROM dimension, r, approaches the minimal ROM dimension, 1, the ROM lengthscale, δ_2, approaches the maximal FOM lengthscale, L (i.e., the dimension of the computational domain).

4. Numerical results

In this section, we perform a numerical investigation of the two lengthscales discussed in Section 3: the standard dimensionality-based ROM lengthscale, δ_1, defined in (16), and the new energy-based ROM lengthscale, δ_2, defined in (26). To this end, we use two fundamentally different ROMs: the ML-ROM presented in Section 2.2 and the EFR-ROM presented in Section 2.3. In each type of ROM, we use the two ROM lengthscales and compare the results. To this end, in our numerical investigation, we test four types of models: (i) ML-ROM1, which is the ML-ROM in which the velocity scale, U_{ML}, is set equal to δ_1; (ii) ML-ROM2, which is the ML-ROM in which the velocity scale, U_{ML}, is set equal to δ_2; (iii) EFR-ROM1, which is the EFR-ROM in which the filter radius, δ, is set equal to δ_1; and (iv) EFR-ROM2, which is the EFR-ROM in which the filter radius, δ, is set equal to δ_2. Thus, to compare the two lengthscales, we compare the four ROMs (i.e., ML-ROM1, ML-ROM2, EFR-ROM1, and EFR-ROM2) in the numerical simulation of the 3D turbulent channel flow at $Re_\tau = 395$. *We emphasize that the goal of this section is not to find the best ML-ROMs and EFR-ROMs. Instead, we investigate whether the two lengthscales are different and, if so, quantify their differences and how these differences impact the ML-ROM and EFR-ROM results.*

4.1. Numerical setting

FOM. The computational domain is a rectangular box, $\Omega = (-2\pi, 2\pi) \times (0,2) \times (-2\pi/3, 2\pi/3)$. We enforce no slip boundary conditions on the walls at $y = 0$ and $y = 2$, and periodic boundary conditions in the $x-$ and $z-$directions. We also use the forcing term $\boldsymbol{f} = [1,0,0]^\top$ and the Reynolds number $Re_\tau = 395$. To generate the snapshots, we use the reduced NS-α (rNS-α) Cuff et al., 2015; Eroglu et al., 2017; Rebholz et al., 2017b,a model, which is an LES model. The rNS-α model was proposed in Cuff et al. (2015) as a stable and efficient C^0 finite element approximation to the NS-α model, which was originally developed from the Camassa–Holm equations (Chen et al., 1998). To generate the snapshots, we utilize the same computational setting as that used in Rebholz et al. (2017a), where the rNS-α was shown to yield accurate results on coarse meshes in the numerical simulation of the turbulent channel flow at $Re_\tau = 395$. Specifically, to ensure that the Reynolds number is kept constant during the simulation, we dynamically adjust the bulk velocity at each time step (John and Roland, 2007). In the spatial discretization of the rNS-α model, we use a finite element discretization and the (P_3, P_2^{disc}) divergence-free Scott–Vogelius elements on a tetrahedral mesh that is heavily weighted towards the wall. The mesh provides 82 281 velocity and 53 760 pressure degrees of freedom. In the time discretization of the rNS-α model, we use the decoupled IMEX-BDF2 method with the time step size $\Delta t = 0.002$. Finally, we use an rNS-α filter radius $\alpha = 0.02$, which is the same as that used in Rebholz et al. (2017a).

ROM. We collect a total of 5000 snapshots from $t = 60$ to $t = 70$ (i.e., from a time interval in which the flow is in the statistically steady regime) and use the POD to generate the ROM basis. For illustrative purposes, we plot the magnitude of the velocity fields of the ROM basis functions $\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_{25}$, and $\boldsymbol{\varphi}_{50}$ in Fig. 1.

For the ROM time discretization, we utilize the commonly used linearized BDF2 temporal discretization with a time step size $\Delta t = 0.002$. As the ROM initial conditions, we use the ROM projections of the LES approximations at $t = 60$ and $t = 60.002$. For convenience, in our ROM simulations, $t = 0$ corresponds to $t = 60$ in the LES model.

To assess the ROMs' performance, we use two different criteria: (i) the time evolution of the kinetic energy, $E(t)$, and (ii) second-order statistics. For the ML-ROM assessment, we also use the energy spectrum.

We define the ROM kinetic energy as follows:

$$E(t) = \frac{1}{2} \int_\Omega \left(u_1(\boldsymbol{x},t)^2 + u_2(\boldsymbol{x},t)^2 + u_3(\boldsymbol{x},t)^2 \right) d\boldsymbol{x}\,, \tag{27}$$

where u_1, u_2, and u_3 are the components of the ROM velocity field approximation.

Following Rebholz et al. (2017a), we consider the following two second-order statistics: (i) the normalized root mean square (RMS) of the streamwise velocity component, U_{RMS}:

$$U_{RMS} := \frac{\left| \widetilde{\mathbb{R}}_{11} - \frac{1}{3} \sum_{j=1}^{3} \widetilde{\mathbb{R}}_{jj} \right|^{1/2}}{u_{1,\tau}}, \tag{28}$$

and (ii) the normalized streamwise-spanwise Reynolds stress tensor component, $\mathbb{R}_{12}$:

$$\mathbb{R}_{12} := \frac{\widetilde{\mathbb{R}}_{12}}{u_{1,\tau}^2}. \tag{29}$$

In these second-order statistics, the Reynolds stress tensor components are calculated as follows:

$$\widetilde{\mathbb{R}}_{ij} = \left\langle \left\langle u_i u_j \right\rangle_s \right\rangle_t - \left\langle \left\langle u_i \right\rangle_s \right\rangle_t \left\langle \left\langle u_j \right\rangle_s \right\rangle_t\,, \tag{30}$$

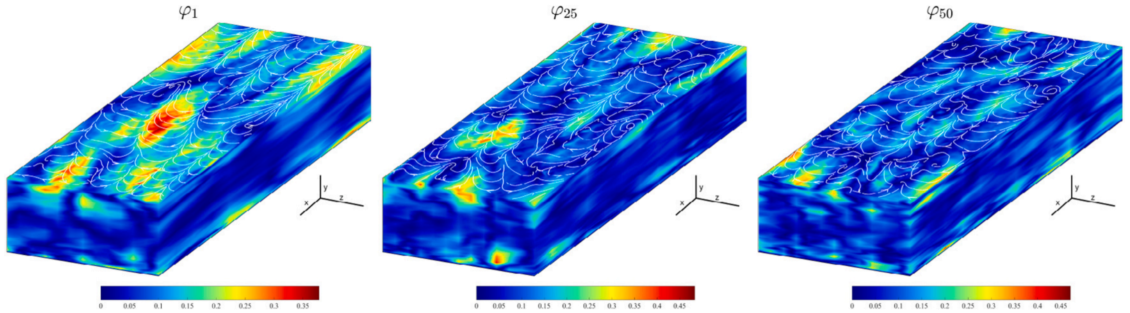


Fig. 1. The 1st, 25th, and 50th POD modes.

where $\langle \cdot \rangle_s$ denotes spatial averaging, $\langle \cdot \rangle_t$ denotes time averaging, and u_i are the components of the given ROM or FOM velocity field approximations. The friction velocity, $u_{1,\tau}$, which is used in (29), is calculated by using the following formula:

$$u_{1,\tau} = \sqrt{\nu \, \frac{U_{\text{mean}}(y_{min})}{y_{min}}}, \tag{31}$$

where ν is the kinematic viscosity, y_{min} the minimum positive y-value of the FOM mesh, and $U_{\text{mean}} = \left\langle \langle u_1 \rangle_s \right\rangle_t$ the average ROM or FOM velocity flow profile.

To calculate the energy spectrum, we perform a fast Fourier transform of the kinetic energy data.

4.2. Numerical results: Lengthscale comparison

In this section, we investigate the relative size of the two ROM lengthscales: (i) the standard dimensionality-based lengthscale, δ_1, defined in (16), and (ii) the novel energy-based ROM lengthscale, δ_2, defined in (26). To calculate δ_2 in Eq. (26), we define the FOM global mesh size, h, as $h = \max_{K \in \mathcal{K}} r_K$, where the mesh $\mathcal{K}$ is the set $\{K\}$ of tetrahedrons, and r_K is the inradius of the local tetrahedron, K. For the test problem used in our numerical investigation, $h = 1.1 \times 10^{-1}$. Furthermore, we define the maximal FOM lengthscale, L, as the vertical size of the computational domain, i.e., $L = 2$.

In Table 1, we list the δ_1 and δ_2 values for r values from 4 to 50. These results show that the two ROM lengthscales have very different behaviors with respect to changes in r.

Magnitude behavior. First, we notice that the δ_2 magnitude is between one and two orders of magnitude larger than the δ_1 magnitude. Indeed, for the low r values (i.e., $r = 4, 8, 16$), δ_2 is more than an order of magnitude larger than δ_1. For the large r values (i.e., $r = 32, 40, 50$), δ_2 is still an order of magnitude larger than δ_1, but the difference between the two lengthscales is smaller.

