fluids

MDPI

Article

Lagrangian Reduced Order Modeling Using Finite Time Lyapunov Exponents

Xuping Xie [1], Peter J. Nolan [2], Shane D. Ross [3], Changhong Mou [4] and Traian Iliescu [4,*]

[1] Courant Institute of Mathematical Sciences, New York University, New York, NY 10012, USA; xxie@nyu.edu
[2] Engineering Mechanics Program, Virginia Tech, Blacksburg, VA 24061, USA; pnolan86@vt.edu
[3] Department of Aerospace and Ocean Engineering, Virginia Tech, Blacksburg, VA 24061, USA; sdross@vt.edu
[4] Department of Mathematics, Virginia Tech, Blacksburg, VA 24061, USA; cmou@vt.edu
* Correspondence: iliescu@vt.edu

Received: 21 September 2020; Accepted: 17 October 2020; Published: 23 October 2020

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

Keywords: Lagrangian reduced order model; Lagrangian inner product; quasi-geostrophic equations; finite time Lyapunov exponent

1. Introduction

Projection-based reduced order models (ROMs) have been successful in the numerical simulation of fluid flows [1–6]. To approximate the dynamics of a given flow variable $\boldsymbol{u}$, the ROM strategy proceeds as follows: (i) Collect the snapshots $\{\boldsymbol{u}^1, \ldots, \boldsymbol{u}^M\}$, which are full order model (FOM) solutions at M different time instances. (ii) Construct the snapshot correlation matrix C using the inner product $(\cdot, \cdot)$:

$$C_{ij} = (\boldsymbol{u}^i, \boldsymbol{u}^j). \tag{1}$$

(iii) Solve the eigenvalue problem for C to obtain the ROM modes $\{\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_R\}$, which represent the recurrent spatial structures in the flow. (iv) Choose the dominant modes $\{\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_r\}$, $r \leq R$, as basis functions for the ROM. (v) Use a Galerkin truncation $\boldsymbol{u}_r = \sum_{j=1}^{r} a_j \boldsymbol{\varphi}_j$. (vi) Replace $\boldsymbol{u}$ with $\boldsymbol{u}_r$ in the underlying equations. (vii) Use a Galerkin projection of the PDE obtained in step (vi) onto the ROM space $\mathbf{X}^r := \text{span}\{\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_r\}$ to obtain a low-dimensional dynamical system, which represents the ROM. (viii) In an offline stage, compute the ROM operators. (ix) In an online stage, repeatedly use the ROM (for various parameter settings and/or longer time intervals). The low-dimensional ROMs can decrease the computational cost of traditional FOMs by orders of magnitude. ROMs, however, can be inaccurate in the numerical simulation of complex flows [3,5]. There are two main approaches to increasing ROM accuracy:

The first approach is to *improve the model*, that is, to add new terms to the standard projection-based ROM. Classical examples are ROM closure (see, e.g., References [7–12]) and ROM stabilization (see, e.g., References [13–16]). We will not follow this approach in this paper.

The second approach to improving the ROM accuracy is to *improve the ROM basis*, that is, to construct ROM bases that yield more accurate ROMs. One of the earliest examples in this class is the H^1-basis proposed in Reference [17], in which the H^1 inner product is used instead of the standard L^2 inner product to construct the ROM basis in order to increase the ROM stability. Similarly, an enstrophy-based ROM for rotational flows was proposed in Reference [18], in which the inner product is defined for vorticity instead of velocity. Other examples in this class are the ROM bases proposed for compressible flows [17,19–21], which use new inner products and different flow variables to construct the ROM basis (see Reference [22] for recent work on magnetohydrodynamics). Improved ROM bases were also proposed for data assimilation [23]. The inner products used to define these improved ROM bases are Eulerian inner products, that is, they are defined only for Eulerian data. To our knowledge, there are only a few Lagrangian inner products, that is, inner products that are defined on both Eulerian and Lagrangian data, that have been recently proposed. In References [24,25], the authors proposed inner products that are defined for velocity (which is an Eulerian variable) and the Lagrangian mesh coordinates (which are Lagrangian variables).

In this paper, we use the second strategy to improve the ROM accuracy, that is, we propose improved ROM bases. Specifically, we propose new Lagrangian inner products that utilize both Eulerian and Lagrangian data. In the new Lagrangian inner products, Lagrangian data steers the resulting Lagrangian ROM basis toward an accurate approximation of Lagrangian quantities, whereas Eulerian data helps the Lagrangian ROM basis yield an accurate approximation of Eulerian quantities. We emphasize that the Lagrangian inner products that we propose are different from the Lagrangian inner products in References [24,25]. As Lagrangian data, we use the finite time Lyapunov exponents (FTLE) field, whereas References [24,25] use the Lagrangian mesh coordinates. To construct the new Lagrangian ROMs, we utilize the new Lagrangian inner products, the resulting Lagrangian ROM bases, and the Galerkin projection. In the numerical simulation of the quasi-geostrophic equations [26–29] (which model large scale ocean circulation), the new Lagrangian ROMs are orders of magnitude more accurate than standard Eulerian ROMs (i.e., ROMs that use standard Eulerian data and inner products to build the ROM bases). Furthermore, the new Lagrangian ROMs are more accurate than the standard Eulerian ROMs in approximating not only Lagrangian fields (e.g., the finite time Lyapunov exponent (FTLE)), but also Eulerian fields (e.g., the streamfunction).

For complex nonlinear systems, it is well known that the low-dimensional ROMs generally need to be equipped with a closure model (see, e.g., References [7–12,30–32]) or a stabilization mechanism (see, e.g., References [13–15]) to model the effect of the discarded ROM modes. We emphasize, however, that we investigate the new Lagrangian ROMs without any closure or stabilization (a challenging test) in order to separate the ROM closure problem from the ROM basis generation, which is the main focus of our paper. This allows us to conclude that the orders of magnitude increase in the new Lagrangian ROMs' accuracy over the standard Eulerian ROMs' accuracy is entirely due to the new Lagrangian inner products used to build the Lagrangian ROMs' bases. Of course, we envision that using closure modeling in addition to the novel Lagrangian inner product will increase even further the Lagrangian ROMs' accuracy.

The rest of the paper is organized as follows: In Section 2, we propose the novel Lagrangian inner products and construct the new Lagrangian ROMs. In Section 3, for the quasi-geostrophic equations, we show that the new Lagrangian ROMs increase the numerical accuracy of standard Eulerian ROMs by orders of magnitude. Finally, in Section 4, we present conclusions and outline future research directions.

2. Lagrangian Reduced Order Models

In this section, we propose two new *Lagrangian ROMs*, which we build as follows: First, we use the quasi-geostrophic equations (QGE) as a mathematical model (although general models, for example, the Boussinesq and Navier-Stokes equations, could be used instead). Next, we perform numerical simulations to generate the QGE velocity field (which is Eulerian data) and the QGE finite time Lyapunov exponent (FTLE) field (which is Lagrangian data). (We note that other Lagrangian fields could be used instead of the FTLE field.) Finally, we propose two new *Lagrangian inner products* that use both Eulerian and Lagrangian data to construct new Lagrangian ROM bases, which yield the new Lagrangian ROMs. For comparison purposes, we also outline standard Eulerian ROMs [33–37], which use only Eulerian data (i.e., the velocity field) to generate the ROM basis. In Section 3, we compare the new Lagrangian ROMs with the standard Eulerian ROM in the numerical simulation of the QGE.

The QGE [26–29] are written as the following PDE:

$$\frac{\partial \omega}{\partial t} + J(\omega, \psi) - Ro^{-1}\frac{\partial \psi}{\partial x} = Re^{-1}\,\Delta\omega + Ro^{-1}F\,, \tag{2a}$$

$$\omega = -\Delta\psi\,, \tag{2b}$$

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

$$\boldsymbol{v} = \left(\frac{\partial \psi}{\partial y}, -\frac{\partial \psi}{\partial x}\right). \tag{3}$$

Details regarding the parameters and nondimensionalization of the QGE (2) are given in, for example, References [34,35,38–40].

2.1. Finite Time Lyapunov Exponents (FTLE) Computation

Next, we briefly describe the calculation of the FTLE field (see, e.g., Reference [41] for details). Given a velocity field $\boldsymbol{v}(\boldsymbol{x}, t)$ (e.g., the QGE velocity field (3)), the trajectories are obtained from the solutions of the ODE system $\dot{\boldsymbol{x}} = \boldsymbol{v}(\boldsymbol{x}, t)$. Each trajectory $\boldsymbol{x}(t; t_0, \boldsymbol{x}_0)$ is a function of time, but it also depends on the initial position $\boldsymbol{x}_0$ and the initial time t_0. For a given initial time t_0 and a given final time t, the flow map is the function

$$\boldsymbol{x}_0 \mapsto \phi_{t_0}^{t}(\boldsymbol{x}_0) = \boldsymbol{x}(t; t_0, \boldsymbol{x}_0)\,. \tag{4}$$

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

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

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

$$C\,(\boldsymbol{x}, t_0, T) = D\phi_{t_0}^{t_0+T}\,(\boldsymbol{x})^{\intercal}\,D\phi_{t_0}^{t_0+T}\,(\boldsymbol{x})\,. \tag{5}$$

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

$$\max \|\delta \boldsymbol{x}(t_0+T)\| = \sqrt{\mu_{\max}\left(C\left(\boldsymbol{x}, t_0, T\right)\right)}\, \|\delta \boldsymbol{x}(t_0)\| \,, \tag{6}$$

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

$$\lambda_{t_0}^{T}(\boldsymbol{x}) = \frac{1}{|T|} \ln \sqrt{\mu_{\max}\left(C\left(\boldsymbol{x}, t_0, T\right)\right)}. \tag{7}$$

Following Reference [41], we note that $\mu_{\max}\left(C\left(\boldsymbol{x}, t_0, T\right)\right) > 1$, so the FTLE in (7) is always positive.

2.2. Eulerian Reduced Order Model (E-ROM)

To generate the ROM basis for the standard Eulerian ROM, we use the proper orthogonal decomposition (POD) [3,5]. We emphasize, however, that the novel Lagrangian ROMs can be used with other ROM bases [1,2,4,6,33,42]. The POD starts by collecting the snapshots $\{\omega_h^1, \ldots, \omega_h^M\}$, which are, for example, finite element (FE) approximations of the vorticity in the QGE (2) at M different time instances. The POD seeks a low-dimensional basis that approximates the snapshots optimally with respect to a certain norm. Probably the most popular inner product is the L^2 inner product :

$$\Big(\omega_1, \omega_2\Big) = \int_{\Omega} \omega_1(\boldsymbol{x})\, \omega_2(\boldsymbol{x})\, d\boldsymbol{x}\,. \tag{8}$$