Asymptotic behavior. Second, the asymptotic behavior of the ROM lengthscales with respect to r is fundamentally different: As r increases, the δ_1 magnitude remains relatively unchanged. In contrast, as r increases, the δ_2 magnitude decreases by almost one order of magnitude. We emphasize that δ_2 has the natural asymptotic behavior expected from a ROM lengthscale, as explained in Remark 3.2: (i) As r increases toward its maximal value, i.e., R (which is $R = 804$ in our numerical investigation), δ_2 approaches h. Indeed, the results in Table 1 show that for the maximal r value (i.e., for $r = 50$), δ_2 achieves the smallest value, $\delta_2 = 4.32 \times 10^{-1}$, which is the same order of magnitude as the FOM mesh size, $h = 1.1 \times 10^{-1}$. (ii) As r approaches its minimal value (i.e., 1), δ_2 approaches L. Indeed, the results in Table 1 show that for the minimal r value (i.e., for $r = 1$), δ_2 achieves the largest value, $\delta_2 = 1.63 \times 10^{0}$, which is the same order of magnitude as the size of the computational domain, $L = 2.0 \times 10^{0}$.

Table 1
ROM lengthscale values for different r values.

r	4	8	16	32	40	50
δ_1	4.64e−02	4.65e−02	4.68e−02	4.68e−02	4.66e−02	4.62e−02
δ_2	1.63e00	1.41e+00	1.08e+00	6.84e−01	5.56e−01	4.32e−01

4.3. Numerical results: ML-ROM investigation

In this section, we investigate the role played by the two ROM lengthscales, δ_1 and δ_2, in the ML-ROM (13):

$$\boldsymbol{\tau} = -\left(\alpha \, U_{ML} \, L_{ML}\right) S_r \, \boldsymbol{a} \, . \tag{32}$$

We denote ML-ROM1 the ML-ROM in which $L_{ML} = \delta_1$ in (32) and ML-ROM2 the ML-ROM in which $L_{ML} = \delta_2$ in (32). To ensure a fair comparison of ML-ROM1 and ML-ROM2, we use the same constant α and the same velocity scale U_{ML} (i.e., the time averaged streamwise velocity component) in (32) for both models, and vary only the ROM lengthscale, i.e., $L_{ML} = \delta_1$ or $L_{ML} = \delta_2$. To vary the ROM lengthscale, we vary the r value in the definitions of δ_1 and δ_2. In Figs. 2–7, we plot the time evolution of the kinetic energy, the second-order statistics, and the energy spectra of the ML-ROM1 and ML-ROM2 for different r values and two different α values: $\alpha = 6 \times 10^{-3}$ (Figs. 2, 4, and 6) and $\alpha = 6 \times 10^{-4}$ (Figs. 3, 5, and 7). (Results for more α values are presented in the preliminary numerical investigation in Mou et al. (2022).) As a benchmark for the ROM results, we use the projection of the FOM results on the ROM basis (denoted as LES-proj in these plots). Thus, for each ROM space dimension, r, we ensure that the benchmark is the best possible approximation of the FOM data in the r-dimensional ROM space.

Stability. Since the results in Table 1 show that δ_2 is between one and two orders of magnitude higher than δ_1, we expect ML-ROM2 to yield more stable results than ML-ROM1. Indeed, since we fix all the ML-ROM parameters (i.e., the constant α and the velocity scale U_{ML}) and δ_2 is significantly larger than δ_1, we expect the ML-ROM2 artificial viscosity to be higher than the ML-ROM1 artificial viscosity and, thus, ML-ROM2 to be more stable than ML-ROM1. Specifically, when there is not enough artificial viscosity to dissipate energy from the system at the correct physical rate, we expect the ML-ROM kinetic energy to grow to unphysically high levels and, eventually, to blow up. In Figs. 2–7, ML-ROM2 yields stable results for *all* r values and for *both* α values. In contrast, for the smallest α value, $\alpha = 6 \times 10^{-4}$ (Figs. 3, 5, and 7), ML-ROM1 blows up for *all* r values. (We note that, in the energy spectrum plot in Fig. 6, the ML-ROM1 blowup is manifested in unphysical energy values, which are much higher than the LES ones.) Furthermore, for the largest α value, $\alpha = 6 \times 10^{-3}$ (Figs. 2, 4, and 6), ML-ROM1 blows up for the small r values (i.e., $r = 4, 8, 16$, and 32). To quantify the stability of the two ML-ROMs, in Table 2, for different r values, we list the threshold α_0 value, i.e., the value that ensures that, if $\alpha > \alpha_0$, then the ML-ROM is stable. These results show that, for each r value, the

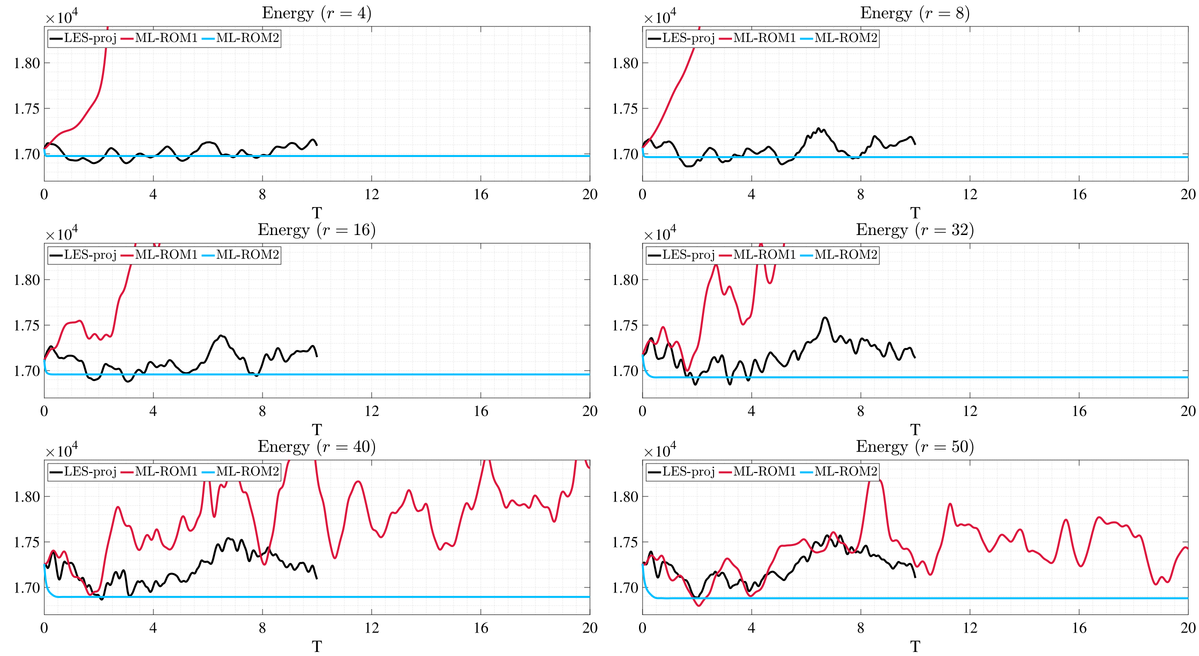


Fig. 2. Time evolution of the ML-ROM kinetic energy for $\alpha = 6 \times 10^{-3}$.