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

$$Y^T M_h Y \widetilde{\varphi}_j = \widetilde{\lambda}_j \widetilde{\varphi}_j, \quad j = 1, \ldots, N, \tag{9}$$

where Y denotes the snapshot matrix, whose columns correspond to the FE coefficients of the snapshots, M_h denotes the FE mass matrix, and N is the dimension of the FE space. The eigenvalues are real and non-negative, so they can be ordered as follows: $\widetilde{\lambda}_1 \geq \widetilde{\lambda}_2 \geq \ldots \geq \widetilde{\lambda}_R \geq \widetilde{\lambda}_{R+1} = \ldots = \lambda_N = 0$. The POD vorticity basis $\{\varphi_j\}_{j=1}^r$ are obtained from the eigenfunctions in (9) that correspond to the first $r \leq R$ largest eigenvalues. Thus, the ROM vorticity space is defined as $X^r := \mathrm{span}\{\varphi_1, \ldots, \varphi_r\}$. We follow References [34,35] and define the POD streamfunction basis as the normalized functions $\{\phi_j\}_{j=1}^r$, which are chosen such that

$$-\Delta \phi_j = \varphi_j\,, \quad j = 1, \ldots, r\,. \tag{10}$$

The ROM approximations of the vorticity and streamfunction are

$$\omega_r(\boldsymbol{x}, t) = \sum_{j=1}^{r} a_j(t)\, \varphi_j(\boldsymbol{x})\,, \qquad \psi_r(\boldsymbol{x}, t) = \sum_{j=1}^{r} a_j(t)\, \phi_j(\boldsymbol{x})\,, \tag{11}$$

where $\{a_j(t)\}_{j=1}^r$ are the sought time-varying ROM coefficients. We emphasize that, with the choices in (10) and (11), once the coefficients a_j are determined from (2a), Equation (2b) is automatically satisfied. Replacing the vorticity ω by ω_r in the QGE (2a) and then using a Galerkin projection onto X^r, we obtain the *Eulerian ROM (E-ROM)* for the QGE: $\forall\, i = 1, \ldots, r$,

$$\begin{aligned} \left(\frac{\partial \omega_r}{\partial t}, \varphi_i\right) + \Big(J(\omega, \psi), \varphi_i\Big) - Ro^{-1}\left(\frac{\partial \psi}{\partial x}, \varphi_i\right) \qquad &+ Re^{-1}\Big(\nabla \omega_r, \nabla \varphi_i\Big) \\ &= Ro^{-1}\Big(F, \varphi_i\Big)\,. \end{aligned} \tag{12}$$

The E-ROM (12) yields the following autonomous dynamical system for the vector of time coefficients, $\mathbf{a}(t)$:

$$\dot{\mathbf{a}} = \mathbf{b} + \mathbf{A}\,\mathbf{a} + \mathbf{a}^\top\,\mathbf{B}\,\mathbf{a}, \tag{13}$$

where $\mathbf{b}$, $\mathbf{A}$, and $\mathbf{B}$ correspond to the constant, linear, and quadratic terms in the numerical discretization of the QGE (2), respectively. The finite dimensional system (13) can be written componentwise as follows: For all $i = 1, \ldots, r$,

$$\dot{a}_i(t) = b_i + \sum_{m=1}^{r} A_{im} a_m(t) + \sum_{m=1}^{r}\sum_{n=1}^{r} B_{imn}\, a_m(t)\, a_n(t), \tag{14}$$

where

$$b_i = Ro^{-1}\left(F, \varphi_i\right), \tag{15}$$

$$A_{im} = Ro^{-1}\left(\frac{\partial \phi_m}{\partial x}, \varphi_i\right) - Re^{-1}\left(\nabla\varphi_m, \nabla\varphi_i\right), \tag{16}$$

$$B_{imn} = -\left(J(\varphi_m, \phi_n), \varphi_i\right). \tag{17}$$

The E-ROM (12) has been investigated in the numerical simulation of the QGE (2) (see, e.g., References [34–37]), where it was shown that it can decrease the computational cost of standard algorithms by orders of magnitude. However, the numerical simulations in References [34,35] have also shown that a low-dimensional E-ROM is not able to produce accurate approximations of standard Eulerian quantities, such as the streamfunction and the velocity fields. (We will also show in Section 3 that the standard E-ROM (12) produces inaccurate approximations of Lagrangian quantities, such as the FTLE field.) The E-ROM's numerical inaccuracy in References [34,35] is due to the lack of a closure model [11,12], that is, a model for the effect of the discarded ROM modes. Thus, to alleviate its numerical inaccuracy, in References [34,35] the standard E-ROM (12) was supplemented with a stabilizing mechanism that yielded relatively accurate results.

In the next section, we pursue a fundamentally different research avenue to improve the standard E-ROM's numerical accuracy. Instead of modifying the ROM (i.e., adding a closure model, as done in Reference [35]), we propose using a novel set of basis functions that combine Lagrangian and Eulerian data.

2.3. Lagrangian ROMs

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

The main tools that we use to construct the new Lagrangian ROMs for FTLE computation are two novel Lagrangian inner products, which are fundamentally different from the standard L^2 inner product (8) used to develop the Eulerian ROM (i.e., the E-ROM (12)). These new inner products are *Lagrangian inner products* $(\cdot,\cdot)_{FTLE}$, which aim at including both Eulerian data (i.e., the vorticity field) and Lagrangian data (i.e., the FTLE field) in the ROM basis generation. We emphasize that adding FTLE data to the snapshots is not redundant, since this yields a significantly different ROM basis (see snapshot difference quotients used for E-ROM (12) in Reference [43] for similar behavior in an Eulerian setting). The two new Lagrangian inner products generate ROM basis functions that are different from the standard E-ROM modes, which are built with the standard L^2 inner product (see Figure 3). These two new bases yield two new Lagrangian ROMs, which we present in

Sections 2.3.1 and 2.3.2. In Section 3, in the numerical simulation of the QGE, we show that these two novel Lagrangian ROMs are more accurate than the standard E-ROM (12).

2.3.1. α-ROM

The first Lagrangian inner product that we propose is

$$\Big((\omega_1, \lambda_1) \, , \, (\omega_2, \lambda_2) \Big)_{FTLE} = \int_{\Omega} \omega_1(\boldsymbol{x}) \, \omega_2(\boldsymbol{x}) + \alpha \, \lambda_1(\boldsymbol{x}) \, \lambda_2(\boldsymbol{x}) \, d\boldsymbol{x} \, , \tag{18}$$

where ω_1 and ω_2 are vorticity fields and λ_1 and λ_2 are FTLE fields. Thus, the Lagrangian inner product (18) combines Lagrangian data (λ_1, λ_2) with Eulerian data (ω_1, ω_2). The parameter α in (18) is a *weighting parameter* that measures the Lagrangian data's contribution to the inner product: When $\alpha = 0$, the Lagrangian data does not play any role, so the inner product (18) is the standard L^2 inner product (8) used to build the standard E-ROM (12). When $\alpha > 0$, the Lagrangian data plays a significant role: The higher the α value, the more important the Lagrangian data contribution to the inner product (18).

Remark 1 (Nondimensional inner product)**.** *The Lagrangian inner product* (18) *combines data (the vorticity field and the FTLE field) that has the same dimensional units (i.e., inverse time). Thus, the two different types of variables (ω and λ) in* (18) *can be added together. Furthermore, the QGE* (2) *used to compute ω and λ are nondimensionalized, so the two types of variables could be added even if they did not have the same dimensional units. Finally, if the QGE were left in their original dimensional form, we would need to scale the variables ω and λ appropriately.*

We use the new Lagrangian inner product (18) to generate the ROM basis for a new Lagrangian ROM. First, we collect snapshots that consist of both vorticity and FTLE approximations. (Note that this is different from the standard E-ROM (12) basis generation, where only vorticity snapshots were collected.) Then, we construct the new Lagrangian ROM basis that approximates the snapshots optimally with respect to the Lagrangian norm

$$\left(\|\omega\|^2 + \alpha \, \|\lambda\|^2 \right)^{\frac{1}{2}} \, . \tag{19}$$

(Again, we note that this is different from the approach used for the standard E-ROM (12), which utilizes the norm $\|\omega\|$.) Finally, from the resulting ROM basis functions, we only use their vorticity components in the ROM (12).

The novel Lagrangian ROM for the FTLE computation is the ROM (12) in which the ROM basis is generated by using the new Lagrangian inner product (18) instead of the standard L^2 inner product (8) used to build the E-ROM (12). In what follows, we will denote by *α-ROM* the resulting new Lagrangian ROM. Since the new α-ROM includes FTLE data (through both the snapshots and the Lagrangian inner product (18)), we expect it to yield a more accurate FTLE approximation than the standard E-ROM (12), which does not explicitly include FTLE data.

2.3.2. λ-ROM

The second Lagrangian inner product that we propose is

$$\left(\omega_1 \, , \, \omega_2 \right)_{FTLE} = \int_{\Omega} \langle \lambda \rangle(\boldsymbol{x}) \, \omega_1(\boldsymbol{x}) \, \omega_2(\boldsymbol{x}) \, d\boldsymbol{x} \, , \tag{20}$$

where $\langle \lambda \rangle$ is the *time average* of the FTLE field, λ. The Lagrangian inner product (20) is similar to the Lagrangian inner product (18) in that both use Lagrangian data (i.e., the FTLE field). We note, however, that the way in which Lagrangian and Eulerian data is combined in the two inner products is different: the Lagrangian data is added to the Eulerian data in (18), whereas in (20) the Lagrangian data is first time averaged and then it is used as a scaling factor for the Eulerian data. We also note that in the numerical investigation in Section 3, we use a QGE setting in which the time averages of the streamfunction and FTLE fields play an important role. Thus, we expect the Lagrangian inner product (20) to produce accurate results in that setting. Finally, as noted in Remark 1, the data used in the Lagrangian inner product (20) is nondimensional, so the definition (20) is appropriate.

When we use the Lagrangian inner product (20) to generate the Lagrangian ROM basis, these basis functions approximate the snapshots optimally with respect to the norm

$$\left(\int_{\Omega} \langle \lambda \rangle (\boldsymbol{x}) \, \omega^2(\boldsymbol{x}) \, d\boldsymbol{x} \right)^{\frac{1}{2}} . \tag{21}$$

Note that, by definition, the FTLE field (7) is always positive. Thus, the Lagrangian inner product (20) and the associated norm (21) are well defined.

The second new Lagrangian ROM for the FTLE computation is the ROM (12) in which the ROM basis is generated by using the new Lagrangian inner product (20) instead of the standard L^2 inner product (8) used to build the E-ROM (12). In what follows, we will denote by *λ-ROM* the resulting new Lagrangian ROM. Again, since the new λ-ROM includes FTLE information, (through both the snapshots and the Lagrangian inner product (20)), we expect it to yield a more accurate FTLE approximation than the E-ROM (12), which does not explicitly include FTLE information.

2.4. Previous Relevant Work

To our knowledge, there is only little work on reduced order modeling for the FTLE calculation [44–46]. We emphasize that the Lagrangian ROMs proposed in this paper are fundamentally different from the ROMs used in References [44,45], which are Eulerian ROMs. The Lagrangian ROMs are also different from the ROM used in Reference [46], since the FTLE field is used in Reference [46] only to choose the number, not the actual form of ROM modes, whereas we explicitly use the FTLE field to define the FTLE inner product (18) and, thus, to construct the ROM basis.

The Lagrangian ROMs proposed in this paper are related to ROMs that aim at tackling the challenges posed by transport-dominated problems, for example, wave-like phenomena, moving interfaces, and moving shocks. The ROMs for transport-dominated problems are surveyed in References [24,25,47] and include development of, for example, local bases [48,49], domain decomposition [50], adaptivity [51,52], symmetry and self similarity transformations [53,54], approximated Lax pairs [55], and transport maps [47,56–62].

There are also connections between the new Lagrangian ROMs and the ROMs that preserve Lagrangian structure [63,64] (see also References [65–69] for ROMs that preserve Hamiltonian structure) as well as the energy-conserving ROMs for the Navier-Stokes equations [34,70–76].

We also note that including Lagrangian information to build the ROM basis is similar in spirit to the difference quotients used in References [43,77] and collecting snapshots for the nonlinear terms in the Empirical Interpolation Method (EIM) [78] and its discrete version, the Discrete Empirical Interpolation Method (DEIM) [79]. Indeed, in all these methods, one collects linear combinations of the snapshots. Of course, this does not change the rank of the snapshot matrix, but can change the ordering of its singular values and, thus, yield different ROM bases. Adding Lagrangian information to the set of snapshots is similar in spirit: Although the FTLE snapshots are computed from Eulerian data, using them together with the Eulerian snapshots yields ROM basis functions that are different from the Eulerian ROM basis functions.