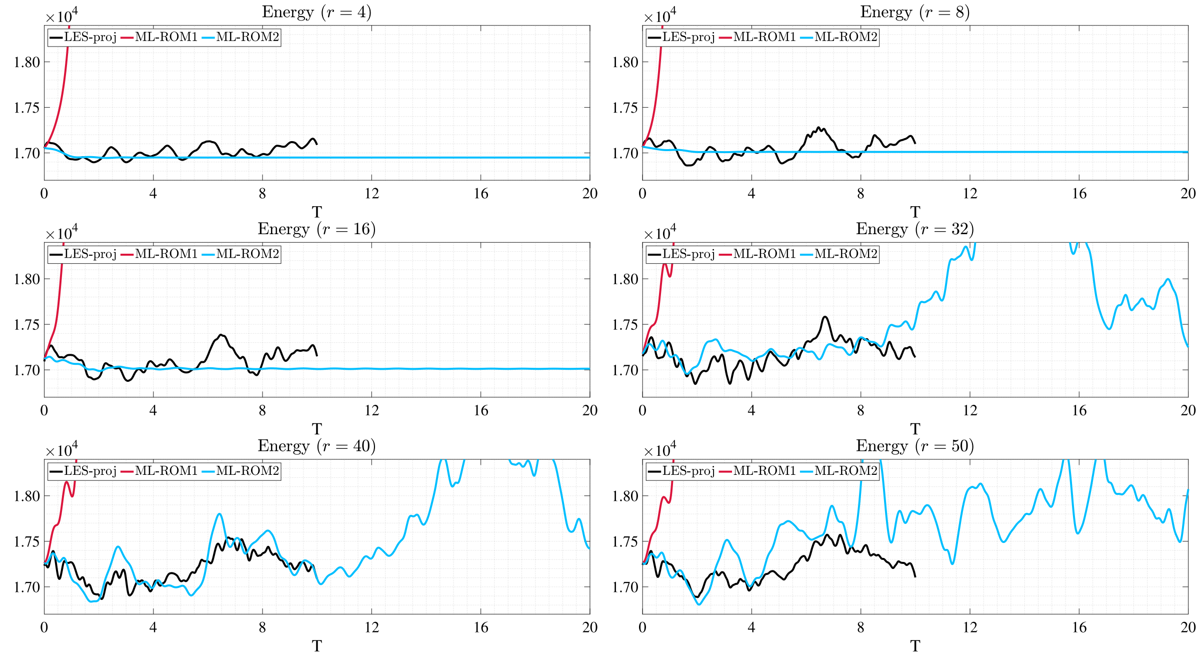


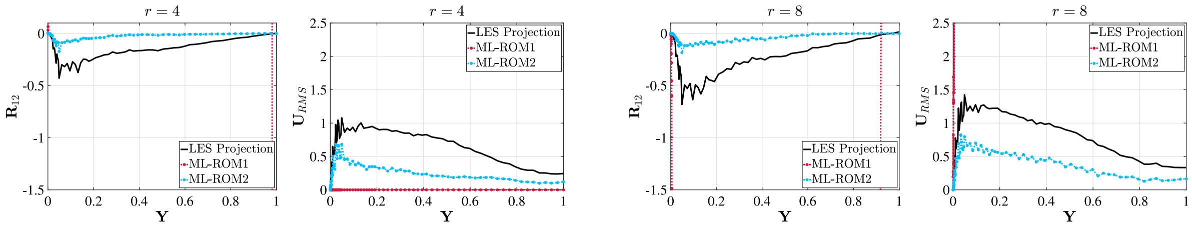
Fig. 3. Time evolution of the ML-ROM kinetic energy for $\alpha = 6 \times 10^{-4}$.

Table 2
ML-ROM threshold α_0 values for different r values.

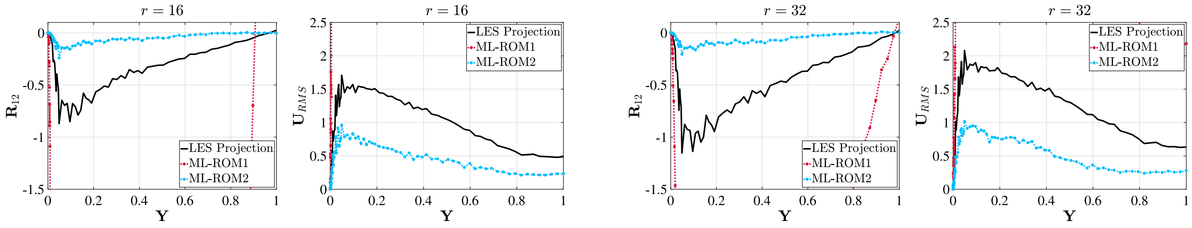
	r	4	8	16	32	40	50
ML-ROM1	α_0	10e−3	9.8e−3	9.2e−3	8.5e−3	6.5e−3	6.2e−3
ML-ROM2	α_0	2.9e−4	3.4e−4	4.1e−4	6.7e−4	5.9e−4	7.5e−4

threshold α_0 value is more than an order of magnitude lower for ML-ROM2 than for ML-ROM1. Thus, we conclude that ML-ROM2 is more stable than ML-ROM1, which is the same conclusion as that yielded by Figs. 2–7.

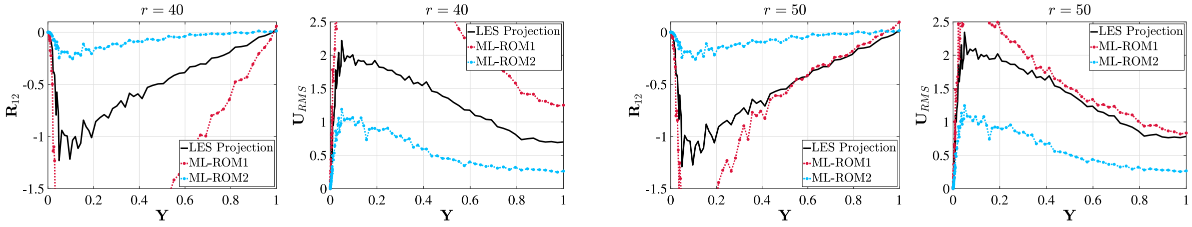
Accuracy. Since δ_2 is between one and two orders of magnitude higher than δ_1, we expect the ML-ROM accuracy to depend on the constant α. Indeed, the ML-ROM1 and ML-ROM2 plots in Figs. 2–7 do not show a clear winner: For the largest α value (i.e., $\alpha = 6 \times 10^{-3}$), ML-ROM1 is more accurate than ML-ROM2 for large r values (i.e., $r = 40$ and 50) and less accurate for small r values (i.e., $r = 4, 8$, and 16). For the smallest α value (i.e., $\alpha = 6 \times 10^{-4}$), ML-ROM2 is more accurate than ML-ROM1 for *all* r values (since ML-ROM1 simply blows up). We also note that, for $\alpha = 6\times10^{-4}$, ML-ROM2 is quite accurate for larger r values (i.e., $r = 32, 40$, and 50).



(a) $r = 4$ (b) $r = 8$



(c) $r = 16$ (d) $r = 32$



(e) $r = 40$ (f) $r = 50$

Fig. 4. Second-order ML-ROM statistics for $\alpha = 6 \times 10^{-3}$.

Table 3
ML-ROM optimal α values for different r values.

	r	4	8	16	32	40	50
ML-ROM1	α	1.50e−2	1.38e−2	1.38e−2	1.28e−2	7.80e−3	6.51e−3
ML-ROM2	α	4.35e−4	4.42e−4	5.33e−4	6.70e−4	6.20e−4	7.50e−4

Parameter sensitivity. Finally, we investigate the ML-ROM's parameter sensitivity. Specifically, we investigate which ROM lengthscale yields ML-ROMs that are less sensitive (i.e., more robust) with respect to the ML-ROM's α parameter. We emphasize that the model sensitivity with respect to model parameters is a well known issue that has hindered the development of closures and stabilizations in CFD over the years (Sagaut, 2006; Berselli et al., 2006; Layton and Rebholz, 2012). Thus, finding robust (i.e., less sensitive) ML-ROMs that require as little parameter tuning as possible is an important practical problem.

To quantify the ML-ROM's parameter sensitivity, in Table 3, for different r values, we list the optimal α value in ML-ROM, i.e., the α value that ensures that the average ROM kinetic energy, $\overline{KE}^{ROM}$, is the closest to the average FOM (LES) kinetic energy, $\overline{KE}^{LES}$. Specifically, we solve the following optimization problem:

$$\min_{\alpha} \left| \overline{KE}^{ROM} - \overline{KE}^{LES} \right| , \tag{33}$$

where $\overline{KE} = \frac{1}{M} \sum_{k=1}^{M} KE(t_k)$, and M is the number of snapshots.

The results in Table 3 show that the optimal ML-ROM α values are sensitive with respect to changes in r. Indeed, as r increases from 4 to 50, α decreases by a factor of 2.3 for ML-ROM1 and increases by a factor of 1.7 for ML-ROM2. Overall, the results in Table 3 show that the ML-ROM's parameter α sensitivity is higher for ML-ROM1 than for ML-ROM2.