3. Numerical Results

In Section 2.3, we proposed two new Lagrangian ROMs (i.e., the α-ROM and the λ-ROM) for the numerical simulation of the FTLE field. For clarity, Table 1 summarizes the inner products used to build the basis functions of the two new Lagrangian ROMs, as well as the standard Eulerian ROM (i.e., the E-ROM (12)). In this section, we perform a numerical investigation of the two new Lagrangian ROMs. To separate the ROM closure modeling from the ROM basis generation, we investigate the two new Lagrangian ROMs without any closure model or stabilization mechanism.

Table 1. The new Lagrangian reduced order models (ROMs) (α-ROM and λ-ROM), the standard Eulerian ROM (E-ROM), and the inner products used to construct their bases.

Inner Product	ROM	Type
Equation (8)	E-ROM	Eulerian
Equation (18)	α-ROM	Lagrangian
Equation (20)	λ-ROM	Lagrangian

In this section, we investigate the Lagrangian α-ROM and λ-ROM in the numerical simulation of the velocity and FTLE fields of the QGE (2). For comparison purposes, we also test the standard Eulerian ROM (i.e., the E-ROM (12)). As a benchmark, we use the full order model (FOM), which is outlined in Algorithm 1:

Algorithm 1 Full Order Model (FOM).

```
(1) Compute high resolution streamfunction $\psi^{FOM}$ on $[T_{min}, T_{max}]$.
(2) Use $\psi^{FOM}$ in (1) and formula (3) to compute high resolution velocity $\mathbf{v}^{FOM}$ on $[T_{min}, T_{max}]$.
(3) Use $\mathbf{v}^{FOM}$ in (2) to calculate (see § 2.1) high resolution FTLE field $\lambda^{FOM}$ on $[T_{min}, T_{max}]$.
```

In the numerical investigation of the three ROMs (i.e., α-ROM, λ-ROM, and E-ROM), we use Algorithm 2. We also use two types of regimes: (i) the *reconstructive* regime; and (ii) the *predictive* regime. The two regimes have fundamentally different goals: The reconstructive regime is an easier test, in which the ROM is validated on the same time interval as the time interval used to train the ROM. The predictive regime is a harder test case, in which the ROM is trained on a short time interval, for example, $\left[T_{min}, \dfrac{T_{max}}{2}\right]$ and validated on a longer time interval $[T_{min}, T_{max}]$.

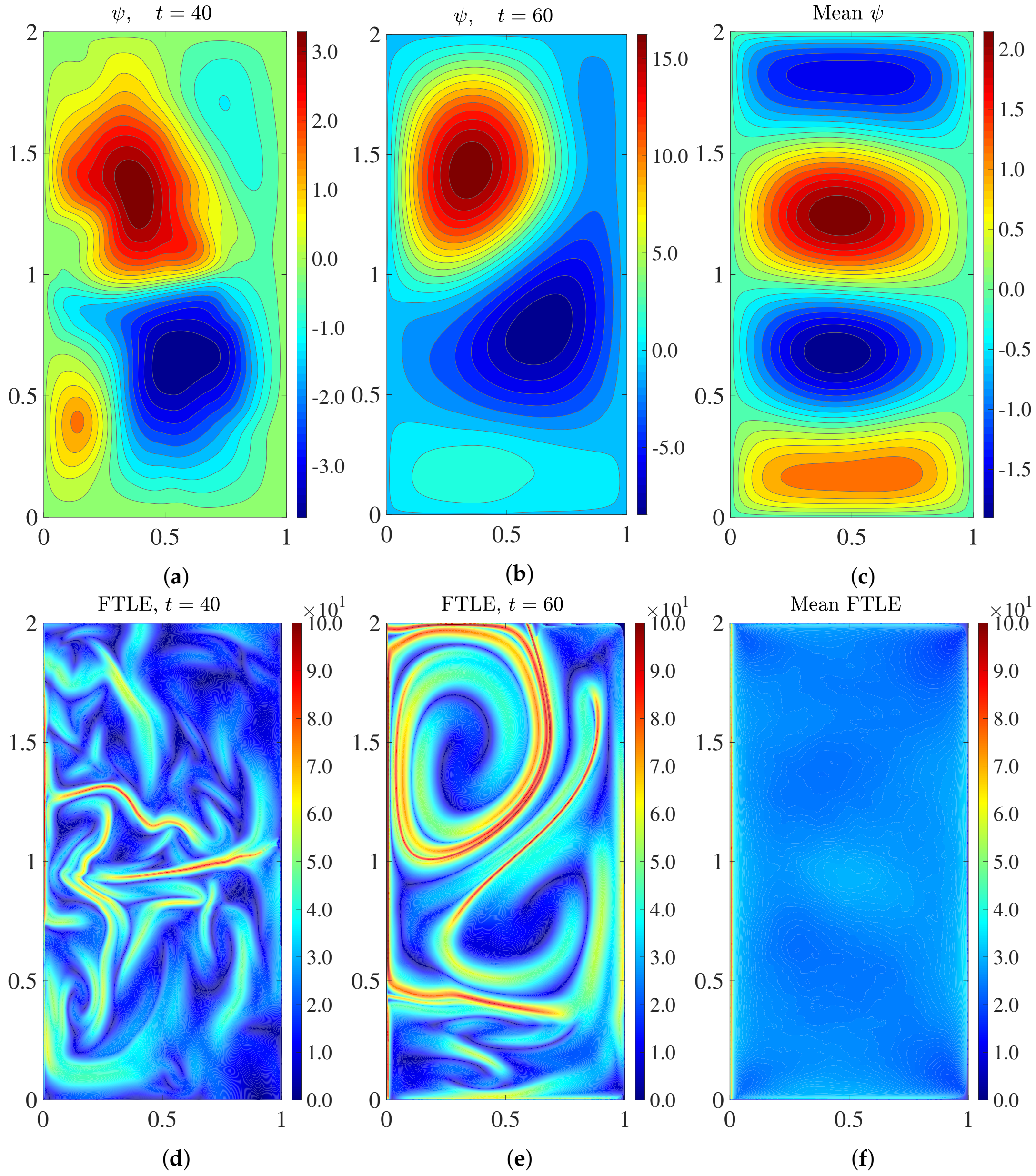
Algorithm 2 Reduced Order Model (ROM).

```
(1) Compute high resolution streamfunction $\psi^{FOM}$ on $[T_{min}, T_{max}]$.
(2) Use $\psi^{FOM}$ in (1) and formula (3) to compute high resolution velocity $\mathbf{v}^{FOM}$ on $[T_{min}, T_{max}]$.
(3) Use $\mathbf{v}^{FOM}$ in (2) on $[T_{min}, T_{max}]$ to construct Lagrangian ROMs ($\alpha$-ROM and $\lambda$-ROM) and Eulerian
    ROM (E-ROM).
(4) Use ROMs in (3) to compute low resolution ROM streamfunction $\psi^{ROM}$ on $[T_{min}, T_{max}]$.
(5) Use low resolution streamfunction $\psi^{ROM}$ in (4) and formula (3) to compute low resolution
    velocity $\mathbf{v}^{ROM}$ on $[T_{min}, T_{max}]$.
(6) Use low resolution velocity $\mathbf{v}^{ROM}$ in (5) to calculate low resolution ROM-FTLE field $\lambda^{ROM}$ on
    $[T_{min}, T_{max}]$.
```

3.1. Test Problem Setup

As a test problem in our numerical investigation, we consider the QGE (2) with a symmetric double-gyre wind forcing given by $F = \sin\big(\pi\,(y-1)\big)$, which yields a four-gyre circulation in the time mean. This test problem has been used in numerous studies (see, e.g., References [34,35,39,40,80–85]) as a simplified model for more realistic ocean dynamics.

We emphasize that the four-gyre QGE test problem represents a significant challenge for standard numerical methods: Indeed, as shown in Reference [81], although a double-gyre wind forcing is used, the long term time-average yields a *four-gyre* pattern (see Figure 1). On realistic coarse meshes, classical numerical methods (e.g., finite element and finite volume methods) generally produce inaccurate approximations to this test problem. In particular, standard numerical discretizations fail to recover the correct four-gyre pattern (see, e.g., References [35,40]).



(**a**) (**b**) (**c**)

(**d**) (**e**) (**f**)

Figure 1. Streamfunction contour plots at $t = 40$ (plot (**a**)), $t = 60$ (plot (**b**)), and time-averaged (plot (**c**)). FTLE contour plots at $t = 40$ (plot (**d**)), $t = 60$ (plot (**e**)), and time-averaged (plot (**f**)). An FTLE movie is available at https://youtu.be/JXqdcBVfhMw.

In the QGE (2), we use the same parameters as those used in References [34,35,40,82]: $Re = 450$ and $Ro = 0.0036$. The spatial domain of the QGE is $[0, 1] \times [0, 2]$. In the FTLE field computation (7), we use $T = 0.05$.

3.2. Criteria

To investigate the numerical accuracy of the three ROMs (i.e., α-ROM, λ-ROM, and E-ROM), we compare the ROM results with the FOM results. To this end, we use two fundamentally different types of criteria:

The first type of criteria are *Eulerian criteria*. Specifically, we compute the L^2 norm of the time-averaged streamfunction errors between ψ^{FOM} obtained in Step (2) of Algorithm 1 and ψ^{ROM} obtained in Step (5) of Algorithm 2:

$$\left\| \frac{1}{M} \sum_{j=1}^{M} \psi^{FOM}(t_j) - \frac{1}{M} \sum_{j=1}^{M} \psi^{ROM}(t_j) \right\|_{L^2}^{2} . \tag{22}$$

In addition to the quantitative criterion (22), we are also using the following qualitative Eulerian criterion: We investigate whether the three ROMs can recover the four-gyre pattern of the time average of the streamfunction in Figure 1, which represents a challenging test for standard numerical methods at realistic low resolutions (see, e.g., References [35,40]).

The second type of criterion we use in our numerical investigation is a *Lagrangian criterion*. Specifically, we compute the L^2 norm of the the time-averaged FTLE errors between λ^{FOM} obtained in Step (3) of Algorithm 1 and λ^{ROM} obtained in Step (6) of Algorithm 2:

$$\left\| \frac{1}{M} \sum_{j=1}^{M} \lambda^{FOM}(t_j) - \frac{1}{M} \sum_{j=1}^{M} \lambda^{ROM}(t_j) \right\|_{L^2}^{2} . \tag{23}$$

3.3. ROM Snapshot Generation

For the FOM (see Algorithm 1) spatial discretization, we use a pseudospectral method that utilizes only odd Fourier modes (i.e., sines, since they automatically enforce the homogeneous boundary conditions), the 3/2s rule to resolve the aliasing problem in the nonlinearity, and a 257 × 513 spatial resolution [34]. For the FOM time discretization, we utilize a time step $\Delta t = 0.01$ and an explicit Runge-Kutta method (Tanaka-Yamashita, an order 7 method with an embedded order 6 method for error control) and an error tolerance of 1.0×10^{-8} in time with adaptive time refinement and coarsening [34]. These spatial and temporal discretizations yield numerical results that are similar to the fine resolution numerical results obtained in References [35,40]. In Figure 2, we plot the time evolution of the spatially averaged kinetic energy, $E(t)$. Figure 2 (see also figure 1 in Reference [35]) shows that the flow converges to a statistically steady state, after a short transient interval that ends around $t = 10$. Thus, in our numerical investigation, we follow References [34,35,40] and consider the FOM results only on $[T_{min}, T_{max}] = [10, 80]$. We compute the FTLE using the FOM results in the time window $[T_{min}, T_{max}] = [10, 80]$. In Figure 1, we display the instantaneous contour plot for the streamfunction field at $t = 40$ and $t = 60$. We emphasize that, although $t = 40$ and $t = 60$ are well within the statistically steady state regime, the flow displays a high degree of variability. Thus, the numerical approximation of this statistically steady regime remains challenging for the low resolution ROMs that we investigate in this section.

To generate the ROM basis (see Section 2.2), we follow References [34,35,40] and collect 701 snapshots in the time interval $[T_{min}, T_{max}] = [10, 80]$ (on which the statistically steady state regime is attained) at equidistant time intervals.

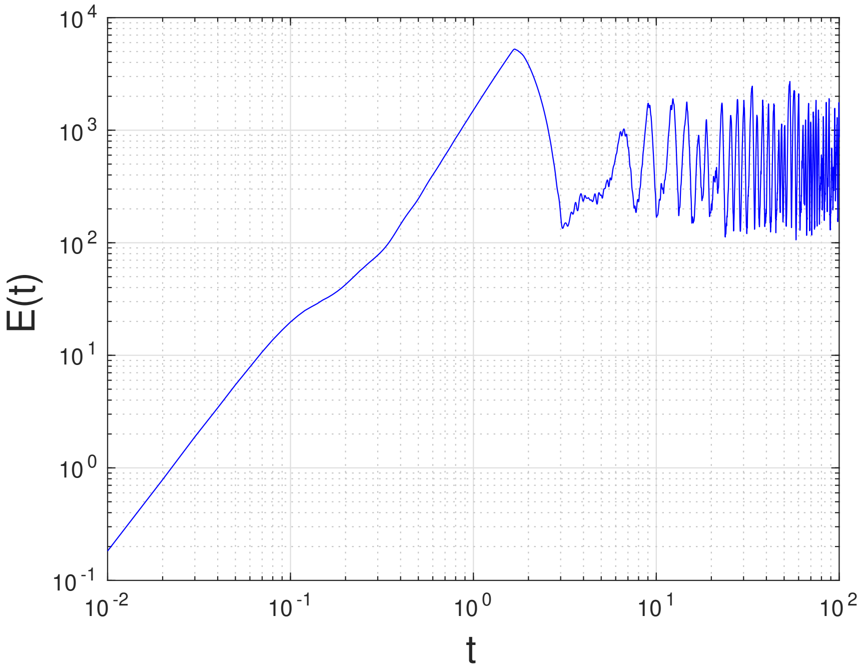


Figure 2. Time evolution of the spatially averaged kinetic energy of the FOM.

3.4. ROM Basis Investigation

The new Lagrangian ROM (i.e., α-ROM and λ-ROM) bases are fundamentally different from the standard E-ROM (12) basis. Indeed, the E-ROM basis is built only from Eulerian data (i.e., the vorticity ω) by using the standard L^2 inner product (8). On the other hand, the α-ROM and λ-ROM bases are constructed from Lagrangian data (i.e., the FTLE field λ) in addition to Eulerian data (i.e., the vorticity ω) by using the new Lagrangian inner product (18) and the new Lagrangian inner product (20), respectively.

To investigate whether the α-ROM and λ-ROM bases are different from the E-ROM (12) basis, in Figure 3 we display the ROM basis functions ψ_{10}, ψ_{20}, and ψ_{30} generated with the standard L^2 inner product (8) (i.e., the E-ROM basis functions), the new Lagrangian inner product (18) (i.e., the new α-ROM basis functions) with $\alpha = 10^4$, and the new Lagrangian inner product (20) (i.e., the new λ-ROM basis functions).

The α-ROM basis functions (second row of Figure 3) are completely different from the E-ROM basis functions (first row of Figure 3) for ψ_{10}, ψ_{20}, and ψ_{30}. The α-ROM basis functions are also completely different from the λ-ROM basis functions (third row of Figure 3). The λ-ROM basis functions (third row of Figure 3) are also different from the E-ROM basis functions (first row of Figure 3), although this time the differences are not as dramatic as before: there are large differences in ψ_{30}, moderate differences in ψ_{20}, and minor differences in ψ_{10}. Overall, the results in Figure 3 show that the new Lagrangian inner product (18), the new Lagrangian inner product (20), and the standard Eulerian L^2 inner product (8) generate completely different bases for the Lagrangian α-ROM, the Lagrangian λ-ROM, and the standard E-ROM, respectively. In the next section, we investigate which of these bases yields more accurate ROMs in the FTLE field computation.

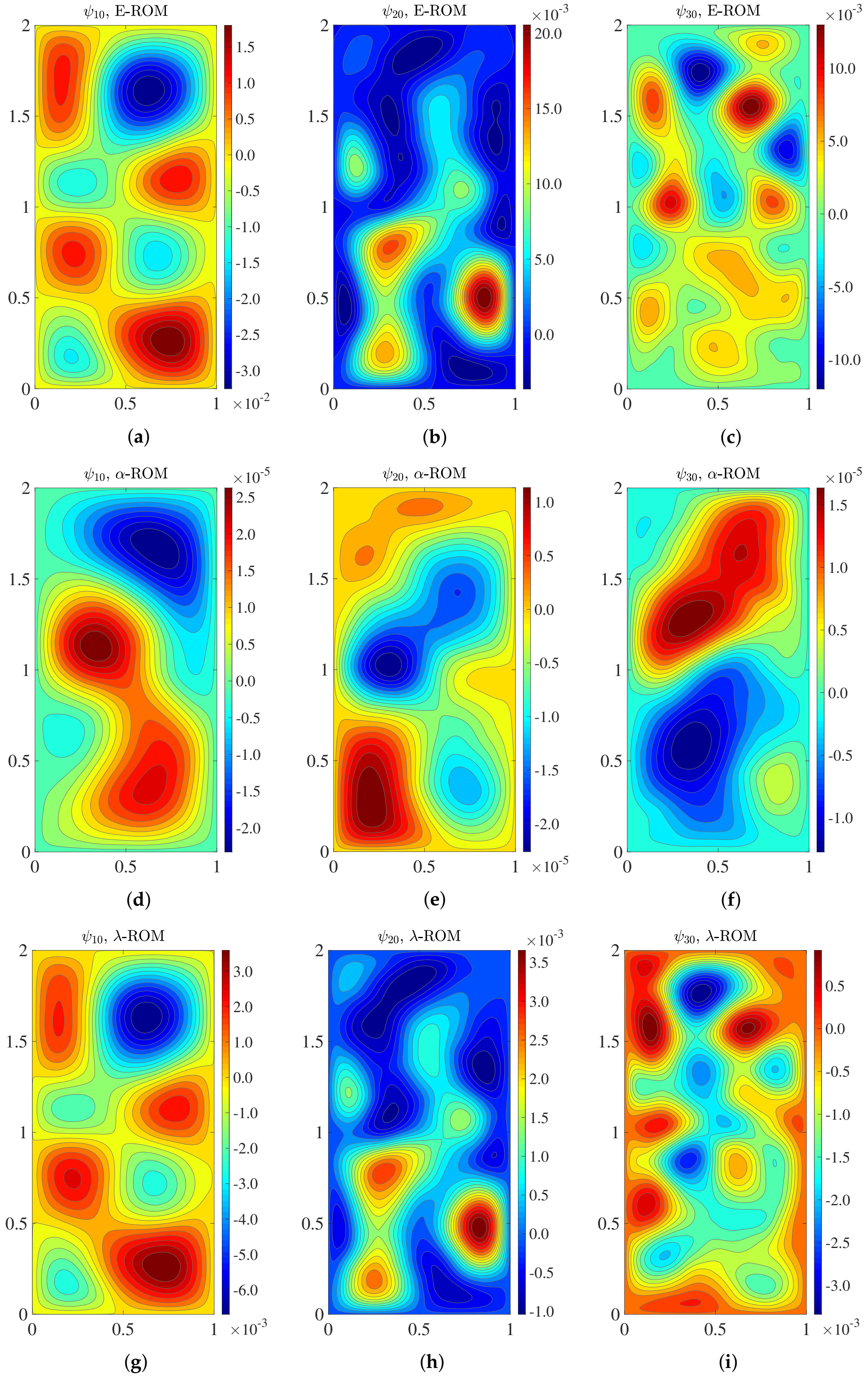


Figure 3. ROM basis functions ψ_{10} (plots (**a**,**d**,**g**)), ψ_{20} (plots (**b**,**e**,**h**)), and ψ_{30} (plots (**c**,**f**,**i**)) for the standard E-ROM (plots (**a**–**c**)), new Lagrangian α-ROM with $\alpha = 10^4$ (plots (**d**–**f**)), and new Lagrangian λ-ROM (plots (**g**–**i**)).

3.5. ROM Numerical Accuracy

In this section, we perform a numerical investigation of the accuracy of the two Lagrangian ROMs (i.e., α-ROM, λ-ROM). We only consider the effect of the basis functions on the ROM accuracy without using a ROM closure model or ROM stabilization mechanism, which is a challenging test.

We compare the Lagrangian α-ROM and λ-ROM accuracy with the standard E-ROM accuracy. As a benchmark for our comparison, we use the FOM results (Algorithm 1). In Section 3.5.1, we perform an Eulerian investigation of the three ROMs, that is, we investigate the ROMs' accuracy in approximating the streamfunction (which is an Eulerian quantity). In Section 3.5.2, we perform a Lagrangian investigation of the three ROMs, that is, we investigate the ROMs' accuracy in approximating the FTLE field (which is a Lagrangian quantity). For both the Eulerian and Lagrangian investigations, we consider both the reconstructive regime and the predictive regime.

In our numerical experiments, we use the following parameter values: For the α-ROM, we use $\alpha = 1, \alpha = 10^2, \alpha = 10^3$, and $\alpha = 10^4$. We choose this wide range of parameter values to elucidate the effect of the Lagrangian data on the new α-ROM. Indeed, the parameter α in (18) is a weighting parameter that measures the Lagrangian data's contribution to the inner product: The higher the α value, the more important the Lagrangian data contribution to the inner product (18). For all the ROM simulations, we use an RK4 time discretization with $\Delta t = 10^{-3}$. Finally, for all the ROMs, we utilize the following r values: $5, 10, 15, 20, 25, 30, 35, 40, 45$, and 50. We choose this wide range of values to clarify the effect of the ROM dimension on the ROM accuracy.

3.5.1. Eulerian Investigation

In this section, we perform an Eulerian investigation of the accuracy of the two Lagrangian ROMs (i.e., the α-ROM and λ-ROM). First, we consider the reconstructive regime and then the more challenging predictive regime. In both regimes, we use the two Eulerian criteria described in Section 3.2: (i) the quantitative Eulerian criterion (22), that is, the L^2 norm of the the time-averaged streamfunction errors between ψ^{FOM} obtained in Step (2) of Algorithm 1 and ψ^{ROM} obtained in Step (5) of Algorithm 2; and (ii) the qualitative Eulerian criterion based on the ability of the ROMs to recover the four-gyre pattern of the time average of the streamfunction in Figure 1 (which is a challenging test for standard numerical methods at realistic low resolutions [35,40]).

Reconstructive Regime:

For the reconstructive regime, we check whether the ROMs can reproduce the dynamics of the underlying system on the same time interval as that used to generate the ROM basis functions, that is, we validate the ROMs on the same time interval as the time interval used to train the ROM.