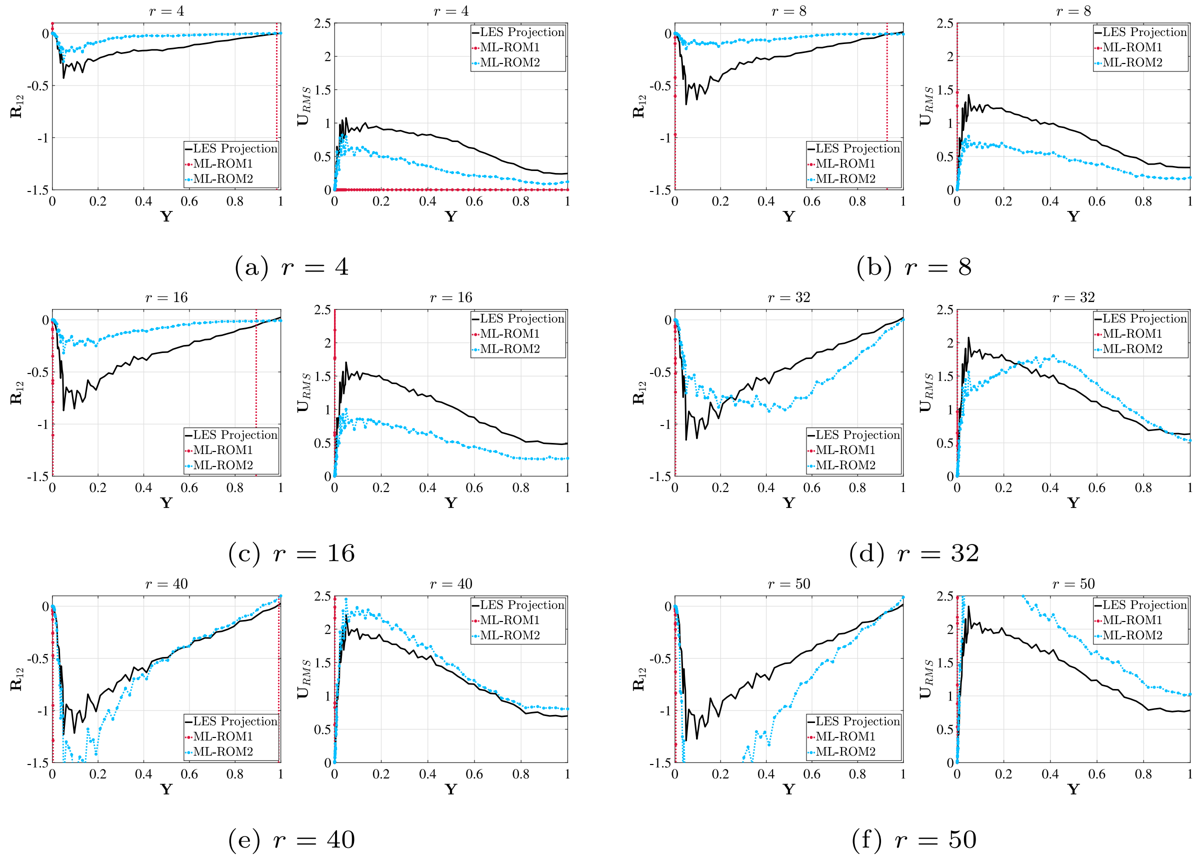
4.4. Numerical results: EFR-ROM investigation

To further investigate the role played by the two ROM lengthscales, δ_1 and δ_2, in reduced order modeling of turbulent flows, in this section we consider the EFR-ROM presented in Section 2.3. We emphasize that the EFR-ROM is completely different from the ML-ROM investigated in Section 4.3: The EFR-ROM is a Reg-ROM based on numerical stabilization, whereas the ML-ROM is a closure model. Thus, the EFR-ROM investigation in this section could shed new light on the novel ROM lengthscale, δ_2.

To ensure a fair comparison of the two ROM lengthscales, δ_1 and δ_2, we fix the parameters γ and χ in the EFR-ROM in Section 2.3 and change only the filter radius, δ, of the differential filter used in Step (II) of the EFR-ROM algorithm. We denote the resulting models as EFR-ROM1 (when $\delta = \delta_1$) and EFR-ROM2 (when $\delta = \delta_2$). In our numerical investigation, we fix the χ value for both EFR-ROM1 and EFR-ROM2 to $\chi = 6 \times 10^{-3}$, which is the minimum value that yields a stable solution for $r = 4$ EFR-ROM1. We also fix the γ value for both EFR-ROM1 and EFR-ROM2. In our numerical investigation, we consider two γ values. To vary the ROM lengthscale, we vary the r value in the definitions of δ_1 and δ_2.

In Figs. 8–11, we plot the time evolution of the kinetic energy and the second-order statistics of the EFR-ROM1 and EFR-ROM2 for different r values and two different γ values: $\gamma = 8\times10^{-2}$ (Figs. 8 and 10) and $\gamma = 9 \times 10^{-1}$ (Figs. 9 and 11). As a benchmark for the ROM results, we use the projection of the FOM results on the ROM basis (denoted as LES-proj in these plots).

Stability. Since the results in Table 1 show that δ_2 is between one and two orders of magnitude higher than δ_1, we expect EFR-ROM2 to yield more stable results than EFR-ROM1. Indeed, since we fix all the EFR-ROM parameters (i.e., χ and γ) and δ_2 is significantly larger than δ_1,



(a) $r = 4$ (b) $r = 8$ (c) $r = 16$ (d) $r = 32$ (e) $r = 40$ (f) $r = 50$

Fig. 5. Second-order ML-ROM statistics for $\alpha = 6 \times 10^{-4}$.

Table 4
EFR-ROM threshold γ_0 values for different r values.

	r	4	8	16	32	40	50
EFR-ROM1	γ_0	1.0e0	1.0e0	9.0e−1	8.0e−1	7.0e−1	7.0e−1
EFR-ROM2	γ_0	2.9e−2	3.3e−2	3.9e−2	5.5e−2	5.9e−2	7.5e−2

Table 5
EFR-ROM optimal γ values for $\chi = 6 \times 10^{-3}$ and different r values.

	r	4	8	16	32	40	50
EFR-ROM1	γ	1.01e0	9.98e0	9.03e−1	8.05e−1	8.44e−1	7.00e−1
EFR-ROM2	γ	2.90e−2	3.30e−2	3.90e−2	5.50e−2	7.08e−2	7.50e−2

we expect the EFR-ROM2 filtering level to be higher than the EFR-ROM1 filtering level (and, thus, EFR-ROM2 to be more stable than EFR-ROM1). This is clearly shown in the plots in Figs. 8–11, in which EFR-ROM2 yields stable results for *all* r values and for *both* γ values. In contrast, for the smallest γ value, $\gamma = 8 \times 10^{-2}$ (Figs. 8 and 10), EFR-ROM1 blows up for *all* r values. Furthermore, for the largest γ value, $\gamma = 9 \times 10^{-1}$ (Figs. 9 and 11), EFR-ROM1 blows up for the small r values (i.e., $r = 4$ and $r = 8$). To quantify the stability of the two EFR-ROMs, in Table 4, for different r values, we list the threshold γ_0 value, i.e., the value that ensures that, if $\gamma > \gamma_0$, then the EFR-ROM is stable. These results show that, for each r value, the threshold γ_0 value is more than an order of magnitude lower for EFR-ROM2 than for EFR-ROM1. Thus, we conclude that EFR-ROM2 is more stable than EFR-ROM1, which is the same conclusion as that yielded by Figs. 8–11.

Accuracy. Since δ_2 is between one and two orders of magnitude higher than δ_1, we expect the EFR-ROM accuracy to depend on the constant γ. Indeed, the EFR-ROM1 and EFR-ROM2 plots in Figs. 8–11 do not display a clear winner: For the largest γ value (i.e., $\gamma = 9 \times 10^{-1}$), EFR-ROM2 is more accurate than EFR-ROM1 for the small r values (i.e., $r = 4$ and 8) since EFR-ROM1 simply blows up. For the remaining r values, EFR-ROM1 and EFR-ROM2 display the same accuracy level. For the smallest γ value (i.e., $\gamma = 8 \times 10^{-2}$), EFR-ROM2 is more accurate than EFR-ROM1 for *all* r values (since EFR-ROM1 simply blows up). We also note that EFR-ROM2 is relatively accurate for the largest r value (i.e., $r = 50$).

Parameter sensitivity. To study the EFR-ROM's parameter sensitivity, we investigate which ROM lengthscale yields EFR-ROMs that are less sensitive (i.e., more robust) with respect to the EFR-ROM's γ parameter. To quantify the ML-ROM's parameter sensitivity, in Table 5, for different r values, we list the optimal γ value in EFR-ROM, i.e., the γ value that ensures that the average ROM kinetic energy, $\overline{KE}^{ROM}$, is the closest to the average FOM (LES) kinetic energy, $\overline{KE}^{LES}$. Specifically, we solve the following optimization problem:

$$\min_{\gamma} \left| \overline{KE}^{ROM} - \overline{KE}^{LES} \right| . \tag{34}$$

The results in Table 5 display a relatively low sensitivity of the optimal EFR-ROM parameter γ with respect to changes in r. Indeed, as r varies, the order of magnitude of the optimal γ remains the same for both EFR-ROM1 and EFR-ROM2, although the latter is more sensitive than the former.

One possible explanation for the relatively low sensitivity of the optimal EFR-ROM parameter γ is that the EFR-ROM parameter χ, which controls the amount of filtering in the relaxation step of the EFR-ROM algorithm, is low ($\chi = 6 \times 10^{-3}$). Thus, only 0.6% filtering is applied at each time step of the EFR-ROM algorithm. Since only a low amount of filtering is used, the effect of the filter radius (i.e., the

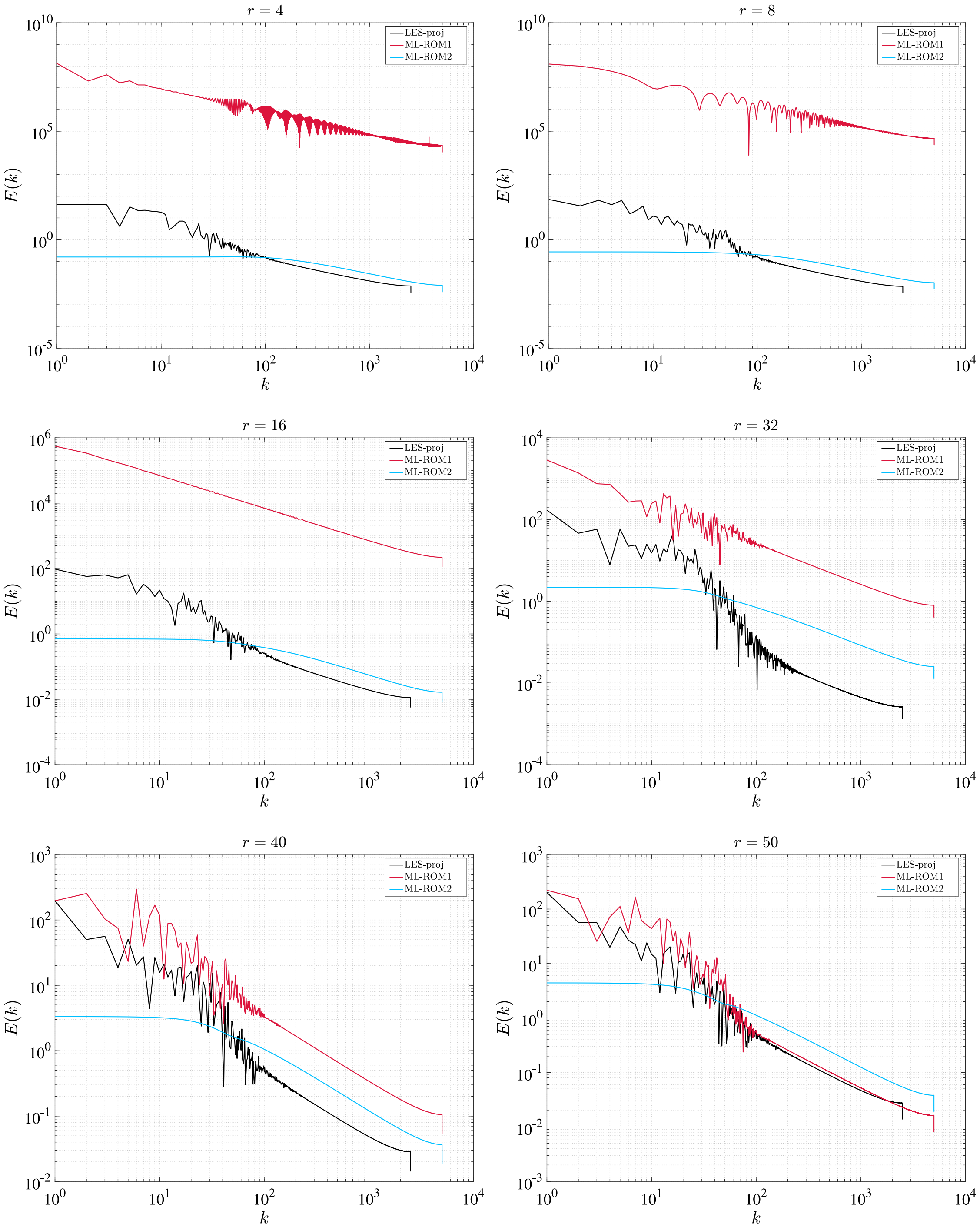


Fig. 6. ML-ROM energy spectrum for $\alpha = 6 \times 10^{-3}$.

Table 6
EFR-ROM optimal γ values for $\chi = 6 \times 10^{-2}$ and different r values.

	r	4	8	16	32	40	50
EFR-ROM1	γ	2.92e−1	6.25e−2	6.00e−2	4.84e−2	4.41e−2	3.61e−2
EFR-ROM2	γ	2.36e−4	6.83e−5	1.12e−04	2.26e−4	3.10e−4	4.13e−4

ROM lengthscales δ_1 and δ_2) is not as important in the EFR-ROM case as in the ML-ROM case.

To investigate whether a higher percentage of filtering yields a higher sensitivity of the optimal EFR-ROM parameter γ, we increase the χ value. Specifically, we choose $\chi = 6 \times 10^{-2}$, i.e., 6% filtering at each EFR-ROM time step (Table 6). The results in Table 6 show that the optimal EFR-ROM γ value is very sensitive with respect to changes in r. Indeed, as r increases from 4 to 50, γ decreases by almost one order of magnitude. The optimal EFR-ROM2 γ values are less sensitive with respect to changes in r than the optimal EFR-ROM1 γ values. Indeed, although the EFR-ROM2 γ values vary, their order of magnitude generally stays constant.

Overall, the results in Tables 5–6 show that the EFR-ROM's parameter γ sensitivity is higher for EFR-ROM1 than for EFR-ROM2 when a high percentage of filtering is used in the EFR-ROM algorithm. As expected, when a low percentage of filtering is used, both EFR-ROM1 and EFR-ROM2 display a relatively low γ sensitivity.

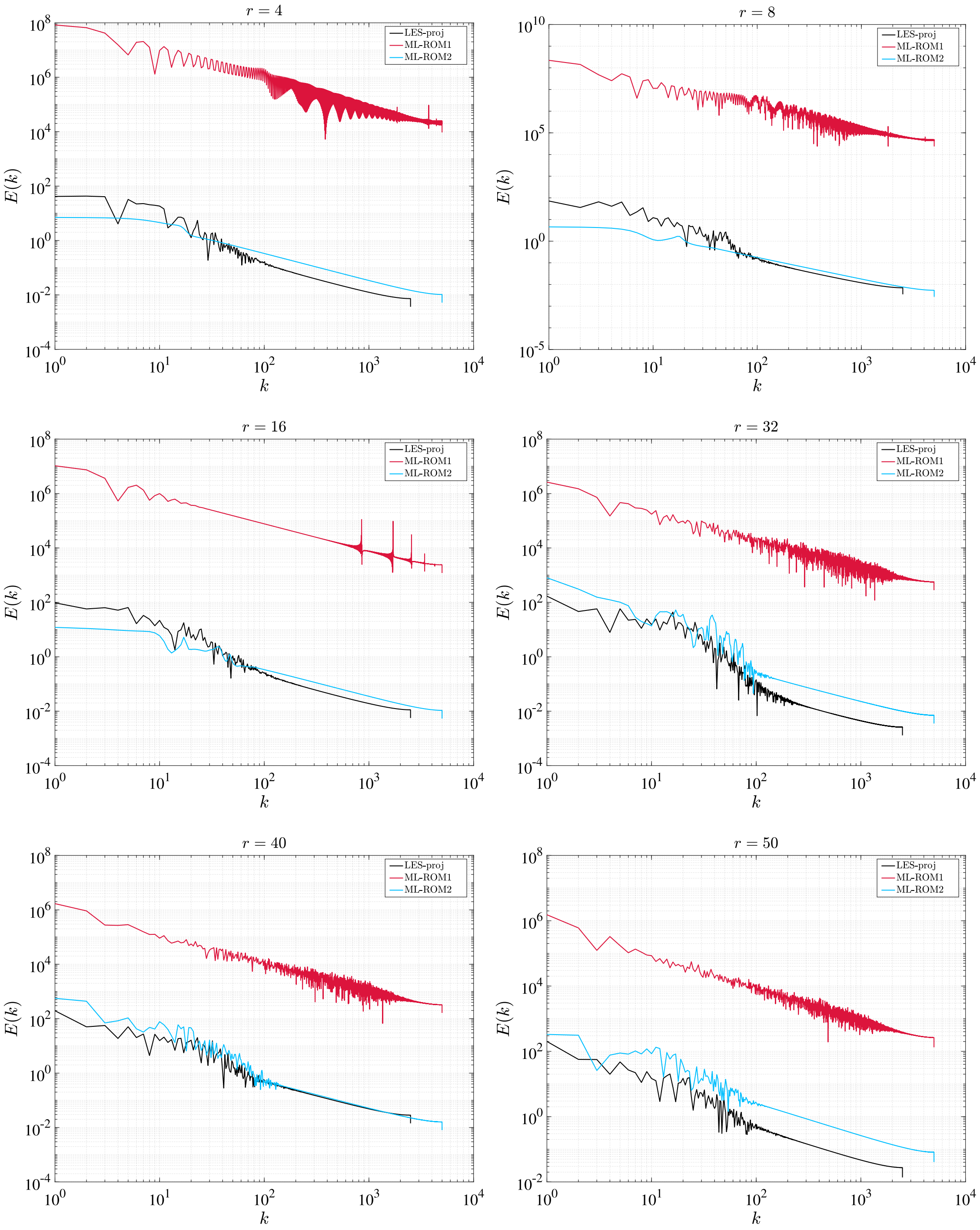


Fig. 7. ML-ROM energy spectrum for $\alpha = 6 \times 10^{-4}$.