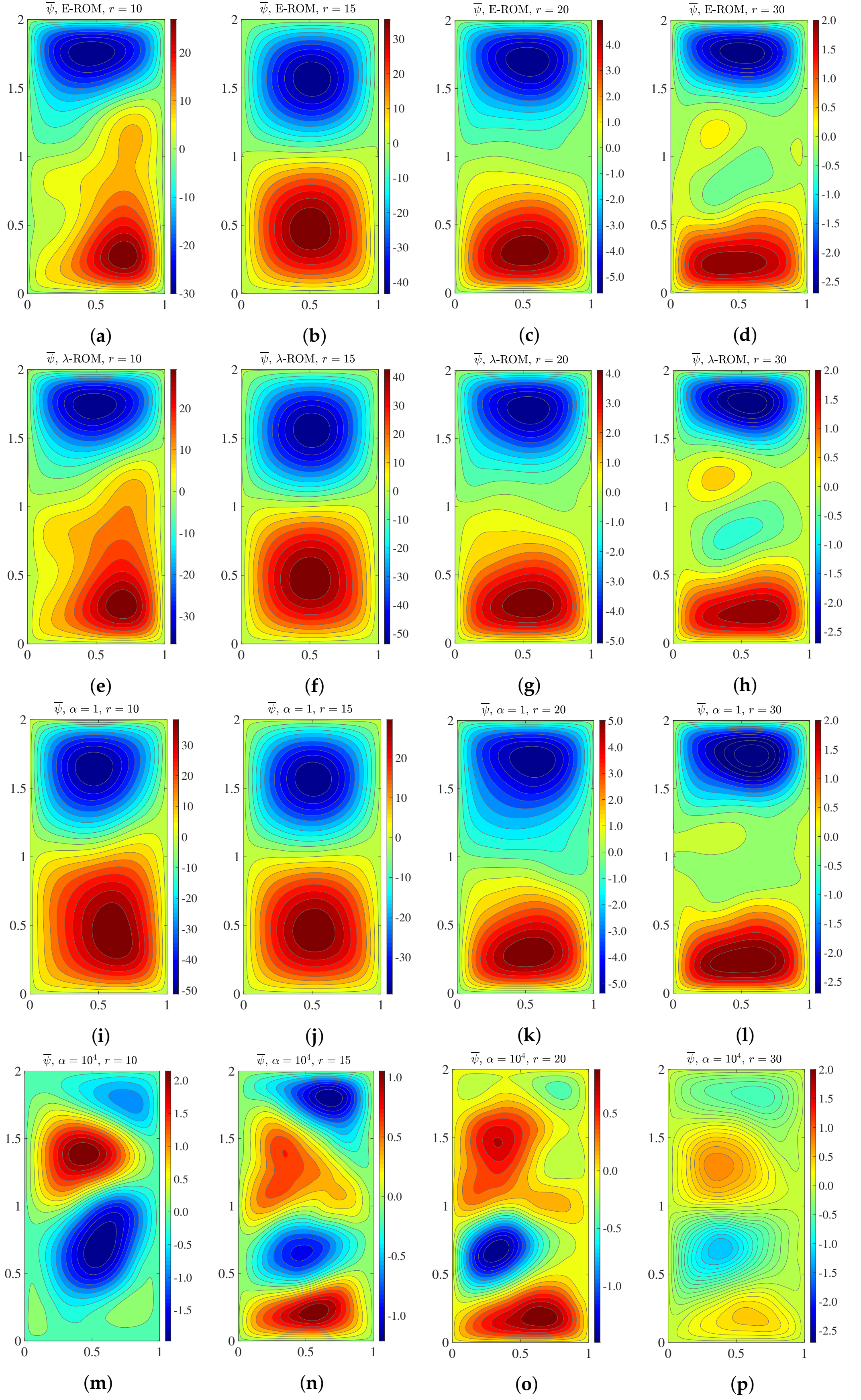
In Table 2, for different r values, we list the L^2 norm of the errors in the time-averaged streamfunction (22) for E-ROM, λ-ROM, and α-ROM with $\alpha = 1, \alpha = 10^2, \alpha = 10^3$, and $\alpha = 10^4$. These results yield the following conclusions: The E-ROM yields inaccurate results for low r values. As expected, the E-ROM results get better for large r values. The λ-ROM results are slightly worse than or similar to the E-ROM results for low r values and somewhat better for large r values. The results for α-ROM with $\alpha = 1$ are generally worse than the E-ROM results. The results for α-ROM with $\alpha = 10$ and $\alpha = 10^2$ are better than the E-ROM results for all r values except $r = 10$. The results for α-ROM with $\alpha = 10^3$ and $\alpha = 10^4$ are dramatically better than the E-ROM results: For example, for $r = 5$, the only ROM that yields acceptable results is the α-ROM with $\alpha = 10^4$; all other ROMs simply blow up (denoted by "N/A" in Table 2). Furthermore, for $r = 10$ and $r = 15$, the errors of α-ROM with $\alpha = 10^3$ and $\alpha = 10^4$ are *two orders of magnitude* lower than the E-ROM error. For the larger r values, the errors of α-ROM with $\alpha = 10^3$ and $\alpha = 10^4$ continue to be lower than the E-ROM errors, although (as expected) the differences decrease as the r values increase. As expected, when the r values increase, the results in Table 2 show that all the ROMs' errors converge until they reach a plateau around 5.0×10^{-1} (which is probably due to the ROM truncation error). Overall, the results in Table 2 show that *the Lagrangian α-ROM* yields *significantly more accurate results than the standard E-ROM and the Lagrangian λ-ROM*. In particular, for the small and medium r values, the Lagrangian α-ROM with high α values (i.e., $\alpha = 10^4$) performs the best, while for the large r values, the Lagrangian α-ROM with medium α values (i.e., $\alpha = 10^2$ and $\alpha = 10^3$) performs the best.

Table 2. Eulerian investigation, reconstructive regime: L^2 norm of the errors in the time-averaged streamfunction (22) for E-ROM (second column), λ-ROM (third column), and α-ROM for $\alpha = 1$ (fourth column), $\alpha = 10$ (fifth column), $\alpha = 10^2$ (sixth column), $\alpha = 10^3$ (seventh column), and $\alpha = 10^4$ (eighth column).

r	**E-ROM**	**λ-ROM**	$\alpha = 1$	$\alpha = 10$	$\alpha = 10^2$	$\alpha = 10^3$	$\alpha = 10^4$
5	N/A	N/A	N/A	N/A	N/A	N/A	1.0×10^1
10	3.6×10^2	4.0×10^2	7.9×10^2	6.2×10^3	6.5×10^2	2.9×10^0	1.2×10^0
15	1.8×10^2	5.5×10^2	4.1×10^2	1.3×10^1	1.1×10^1	1.1×10^0	2.3×10^0
20	1.3×10^1	8.0×10^0	1.5×10^1	4.2×10^0	2.7×10^0	1.1×10^0	1.1×10^0
25	3.3×10^0	2.7×10^0	4.7×10^0	3.0×10^0	2.1×10^0	3.9×10^{-1}	3.4×10^{-1}
30	2.8×10^0	2.0×10^0	2.5×10^0	2.5×10^0	1.1×10^0	5.9×10^{-1}	4.2×10^{-1}
35	1.5×10^0	9.3×10^{-1}	1.2×10^0	1.3×10^0	8.6×10^{-1}	5.1×10^{-1}	3.1×10^{-1}
40	1.0×10^0	6.1×10^{-1}	9.3×10^{-1}	9.9×10^{-1}	5.4×10^{-1}	4.6×10^{-1}	5.4×10^{-1}
45	5.4×10^{-1}	5.2×10^{-1}	7.0×10^{-1}	6.8×10^{-1}	5.1×10^{-1}	4.0×10^{-1}	7.0×10^{-1}
50	4.5×10^{-1}	5.0×10^{-1}	5.0×10^{-1}	6.2×10^{-1}	5.9×10^{-1}	6.0×10^{-1}	3.8×10^{-1}

Next, we use the qualitative Eulerian criterion to investigate the ability of the ROMs to recover the four-gyre pattern of the time average of the streamfunction in Figure 1. In Figure 4, for $r = 10, 15, 20,$ and 30, we plot the mean streamfunction for E-ROM, λ-ROM, and α-ROM with $\alpha = 1$ and $\alpha = 10^4$. These results yield the following conclusions: The E-ROM, the λ-ROM, and the α-ROM with a low α value (i.e., $\alpha = 1$) yield similar results: These ROMs cannot recover the four-gyre pattern for any of the four r values. However, the α-ROM with a large α value (i.e., $\alpha = 10^4$) yields dramatically better results: This ROM can clearly capture the four-gyre pattern for $r = 30$; for $r = 20$, the pattern is somewhat captured, although not as clearly as for $r = 30$; finally, for $r = 10$ and $r = 15$, only hints of the four-gyre pattern are present. Overall, the plots in Figure 4 show that the *Lagrangian α-ROM with a large α value* (i.e., $\alpha = 10^4$) *can capture the four-gyre pattern*, whereas the *standard E-ROM and the Lagrangian λ-ROM cannot*.

The results in Table 2 and Figure 4 consistently show that the *new Lagrangian α-ROM with large α values outperforms the standard E-ROM and the Lagrangian λ-ROM* with respect to the two Eulerian metrics used in this section. These results also show that the Lagrangian data used to construct the new Lagrangian α-ROM play an important role: *the higher the Lagrangian data contribution* (i.e., the higher the α value), *the more accurate the results*.



(a) **(b)** **(c)** **(d)**

(e) **(f)** **(g)** **(h)**

(i) **(j)** **(k)** **(l)**

(m) **(n)** **(o)** **(p)**

Figure 4. Eulerian investigation, reconstructive regime: Mean streamfunction from E-ROM (plots (**a**–**d**)), λ-ROM (plots (**e**–**h**)) α-ROM with $\alpha = 1$ (plots (**i**–**l**)), and $\alpha = 10^4$ (plots (**m**–**p**)), for $r = 10$ (plots (**a**,**e**,**i**,**m**)), $r = 15$ (plots (**b**,**f**,**j**,**n**)), $r = 20$ (plots (**c**,**g**,**k**,**o**)), and $r = 30$ (plots (**d**,**h**,**l**,**p**)).

Predictive Regime:

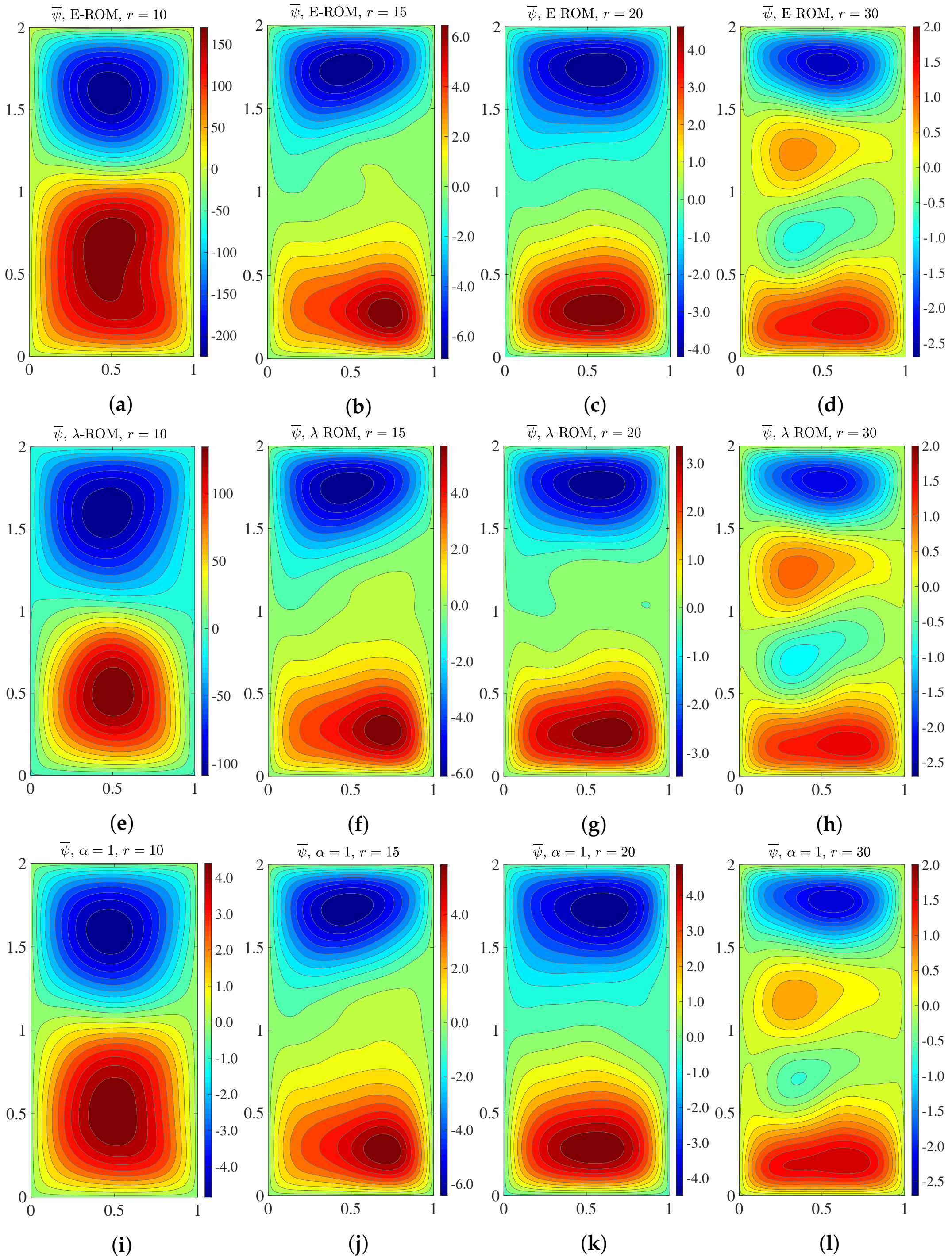
For the predictive regime, we check whether the investigated ROMs can predict the dynamics of the underlying system. Specifically, we validate the ROMs on a time interval that is twice as long as the time interval used to train the ROMs. In Table 3, for different r values, we list the L^2 norm of the errors in the time-averaged streamfunction (22) for E-ROM, λ-ROM, and α-ROM with $\alpha = 1, \alpha = 10, \alpha = 10^2, \alpha = 10^3$, and $\alpha = 10^4$. These results yield conclusions that are similar to those drawn in the reconstructive regime: For low r values, the E-ROM yields inaccurate results. The λ-ROM results are similar to or slightly better than the E-ROM results for all r values. The results for α-ROM with $\alpha = 1, \alpha = 10$, and $\alpha = 10^2$ are generally better than or similar to the E-ROM results. The results for α-ROM with $\alpha = 10^3$ and $\alpha = 10^4$ are dramatically better than the E-ROM results: For example, for $r = 5$, the only ROM that yields acceptable results is the α-ROM with $\alpha = 10^4$; all other ROMs simply blow up (this is denoted by "N/A" in Table 3). Furthermore, for $r = 10$, $r = 15$ and $r = 25$, the errors of α-ROM with $\alpha = 10^3$ and $\alpha = 10^4$ are *at least one order of magnitude* lower than the E-ROM error. For the larger r values, the errors of α-ROM with $\alpha = 10^3$ and $\alpha = 10^4$ continue to be lower than the E-ROM errors, although (as expected) the differences decrease as the r values increase. Overall, the results in Table 3 show that, in the predictive regime, *the Lagrangian α-ROM* yields *significantly more accurate results than the standard E-ROM and the Lagrangian λ-ROM*. Specifically, for the small r values, the Lagrangian α-ROM with high α values (i.e., $\alpha = 10^3$ and $\alpha = 10^4$) performs the best, while for the large r values, the Lagrangian α-ROM with small α values (i.e., $\alpha = 1$) performs the best.

Table 3. Eulerian investigation, predictive regime: L^2 norm of the errors in the time-averaged streamfunction (22) for E-ROM (second column), λ-ROM (third column), and α-ROM for $\alpha = 1$ (fourth column), $\alpha = 10$ (fifth column), $\alpha = 10^2$ (sixth column), $\alpha = 10^3$ (seventh column), and $\alpha = 10^4$ (eighth column).

r	E-ROM	λ-ROM	$\alpha = 1$	$\alpha = 10$	$\alpha = 10^2$	$\alpha = 10^3$	$\alpha = 10^4$
5	N/A	N/A	N/A	N/A	N/A	2.8×10^1	1.2×10^1
10	2.6×10^4	8.6×10^3	1.6×10^4	5.3×10^3	4.8×10^1	8.1×10^0	7.7×10^0
15	1.1×10^1	1.8×10^1	1.9×10^1	2.2×10^1	7.3×10^0	1.8×10^0	1.3×10^0
20	9.0×10^0	6.3×10^0	8.8×10^0	9.5×10^0	1.7×10^0	3.5×10^{-1}	5.8×10^{-1}
25	4.0×10^0	2.5×10^0	3.8×10^0	2.7×10^0	8.4×10^{-1}	3.3×10^{-1}	3.5×10^{-1}
30	1.0×10^0	7.7×10^{-1}	8.8×10^{-1}	6.8×10^{-1}	6.0×10^{-1}	1.9×10^{-1}	6.2×10^{-1}
35	4.3×10^{-1}	4.9×10^{-1}	5.8×10^{-1}	5.5×10^{-1}	5.4×10^{-1}	3.4×10^{-1}	5.2×10^{-1}
40	4.8×10^{-1}	3.8×10^{-1}	3.8×10^{-1}	4.9×10^{-1}	4.4×10^{-1}	7.6×10^{-1}	5.7×10^{-1}
45	6.4×10^{-1}	4.4×10^{-1}	4.6×10^{-1}	4.9×10^{-1}	6.3×10^{-1}	7.5×10^{-1}	5.3×10^{-1}
50	4.0×10^{-1}	4.9×10^{-1}	3.52×10^{-1}	3.8×10^{-1}	6.1×10^{-1}	4.0×10^{-1}	4.3×10^{-1}

Next, we use the qualitative Eulerian criterion to investigate the ability of the ROMs to *predict* the four-gyre pattern of the time average of the streamfunction. In Figure 5, for $r = 10, 15, 20$, and 30, we plot the mean streamfunction for E-ROM, λ-ROM, and α-ROM with $\alpha = 1$ and $\alpha = 10^4$. These results yield the following conclusions: The E-ROM, the λ-ROM, and the α-ROM with a low α value (i.e., $\alpha = 1$) yield similar results. These ROMs cannot recover the four-gyre pattern for $r = 10, 15$, and 20, although they can capture the four-gyre pattern for $r = 30$. However, the α-ROM with a large α value (i.e., $\alpha = 10^4$) yields dramatically better results. This ROM can clearly capture the four-gyre pattern not only for $r = 30$, but also for $r = 30$; for $r = 15$, the pattern is somewhat captured, although not as clearly as for $r = 20$ and $r = 30$; finally, for $r = 10$, only hints of the four-gyre pattern are present. Overall, the plots in Figure 5 show that the *Lagrangian α-ROM with a large α value* (i.e., $\alpha = 10^4$) *can capture the four-gyre pattern*, whereas the *standard E-ROM and the Lagrangian λ-ROM cannot*.

The results in Table 3 and Figure 5 consistently show that, in the predictive regime, the *new Lagrangian α-ROM with large α values outperforms the standard E-ROM and the Lagrangian λ-ROM* with respect to the two Eulerian metrics used in this section. These results also support the conclusion from the reconstructive regime, that is, *the higher the Lagrangian data contribution* (i.e., the higher the α value), *the more accurate the results*.

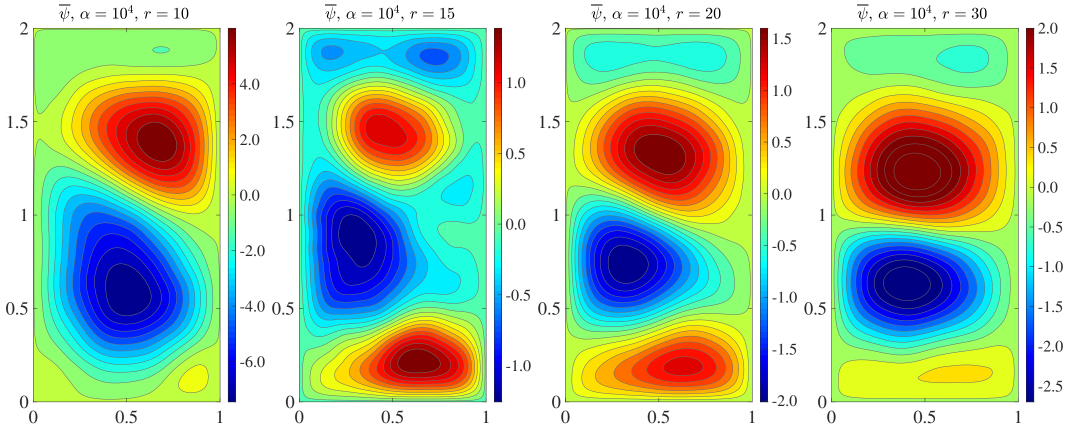


(**a**) (**b**) (**c**) (**d**)

(**e**) (**f**) (**g**) (**h**)

(**i**) (**j**) (**k**) (**l**)

Figure 5. *Cont.*



(**m**) (**n**) (**o**) (**p**)

Figure 5. Eulerian investigation, predictive regime: Mean streamfunction from E-ROM (plots (**a**–**d**)), λ-ROM (plots (**e**–**h**)) α-ROM with $\alpha = 1$ (plots (**i**–**l**)), and $\alpha = 10^4$ (plots (**m**–**p**)), for $r = 10$ (plots (**a**,**e**,**i**,**m**)), $r = 15$ (plots (**b**,**f**,**j**,**n**)), $r = 20$ (plots (**c**,**g**,**k**,**o**)), and $r = 30$ (plots (**d**,**h**,**l**,**p**)).

3.5.2. Lagrangian Investigation

In this section, we perform a Lagrangian investigation of the accuracy of the two Lagrangian ROMs (i.e., the α-ROM and λ-ROM). We follow the same format as that used in the Eulerian investigation in Section 3.5.1.

Reconstructive Regime:

In Table 4, we list the L^2 norm of the errors in the time-averaged FTLE (23) for E-ROM, λ-ROM, and α-ROM for different r values. The results in Table 4 show that the α-ROM with high α values (i.e., $\alpha = 10^3$ and $\alpha = 10^4$) consistently outperforms the λ-ROM and the E-ROM for all r values, but especially for the small r values. We also note that the relatively high magnitudes of the errors in Table 4 are due to the errors in the ROM velocity field approximations. Decreasing the magnitude of the errors in the ROM velocity field approximations (e.g., by increasing the ROM dimension, r) would probably decrease the magnitude of the errors in the FTLE field approximation in Table 4.

Table 4. Lagrangian investigation, reconstructive regime: L^2 norm of the errors in the time-averaged FTLE (23) for E-ROM (second column) λ-ROM (third column), and α-ROM for $\alpha = 1$ (fourth column), $\alpha = 10$ (fifth column), $\alpha = 10^2$ (sixth column), $\alpha = 10^3$ (seventh column), and $\alpha = 10^4$ (eighth column).

r	E-ROM	λ-ROM	$\alpha = 1$	$\alpha = 10$	$\alpha = 10^2$	$\alpha = 10^3$	$\alpha = 10^4$
10	6.0×10^3	5.9×10^3	5.4×10^3	4.2×10^3	5.2×10^3	4.8×10^1	2.5×10^1
15	4.6×10^2	1.2×10^3	1.2×10^2	8.3×10^1	7.8×10^1	3.2×10^1	6.1×10^1
20	9.8×10^1	7.4×10^1	8.5×10^1	5.9×10^1	2.2×10^1	1.2×10^1	1.0×10^1
25	8.2×10^1	6.2×10^1	8.5×10^1	7.6×10^1	1.8×10^1	1.0×10^1	9.5×10^0
30	9.4×10^1	8.7×10^1	9.2×10^1	7.0×10^1	1.8×10^1	1.2×10^1	8.3×10^0
35	7.8×10^1	5.8×10^1	7.6×10^1	5.9×10^1	1.3×10^1	1.1×10^1	1.2×10^1
40	7.5×10^1	5.3×10^1	5.8×10^1	5.7×10^1	1.3×10^1	2.0×10^1	2.6×10^1
45	5.9×10^1	5.0×10^1	5.5×10^1	4.9×10^1	1.4×10^1	2.4×10^1	2.4×10^1
50	4.8×10^1	4.4×10^1	4.6×10^1	3.8×10^1	1.1×10^1	1.4×10^1	2.2×10^1

Predictive Regime:

In Table 5, we list the L^2 norm of the errors in the time-averaged FTLE (23) for E-ROM, λ-ROM, and α-ROM for different r values. The results in Table 5 show that, as in the reconstructive regime, the α-ROM with high α values (i.e., $\alpha = 10^3$ and $\alpha = 10^4$) consistently outperforms the λ-ROM and the E-ROM for all r values.