Based on the results in Figs. 8–11 and in Tables 5–6, we conclude that the EFR-ROM investigation in this section yields qualitative results that are similar to those yielded by the ML-ROM investigation in Section 4.3.

4.5. *Numerical results: Computational cost*

In Table 7, we list the online CPU time (in seconds) for ML-ROM1, ML-ROM2, EFR-ROM1, and EFR-ROM2 for different r values. We note that, as expected, as we increase the ROM dimension, r, the CPU time increases. We emphasize, however, that this increase in CPU time is modest for all the r values considered. Furthermore, we note that the CPU time of all four ROMs and all six r values is on the same order of magnitude, i.e., $\mathcal{O}(10^2)$. Given that the online CPU time of the LES model used to generate the data for the ROM basis construction is $\mathcal{O}(10^5)$ (Eroglu et al., 2017; Rebholz et al., 2017a), we obtain speedup factors on the order of $\mathcal{O}(10^3)$ for all the ROMs.

5. Conclusions and outlook

In this paper, we proposed a novel ROM lengthscale definition. This new ROM lengthscale, denoted δ_2, was constructed by using energy distribution arguments. Specifically, we balanced the ROM and FOM energy content with the energy content in the δ_2 and h scales,

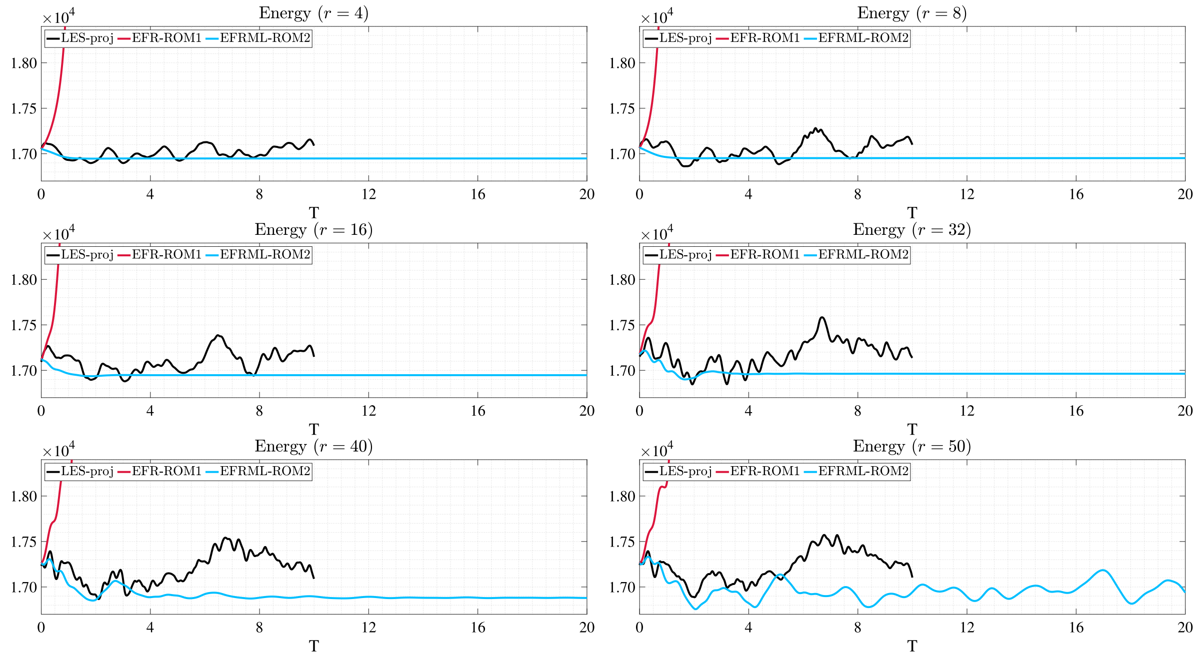


Fig. 8. Time evolution of the EFR-ROM kinetic energy for $\gamma = 8 \times 10^{-2}$.

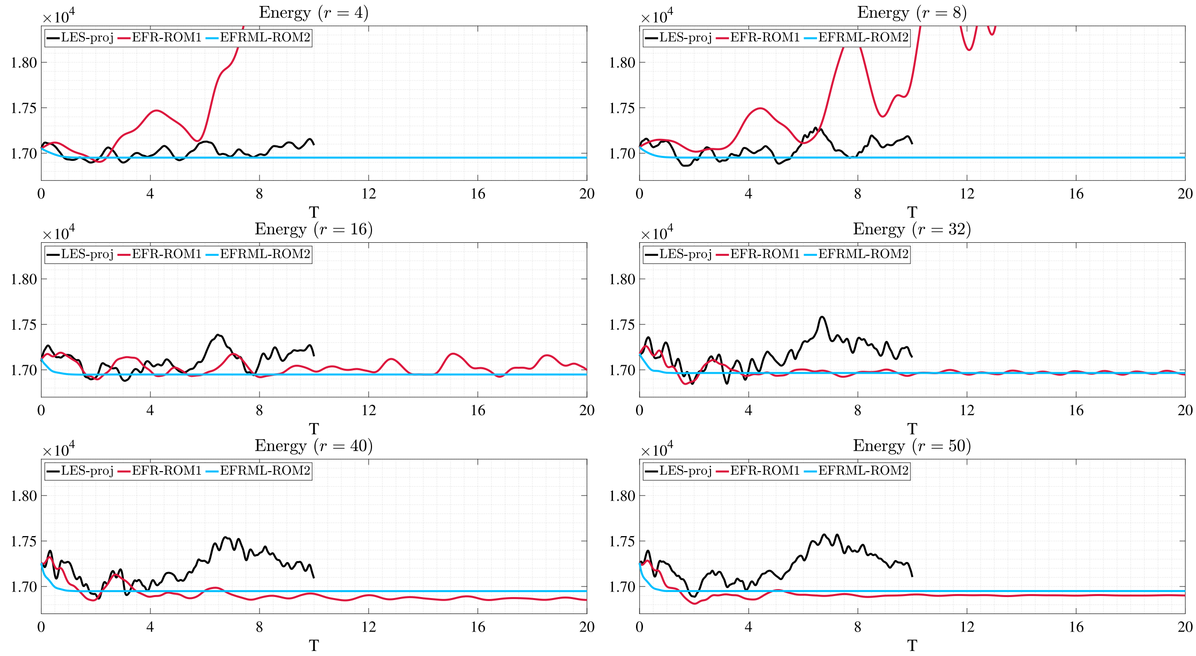


Fig. 9. Time evolution of the EFR-ROM kinetic energy for $\gamma = 9 \times 10^{-1}$.

Table 7
ML-ROM and EFR-ROM CPU time (in seconds) for different r values.