Table 5. Lagrangian investigation, predictive regime: L^2 norm of the errors in the time-averaged FTLE (23) for E-ROM (second column) λ-ROM (third column), and α-ROM for $\alpha = 1$ (fourth column), $\alpha = 10$ (fifth column), $\alpha = 10^2$ (sixth column), $\alpha = 10^3$ (seventh column), and $\alpha = 10^4$ (eighth column).

r	**E-ROM**	$\boldsymbol{\lambda}$**-ROM**	$\boldsymbol{\alpha = 1}$	$\boldsymbol{\alpha = 10}$	$\boldsymbol{\alpha = 10^2}$	$\boldsymbol{\alpha = 10^3}$	$\boldsymbol{\alpha = 10^4}$
10	3.6×10^3	4.4×10^3	3.0×10^3	4.1×10^3	3.0×10^3	4.4×10^2	2.4×10^2
15	5.7×10^2	5.2×10^2	4.1×10^2	5.0×10^2	1.0×10^2	1.8×10^1	1.2×10^1
20	1.5×10^2	8.4×10^1	1.5×10^2	1.7×10^2	7.7×10^1	1.3×10^1	8.9×10^0
25	1.2×10^2	9.2×10^1	9.5×10^1	1.1×10^2	5.8×10^1	8.9×10^0	9.3×10^0
30	7.6×10^1	7.3×10^1	1.2×10^2	6.5×10^1	2.1×10^1	6.6×10^0	6.7×10^0
35	6.7×10^1	4.6×10^1	5.9×10^1	4.6×10^1	1.6×10^1	6.6×10^0	5.6×10^0
40	3.1×10^1	2.8×10^1	2.6×10^1	3.0×10^1	1.1×10^1	1.3×10^1	2.0×10^1
45	2.2×10^1	1.8×10^1	2.0×10^1	1.6×10^1	9.3×10^0	1.5×10^1	1.9×10^1
50	1.9×10^1	1.6×10^1	1.8×10^1	1.5×10^1	9.1×10^0	1.4×10^1	1.5×10^1

3.6. ROM Computational Efficiency

In this section, we investigate the computational efficiency of the new Lagrangian ROMs (i.e., α-ROM and λ-ROM).

3.6.1. Computational Environments

We use the following computational environments: To generate the FOM velocity fields, we run the code on one processor (and one thread) on a Dell workstation with a 2.00 GHZ Intel Xeon CPU running on a 64-bit Linux system. To generate the ROM velocity fields, we use one Apple laptop with a single 2.70 GHZ CPU, running on a 64-bit Macintosh operating systems. To generate the FTLE fields, we utilize: (i) One computing cluster composed of 5 nodes, each node comprised of dual, quad core, hyperthreaded 2.4 GHz Intel Xeon E5620 CPUs (16 processor threads), 24 GB RAM, and a 40 Gbps InfiniBand host card and cable; and (ii) Five nodes at 12 threads per node, for a total of 60 threads, and 4749 mb of memory for each thread.

3.6.2. Speed-Up Factors

The FOM CPU time has two components: the CPU time of generating the velocity field; and the CPU time of generating the FTLE field from the velocity field. The ROM CPU time has three components: the CPU time of the offline phase (i.e., the construction of ROM operators); the CPU time of the online phase (i.e., running the ROMs to generate the velocity field); and the CPU time of generating the FTLE field from the velocity field.

In this section, we investigate the CPU times of the velocity computation, since this is the main target of the proposed Lagrangian ROMs. Thus, we first investigate the ROM speed-ups in the velocity computation and then briefly discuss the CPU times in the FTLE field computation. Furthermore, as often done in ROM investigations, we monitor only the CPU time of the online phase of the ROMs, since the offline CPU time is offset by running the ROMs in the predictive regime, that is, for longer time intervals (as done in this paper) or for different parameters.

To compute the computational efficiency of the new Lagrangian ROMs, we compute the ROM *speed-up factors (S_f)*, which are defined as follows:

$$S_f = \frac{\text{FOM CPU time}}{\text{ROM CPU time}}, \tag{24}$$

where the FOM CPU time is the CPU time of generating the velocity field and the ROM CPU time is the CPU time of the ROM online phase.

In Table 6, for different r values, we list the speed-up factors (24) for E-ROM, λ-ROM (second column), and α-ROM with $\alpha = 1, \alpha = 10^2, \alpha = 10^3$, and $\alpha = 10^4$. These results show that the new Lagrangian ROMs and the standard Eulerian ROM are *more than three orders of magnitude faster* than the FOM.

Table 6. Speed-up factors (24) for velocity field computation: E-ROM (second column), λ-ROM (third column), and α-ROM for $\alpha = 1$ (fourth column), $\alpha = 10$ (fifth column), $\alpha = 10^2$ (sixth column), $\alpha = 10^3$ (seventh column), and $\alpha = 10^4$ (eighth column).

r	**E-ROM**	λ**-ROM**	$\alpha = 1$	$\alpha = 10$	$\alpha = 10^2$	$\alpha = 10^3$	$\alpha = 10^4$
10	1.4×10^4	1.5×10^4	1.5×10^4	1.5×10^4	1.5×10^4	1.5×10^4	1.5×10^4
15	8.4×10^3	8.2×10^3	8.2×10^3	8.3×10^3	7.8×10^3	8.0×10^3	8.0×10^3
20	5.6×10^3	5.7×10^3	5.7×10^3	5.9×10^3	6.1×10^3	5.8×10^3	5.8×10^3
25	4.8×10^3	4.6×10^3	4.7×10^3	4.8×10^3	4.5×10^3	4.6×10^3	4.6×10^3
30	3.7×10^3	3.6×10^3	3.7×10^3	3.9×10^3	3.6×10^3	3.7×10^3	3.7×10^3
35	2.9×10^3	2.9×10^3	3.0×10^3	3.0×10^3	2.9×10^3	2.9×10^3	2.9×10^3
40	2.5×10^3	2.6×10^3	2.6×10^3	2.6×10^3	2.5×10^3	2.6×10^3	2.5×10^3
45	2.0×10^3	2.0×10^3	2.1×10^3	2.1×10^3	2.0×10^3	2.0×10^3	2.0×10^3
50	1.7×10^3	1.7×10^3	1.8×10^3	1.8×10^3	1.7×10^3	1.7×10^3	1.7×10^3

Although the speed-up factors for the ROM velocity computation in Table 6 are the main focus of the proposed Lagrangian ROMs, we briefly comment on the CPU time of the ROM computation of the FTLE field. Overall, the CPU time of the ROM computation of the FTLE field is generally slightly higher than the CPU time of the FOM computation of the FTLE field, especially when relative low r values are used. We also note that, for low r values, this CPU time increase is generally lower for the new Lagrangian α-ROM with $\alpha = 10^3$ and $\alpha = 10^4$ than for the standard E-ROM. We believe that the reason for this slight CPU time increase is that, as expected, the ROM velocity accuracy is lower than the FOM velocity accuracy, which results in a slight increase in the CPU time of the FTLE field computation. We plan to investigate this in a future study.

To conclude, the overall CPU time of the FTLE field computation is generally several times lower for the two new Lagrangian ROMs (as well as the standard E-ROM) than for the DNS. Indeed, the two new Lagrangian ROMs decrease the CPU time of the FOM velocity field computation by orders of magnitude and only slightly increase the CPU time of the FOM FTLE field computation.

4. Conclusions and Outlook

In this paper, we proposed Lagrangian ROMs that use new Lagrangian inner products to build the ROM basis. In these Lagrangian inner products, Lagrangian data "steers" the resulting Lagrangian ROM basis toward an accurate approximation of Lagrangian quantities, whereas Eulerian data helps the Lagrangian ROM basis yield an accurate approximation of Eulerian quantities.

For complex nonlinear systems, the low-dimensional ROMs generally need to be equipped with closure models or stabilization mechanisms [11,12,34]. We emphasize, however, that we studied the new Lagrangian ROMs without any closure model (a challenging test) in order to separate the ROM closure problem from the ROM basis generation, which is the main focus of our paper.

We investigated the new Lagrangian ROMs in the numerical simulation of the QGE. We considered both the reconstructive regime (in which the ROM is validated on the same time interval as the time interval used to train the ROM) and the predictive regime (in which the ROM is trained on a short time interval and validated on a longer time interval). In both the reconstructive and predictive regimes, we showed that the new Lagrangian ROMs numerical accuracies were higher than the standard Eulerian ROM accuracy in approximating both Eulerian fields (i.e., the velocity field) and Lagrangian fields (i.e., the FTLE field). In some cases, the α-ROMs' accuracies were *orders of magnitude* higher than the standard Eulerian ROM accuracy. We emphasize that, since the new Lagrangian ROMs did not employ any closure modeling, the dramatic increase in the new Lagrangian ROMs' accuracy is entirely due to the new Lagrangian inner products used to build the Lagrangian ROM basis. Furthermore, we showed that, for the velocity field computations, the online CPU times of the new Lagrangian ROMs are orders of magnitude lower than the CPU time of the corresponding FOM.

There are numerous research directions that could provide improvements both in the efficiency and the accuracy of the new Lagrangian ROMs. Probably the most important research avenue is the investigation of ROM closure models for the new Lagrangian ROMs. Indeed, the new Lagrangian ROMs improved the standard Eulerian ROM's accuracy solely by using a ROM basis constructed with the new Lagrangian inner products. We plan to further increase the accuracy of the new Lagrangian ROMs by adding ROM closure models for the effect of the discarded ROM modes, for example, data-driven ROM closure models [11,12,34] or eddy viscosity ROM closure models [35]. Another potential research direction is finding the optimal α value in the new Lagrangian α-ROM. Although the α-ROM with higher α values yielded the most accurate results in our numerical investigation, finding the *optimal* α value is still an open question. To find this optimal parameter value, one could try to extend to the Lagrangian setting the mathematical tools developed for Eulerian ROMs [16,77]. Another research avenue is the extension of the new Lagrangian ROMs and the novel Lagrangian inner products to the computation of other structures that characterize transport and mixing. For example, instead of geometric approaches (such as the FTLE field), one could approximate probabilistic measures, such as the almost invariant sets [86,87]. An additional research direction is to investigate whether the new Lagrangian inner products can be used not only to construct the ROM basis, but also to build the ROM operators. Yet another research direction worthy of investigation is the delicate interplay between the QGE data used to construct the ROM basis and the reconstructive and predictive capabilities of the resulting Lagrangian and Eulerian ROMs. Finally, although the new Lagrangian ROMs dramatically reduced the computational cost of velocity field computation, we intend to explore different approaches for speeding up the FTLE field computation from available ROM velocity data. To this end, we plan to use the new Lagrangian ROMs in conjunction with the algorithms that have been recently proposed in References [88,89].