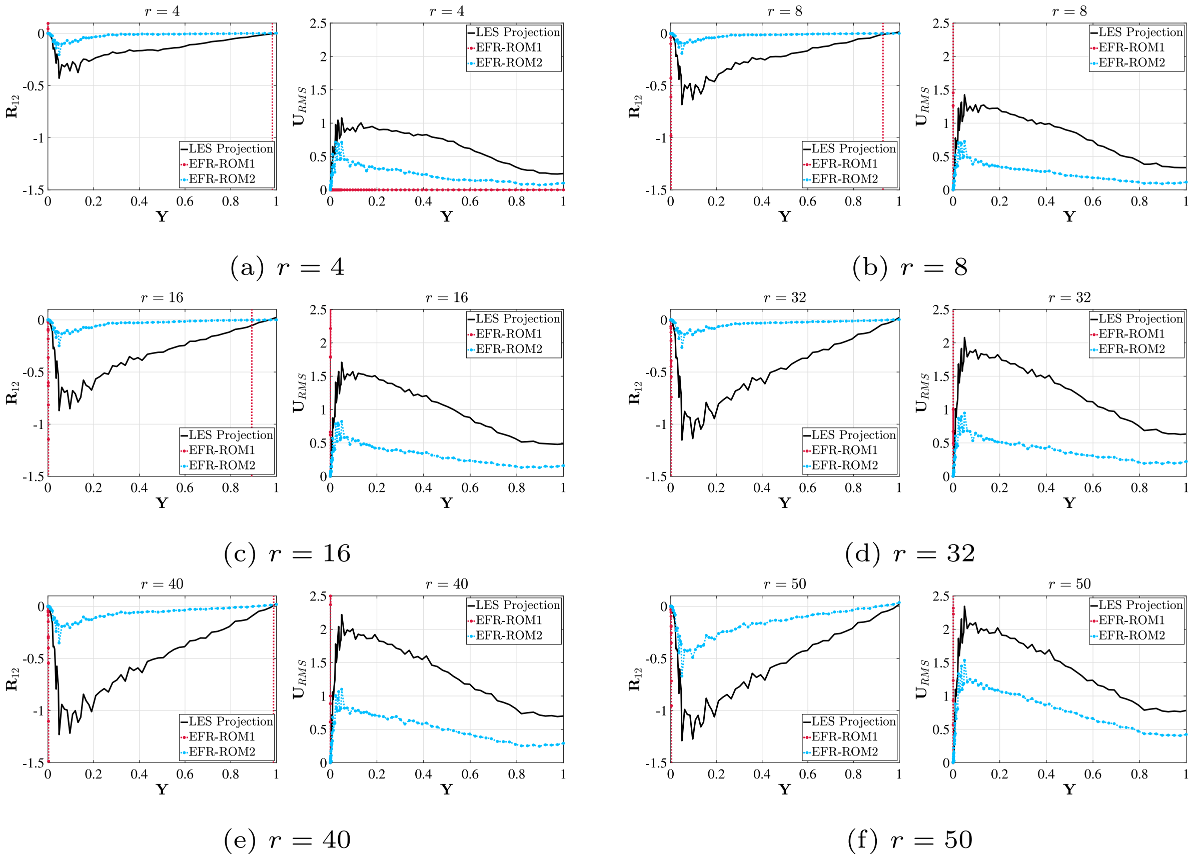
r	4	8	16	32	40	50
ML-ROM1	2.83e2	2.86e2	2.93e2	3.06e2	3.16e2	3.20e2
ML-ROM2	2.82e2	2.83e2	3.02e2	3.07e2	3.12e2	3.31e2
EFR-ROM1	2.79e2	2.86e2	2.93e2	3.04e2	3.16e2	3.25e2
EFR-ROM2	2.83e2	2.87e2	2.91e2	3.16e2	3.11e2	3.22e2

respectively, where h is the FOM mesh size. We emphasize that the novel ROM lengthscale, δ_2, is fundamentally different from the current ROM lengthscales, which are built by using dimensional arguments.

We compared the new energy-based ROM lengthscale, δ_2, with a standard dimensionality-based ROM lengthscale, denoted δ_1. To this end, we used these two ROM lengthscales to build two mixing-length ROMs (ML-ROMs) and two evolve-filter-relax ROMs (EFR-ROMs) in which all the other parameters were the same. We investigated the four resulting ML-ROMs and EFR-ROMs in the numerical simulation of the turbulent channel flow at $Re_\tau = 395$.

The numerical results of our investigation yielded the following conclusions:

1. The new energy-based ROM lengthscale, δ_2, was significantly (two orders of magnitude) larger than the standard ROM lengthscale, δ_1. As a result, the ML-ROMs and EFR-ROMs based on the



(a) $r = 4$

(b) $r = 8$

(c) $r = 16$

(d) $r = 32$

(e) $r = 40$

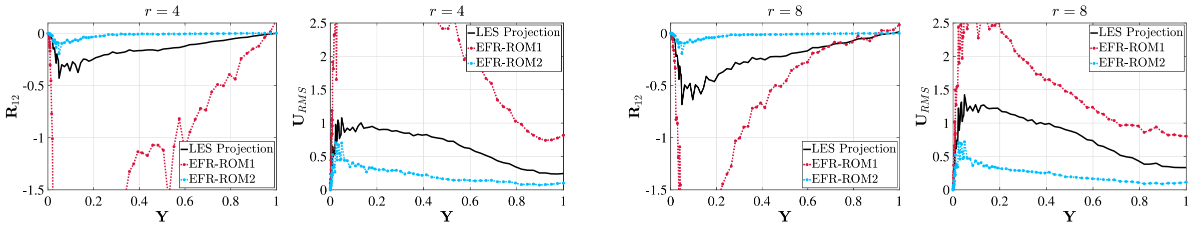
(f) $r = 50$

Fig. 10. Second-order EFR-ROM statistics for $\gamma = 8 \times 10^{-2}$.

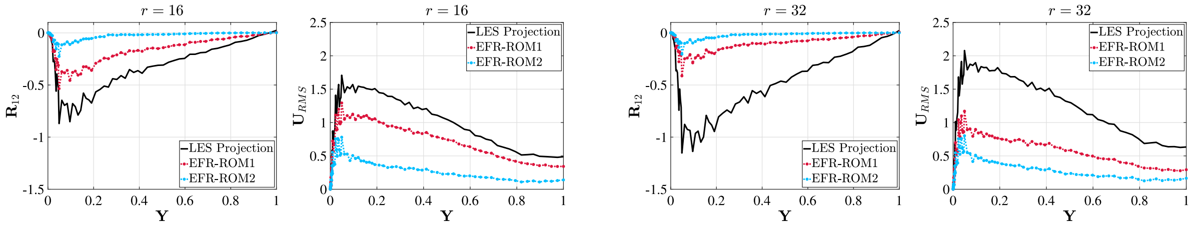
new ROM lengthscale were significantly more stable than the ML-ROMs and EFR-ROMs based on the standard ROM lengthscale.

2. The new energy-based ROM lengthscale displayed the correct asymptotic behavior with respect to the ROM dimension, whereas the standard dimensionality-based ROM lengthscale did not. Indeed, as r increased, the δ_1 magnitude remained relatively unchanged. In contrast, as r increased, the δ_2 magnitude decreased by almost one order of magnitude. We emphasize that δ_2 displayed the natural asymptotic behavior expected from a ROM lengthscale: As r increased toward its maximal value, δ_2 approached the mesh size, h (i.e., the minimum spatial scale that can be approximated with the given FOM data). Furthermore, as r approached its minimal value (i.e., 1), δ_2 approached the computational domain characteristic lengthscale, L (i.e., the maximum spatial scale that can be approximated with the given FOM data). We believe that δ_2's *adaptivity* with respect to the ROM dimension, r, is important in realistic settings where using the FOM data to determine an optimal ROM lengthscale may not be possible. Indeed, as the ROM dimension is changed according to the user's needs, δ_2 adapts to the new setting automatically, without the user's intervention.
3. The ML-ROM parameters based on the new energy-based ROM lengthscale were less sensitive (i.e., more robust) with respect to changes in the ROM dimension (r) than the ML-ROMs parameters based on the standard dimensionality-based ROM lengthscale. The EFR-ROM parameters based on the new lengthscale were less sensitive with respect to changes in r than the EFR-ROMs parameters based on the standard lengthscale when a significant percentage of filtering was performed. For a low filtering percentage, as expected, the EFR-ROM parameters based on the two lengthscales displayed a relatively low sensitivity. In this setting, the standard lengthscale yielded less sensitive parameters than the new lengthscale.

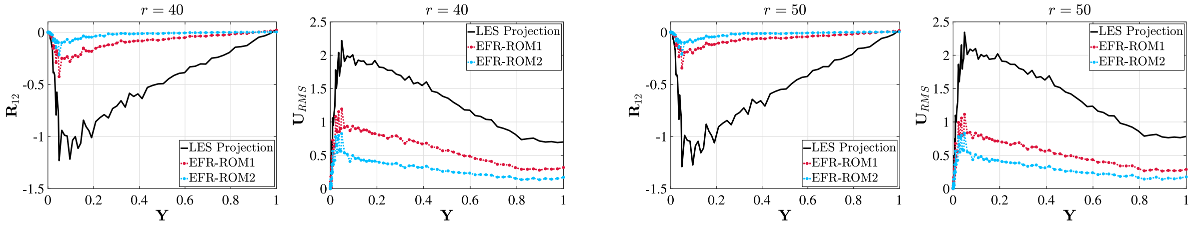
We note that the importance of advanced closure models for turbulent heat flux in nuclear research was recently emphasized in Fiore et al. (2022). An example of this type of strategy is the introduction of transport equations for statistics, which enable the computation of thermal lengthscales and timescales, as described in Manservisi and Menghini (2014). The use of lengthscales to develop new ROMs for nuclear engineering problems is a relatively new concept. The numerical assessment of the new energy-based ROM lengthscale yielded encouraging results. We plan to further investigate this ROM lengthscale in new settings. For example, we will study the new ROM lengthscale in the construction of other types of ROMs, e.g., LES ROMs (Wang et al., 2012; Xie et al., 2017). Furthermore, we plan to investigate the effect of the FOM resolution (e.g., mesh size and LES filter radius) on the corresponding ROM lengthscale results. We also intend to leverage the new energy-based lengthscale to develop scale-aware ROM strategies that are better suited for flow-specific applications. Finally, we plan to formulate our method using reinforcement learning (San et al., 2022) to discover latent underlying physics-specific lengthscale dynamics in an adaptable manner.



(a) $r = 4$ (b) $r = 8$



(c) $r = 16$ (d) $r = 32$



(e) $r = 40$ (f) $r = 50$

Fig. 11. Second-order EFR-ROM statistics for $\gamma = 9 \times 10^{-1}$.

CRediT authorship contribution statement

Changhong Mou: Methodology, Software, Investigation, Visualization. **Elia Merzari:** Methodology, Writing – review & editing. **Omer San:** Methodology, Writing – review & editing. **Traian Iliescu:** Conceptualization, Methodology, Writing – original draft, Supervision.

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 thank the reviewers for their comments and suggestions that significantly improved the paper. The work of the first and fourth authors was supported by National Science Foundation through grant DMS-2012253 and CDS&E-MSS-1953113. The third author gratefully acknowledges the U.S. DOE Early Career Research Program support through grant DE-SC0019290 and the National Science Foundation support through grant DMS-2012255. Part of this work was funded under the nuclear energy advanced modeling and simulation program.

References

Ahmed, S.E., Pawar, S., San, O., Rasheed, A., Iliescu, T., Noack, B.R., 2021. On closures for reduced order models – A spectrum of first-principle to machine-learned avenues. Phys. Fluids 33 (9), 091301.

Aradag, S., Siegel, S., Seidel, J., Cohen, K., McLaughlin, T., 2011. Filtered POD-based low-dimensional modeling of the 3D turbulent flow behind a circular cylinder. Internat. J. Numer. Methods Fluids 66 (1), 1–16.

Aubry, N., Holmes, P., Lumley, J.L., Stone, E., 1988. The dynamics of coherent structures in the wall region of a turbulent boundary layer. J. Fluid Mech. 192, 115–173.

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

Berselli, L.C., Iliescu, T., Layton, W.J., 2006. Mathematics of Large Eddy Simulation of Turbulent Flows. In: Scientific Computation, Springer-Verlag, Berlin, p. xviii+348.

Bertagna, L., Quaini, A., Veneziani, A., 2016. Deconvolution-based nonlinear filtering for incompressible flows at moderately large Reynolds numbers. Internat. J. Numer. Methods Fluids 81 (8), 463–488.

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

Chen, S., Foias, C., Holm, D.D., Olson, E., Titi, E.S., Wynne, S., 1998. Camassa–Holm equations as a closure model for turbulent channel and pipe flow. Phys. Rev. Lett. 81 (24), 5338.

Couplet, M., Sagaut, P., Basdevant, C., 2003. Intermodal energy transfers in a proper orthogonal decomposition–Galerkin representation of a turbulent separated flow. J. Fluid Mech. 491, 275–284.

Cuff, V.M., Dunca, A.A., Manica, C.C., Rebholz, L.G., 2015. The reduced order NS-α model for incompressible flow: Theory, numerical analysis and benchmark testing. ESAIM Math. Model. Numer. Anal. 49 (3), 641–662.

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

Eroglu, F.G., Kaya, S., Rebholz, L.G., 2017. A modular regularized variational multiscale proper orthogonal decomposition for incompressible flows. Comput. Methods Appl. Mech. Engrg. 325, 350–368.

Ervin, V.J., Layton, W.J., Neda, M., 2012. Numerical analysis of filter-based stabilization for evolution equations. SIAM J. Numer. Anal. 50 (5), 2307–2335.

Farcas, I., Munipalli, R., Willcox, K.E., 2022. On filtering in non-intrusive data-driven reduced-order modeling. In: AIAA AVIATION 2022 Forum. p. 3487.

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

Fiore, M., Koloszar, L., Mendez, M.A., Duponcheel, M., Bartosiewicz, Y., 2022. Turbulent heat flux modelling in forced convection flows using artificial neural networks. Nucl. Eng. Des. 399, 112005.

Fischer, P.F., Mullen, J., 2001. Filter-based stabilization of spectral element methods. C. R. Acad. Sci. Paris Sér. I Math. 332 (3), 265–270.

Frisch, U., 1995. Turbulence, the Legacy of A.N. KOlmogorov. Cambridge University Press, Cambridge.

German, P., Tano, M., Fiorina, C., Ragusa, J.C., 2022. GeN-ROM—An OpenFOAM®-based multiphysics reduced-order modeling framework for the analysis of Molten Salt Reactors. Prog. Nucl. Energy 146, 104148.

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

Girfoglio, M., Quaini, A., Rozza, G., 2021b. Pressure stabilization strategies for a LES filtering reduced order model. Fluids 6 (9), 302.

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

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

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

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

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

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

John, V., Roland, M., 2007. Simulations of the turbulent channel flow at Reτ=180 with projection-based finite element variational multiscale methods. Internat. J. Numer. Methods Fluids 55 (5), 407–429.

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

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

Liu, Y., Hu, R., Zou, L., Hu, G., Nunez, D., 2022. SAM-ML: Machine learning enhanced system analysis module for thermal stratification analysis. In: 19th International Conference in Nuclear Reactor Thermal-Hydraulics.

Lorenzi, S., Cammi, A., Luzzi, L., Rozza, G., 2016. POD-Galerkin method for finite volume approximation of Navier–Stokes and RANS equations. Comput. Methods Appl. Mech. Engrg. 311, 151–179.

Manservisi, S., Menghini, F., 2014. A CFD four parameter heat transfer turbulence model for engineering applications in heavy liquid metals. Int. J. Heat Mass Transfer 69, 312–326.

Merzari, E., Obabko, A., Fischer, P., Halford, N., Walker, J., Siegel, A., Yu, Y., 2017. Large-scale large eddy simulation of nuclear reactor flows: Issues and perspectives. Nucl. Eng. Des. 312, 86–98.

Merzari, E., Pointer, W.D., Fischer, P., 2011. A POD-based solver for the advection-diffusion equation. In: Fluids Engineering Division Summer Meeting, Vol. 44403. pp. 1139–1147.

Mohebujjaman, M., Rebholz, L.G., Iliescu, T., 2019. Physically-constrained data-driven correction for reduced order modeling of fluid flows. Internat. J. Numer. Methods Fluids 89 (3), 103–122.

Mou, C., Merzari, E., San, O., Iliescu, T., 2022. A numerical investigation of the lengthscale in the mixing-length reduced order model of the turbulent channel flow. In: 19th International Topical Meeting on Nuclear Reactor Thermal Hydraulics. NURETH-19, Brussels, Belgium.

Mullen, J.S., Fischer, P.F., 1999. Filtering techniques for complex geometry fluid flows. Commun. Numer. Methods. Eng. 15 (1), 9–18.

Pope, S., 2000. Turbulent Flows. Cambridge University Press, Cambridge, p. xxxiv+771.

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

Rebholz, L., Kim, T., Byon, Y., 2017a. On an accurate α model for coarse mesh turbulent channel flow simulation. Appl. Math. Model. 43, 139–154.

Rebholz, L., Zerfas, C., Zhao, K., 2017b. Global in time analysis and sensitivity analysis for the reduced NS-α model of incompressible flow. J. Math. Fluid Mech. 19 (3), 445–467.

Roelofs, F., Gopala, V.R., Chandra, L., Viellieber, M., Class, A., 2012. Simulating fuel assemblies with low resolution CFD approaches. Nucl. Eng. Des. 250, 548–559.

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

Sagaut, P., 2006. Large Eddy Simulation for Incompressible Flows, third ed. In: Scientific Computation, Springer-Verlag, Berlin, p. xxx+556.

San, O., Pawar, S., Rasheed, A., 2022. Variational multiscale reinforcement learning for discovering reduced order closure models of nonlinear spatiotemporal transport systems. Sci. Rep. 12 (1), 17947.

Stabile, G., Hijazi, S., Mola, A., Lorenzi, S., Rozza, G., 2017. POD-Galerkin reduced order methods for CFD using finite volume discretisation: vortex shedding around a circular cylinder. Commun. Appl. Ind. Math. 8 (1), 210–236.

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

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

Tsai, P.H., Fischer, P., 2022. Parametric model-order-reduction development for unsteady convection. Front. Phys. 711.

Vergari, L., Cammi, A., Lorenzi, S., 2020. Reduced order modeling approach for parametrized thermal-hydraulics problems: inclusion of the energy equation in the POD-FV-ROM method. Prog. Nucl. Energy 118, 103071.

Vergari, L., Cammi, A., Lorenzi, S., 2021. Reduced order modeling for coupled thermal-hydraulics and reactor physics problems. Prog. Nucl. Energy 140, 103899.

Volkwein, S., 2013. Proper Orthogonal Decomposition: Theory and Reduced-Order Modelling. In: Lecture Notes, University of Konstanz, http://www.math.uni-konstanz.de/numerik/personen/volkwein/teaching/POD-Book.pdf.

Wang, Z., Akhtar, I., Borggaard, J., Iliescu, T., 2012. Proper orthogonal decomposition closure models for turbulent flows: A numerical comparison. Comput. Methods Appl. Mech. Engrg. 237–240, 10–26.

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

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