Author Contributions: Conceptualization, T.I. and S.D.R.; methodology, T.I. and S.D.R.; software, C.M., P.J.N. and X.X.; validation, C.M., P.J.N. and X.X.; formal analysis, C.M., P.J.N. and X.X.; investigation, C.M., P.J.N. and X.X.; resources, C.M., P.J.N. and X.X.; data curation, C.M., P.J.N. and X.X.; writing–original draft preparation, T.I. and S.D.R.; writing–review and editing, T.I. and S.D.R.; visualization, C.M., P.J.N. and X.X.; supervision, T.I. and S.D.R.; project administration, T.I. and S.D.R.; funding acquisition, T.I. and S.D.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Science Foundation under grants DMS-1821145 and DMS-2012253.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Brunton, S.L.; Kutz, J.N. *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*; Cambridge University Press: Cambridge, England, 2019.
2. Hesthaven, J.S.; Rozza, G.; Stamm, B. *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*; Springer: Berlin/Heidelberg, Germany, 2015.
3. Holmes, P.; Lumley, J.L.; Berkooz, G. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*; Cambridge University Press: Cambridge, England, 1996.
4. Quarteroni, A.; Manzoni, A.; Negri, F. *Reduced Basis Methods for Partial Differential Equations: An Introduction*; Springer: Berlin/Heidelberg, Germany, 2015; Volume 92.
5. Noack, B.R.; Morzynski, M.; Tadmor, G. *Reduced-Order Modelling for Flow Control*; Springer: Berlin/Heidelberg, Germany, 2011; Volume 528.
6. Taira, K.; Hemati, M.S.; Brunton, S.L.; Sun, Y.; Duraisamy, K.; Bagheri, S.; Dawson, S.T.M.; Yeh, C.A. Modal analysis of fluid flows: Applications and outlook. *AIAA J.* **2020**, *58*, 998–1022.
7. Chekroun, M.D.; Liu, H.; McWilliams, J.C. Variational approach to closure of nonlinear dynamical systems: Autonomous case. *J. Stat. Phys.* **2020**, *179*, 1073–1160.

8. Hijazi, S.; Stabile, G.; Mola, A.; Rozza, G. Data-driven POD-Galerkin reduced order model for turbulent flows. *J. Comput. Phys.* **2020**, *416*, 109513.
9. Parish, E.J.; Wentland, C.; Duraisamy, K. The adjoint Petrov–Galerkin method for non-linear model reduction. *Comput. Meth. Appl. Mech. Eng.* **2020**, *365*, 112991.
10. Reyes, R.; Codina, R. Projection-based reduced order models for flow problems: A variational multiscale approach. *Comput. Methods Appl. Mech. Engrg.* **2019**, submitted.
11. Xie, X.; Mohebujjaman, M.; Rebholz, L.G.; Iliescu, T. Data-Driven Filtered Reduced Order Modeling of Fluid Flows. *SIAM J. Sci. Comput.* **2018**, *40*, B834–B857.
12. Mou, C.; Koc, B.; San, O.; Rebholz, L.G.; Iliescu, T. Data-Driven Variational Multiscale Reduced Order Models. *Comput. Methods Appl. Mech. Engrg.* **2020**, *373*, 113470.
13. Azaïez, M.; Rebollo, T.C.; Rubino, S. Streamline derivative projection-based POD-ROM for convection-dominated flows. Part I: Numerical Analysis. *arXiv* **2017**, arXiv:1711.09780.
14. Grimberg, S.; Farhat, C.; Youkilis, N. On the stability of projection-based model order reduction for convection-dominated laminar and turbulent flows. *arXiv* **2020**, arXiv:2001.10110.
15. Gunzburger, M.; Iliescu, T.; Mohebujjaman, M.; Schneier, M. An evolve-filter-relax stabilized reduced order stochastic collocation method for the time-dependent Navier-Stokes equations. *SIAM-ASA J. Uncertain.* **2019**, *7*, 1162–1184.
16. Xie, X.; Wells, D.; Wang, Z.; Iliescu, T. Numerical Analysis of the Leray Reduced Order Model. *J. Comput. Appl. Math.* **2018**, *328*, 12–29.
17. Iollo, A.; Lanteri, S.; Désidéri, J.A. Stability properties of POD–Galerkin approximations for the compressible Navier–Stokes equations. *Theoret. Comput. Fluid Dyn.* **2000**, *13*, 377–396.
18. Sengupta, T.K.; Haider, S.I.; Parvathi, M.K.; Pallavi, G. Enstrophy-based proper orthogonal decomposition for reduced-order modeling of flow past a cylinder. *Phys. Rev. E* **2015**, *91*, 043303.
19. Barone, M.F.; Kalashnikova, I.; Segalman, D.J.; Thornquist, H.K. Stable Galerkin reduced order models for linearized compressible flow. *J. Comput. Phys.* **2009**, *228*, 1932–1946.
20. Kalashnikova, I.; Arunajatesan, S.; Barone, M.F.; van Bloemen Waanders, B.G.; Fike, J.A. Reduced order modeling for prediction and control of large-scale systems. In *Sandia National Laboratories Report*; SAND: Albuquerque, NM, USA; Livermore, CA, UAS; 2014.
21. Rowley, C.W.; Colonius, T.; Murray, R.M. Model reduction for compressible flows using POD and Galerkin projection. *Phys. D: Nonlinear Phenom.* **2004**, *189*, 115–129.
22. Kaptanoglu, A.A.; Morgan, K.D.; Hansen, C.J.; Brunton, S.L. Physics-constrained, low-dimensional models for MHD: First-principles and data-driven approaches. *arXiv* **2020**, arXiv:2004.10389.
23. Daescu, D.N.; Navon, I.M. A dual-weighted approach to order reduction in 4DVAR data assimilation. *Mon. Weather Rev.* **2008**, *136*, 1026–1041.
24. Lu, H.; Tartakovsky, D.M. Lagrangian dynamic mode decomposition for construction of reduced-order models of advection-dominated phenomena. *J. Comput. Phys.* **2020**, *407*, 109229.
25. Mojgani, R.; Balajewicz, M. Lagrangian basis method for dimensionality reduction of convection dominated nonlinear flows. *arXiv* **2017**, arXiv:1701.04343.
26. Cushman-Roisin, B.; Beckers, J.M. *Introduction to Geophysical Fluid Dynamics: Physical and Numerical Aspects*; Academic Press: Cambridge, MA, USA, 2011.
27. Majda, A. *Introduction to PDEs and Waves for the Atmosphere and Ocean*; American Mathematical Society: New York, NY, USA, 2003.
28. Majda, A.J.; Wang, X. *Nonlinear Dynamics and Statistical Theories for Basic Geophysical Flows*; Cambridge University Press: Cambridge, UK, 2006; p. xii+551.
29. Vallis, G.K. *Atmospheric and Oceanic Fluid Dynamics: Fundamentals and Large-Scale Circulation*; Cambridge University Press: Cambridge, England, 2006.
30. Chen, N.; Majda, A.J. Conditional Gaussian systems for multiscale nonlinear stochastic systems: Prediction, state estimation and uncertainty quantification. *Entropy* **2018**, *20*, 509.
31. Majda, A.J.; Chen, N. Model error, information barriers, state estimation and prediction in complex multiscale systems. *Entropy* **2018**, *20*, 644.
32. Majda, A.J.; Qi, D. Strategies for reduced-order models for predicting the statistical responses and uncertainty quantification in complex turbulent dynamical systems. *SIAM Rev.* **2018**, *60*, 491–549.

33. Crommelin, D.T.; Majda, A.J. Strategies for model reduction: Comparing different optimal bases. *J. Atmos. Sci.* **2004**, *61*, 2206–2217.
34. Mou, C.; Liu, H.; Wells, D.R.; Iliescu, T. Data-Driven Correction Reduced Order Models for the Quasi-Geostrophic Equations: A Numerical Investigation. *Int. J. Comput. Fluid Dyn.* **2020**, *34*, 147–159.
35. San, O.; Iliescu, T. A stabilized proper orthogonal decomposition reduced-order model for large scale quasigeostrophic ocean circulation. *Adv. Comput. Math.* **2015**, 1289–1319.
36. Selten, F.M. An efficient description of the dynamics of barotropic flow. *J. Atmos. Sci.* **1995**, *52*, 915–936.
37. Strazzullo, M.; Ballarin, F.; Mosetti, R.; Rozza, G. Model Reduction for Parametrized Optimal Control Problems in Environmental Marine Sciences and Engineering. *SIAM J. Sci. Comput.* **2018**, *40*, B1055–B1079.
38. Foster, E.L.; Iliescu, T.; Wang, Z. A finite element discretization of the streamfunction formulation of the stationary quasi-geostrophic equations of the ocean. *Comput. Methods Appl. Mech. Engrg.* **2013**, *261*, 105–117.
39. Monteiro, I.O.; Manica, C.C.; Rebholz, L.G. Numerical study of a regularized barotropic vorticity model of geophysical flow. *Numer. Methods Partial Differ. Equations* **2015**, 1492–1514.
40. San, O.; Staples, A.E.; Wang, Z.; Iliescu, T. Approximate deconvolution large eddy simulation of a barotropic ocean circulation model. *Ocean Model.* **2011**, *40*, 120–132.
41. Haller, G. Lagrangian coherent structures. *Ann. Rev. Fluid Mech.* **2015**, *47*, 137–162.
42. Perotto, S.; Reali, A.; Rusconi, P.; Veneziani, A. HIGAMod: A Hierarchical IsoGeometric Approach for MODel reduction in curved pipes. *Comput. Fluids* **2017**, *142*, 21–29.
43. Iliescu, T.; Wang, Z. Are the Snapshot Difference Quotients Needed in the Proper Orthogonal Decomposition? *SIAM J. Sci. Comput.* **2014**, *36*, A1221–A1250.
44. Babaee, H.; Farazmand, M.; Haller, G.; Sapsis, T.P. Reduced-order description of transient instabilities and computation of finite-time Lyapunov exponents. *Chaos* **2017**, *27*, 063103.
45. Kourentis, L.; Konstantinidis, E. Uncovering large-scale coherent structures in natural and forced turbulent wakes by combining PIV, POD, and FTLE. *Exp. Fluids* **2012**, *52*, 749–763.
46. Surana, A.; Hariharan, N.; Narayanan, S.; Banaszuk, A. Reduced order modeling for contaminant transport and mixing in building systems: A case study using dynamical systems techniques. In Proceedings of the 2008 American Control Conference, Seattle, WA, USA, 11–13 June 2008; pp. 902–907.
47. Nonino, M.; Ballarin, F.; Rozza, G.; Maday, Y. Overcoming slowly decaying Kolmogorov n-width by transport maps: Application to model order reduction of fluid dynamics and fluid–structure interaction problems. *arXiv* **2019**, arXiv:1911.06598.
48. Amsallem, D.; Zahr, M.J.; Farhat, C. Nonlinear model order reduction based on local reduced-order bases. *Int. J. Num. Meth. Eng.* **2012**, *92*, 891–916.
49. San, O.; Borggaard, J. Principal interval decomposition framework for POD reduced-order modeling of convective Boussinesq flows. *Int. J. Num. Meth. Fluids* **2015**, *78*, 37–62.
50. Lucia, D.J. *Reduced Order Modeling for High Speed Flows with Moving Shocks*; Technical Report; Air Force Institute of Technology, Wright-Patterson Air Force Base: Greene County, OH, USA, 2001.
51. Carlberg, K. Adaptive h-refinement for reduced-order models. *Int. J. Num. Meth. Eng.* **2015**, *102*, 1192–1210.
52. Peherstorfer, B. Model reduction for transport-dominated problems via online adaptive bases and adaptive sampling. *arXiv* **2018**, arXiv:1812.02094.
53. Rowley, C.W.; Kevrekidis, I.G.; Marsden, J.E.; Lust, K. Reduction and reconstruction for self-similar dynamical systems. *Nonlinearity* **2003**, *16*, 1257.
54. Rowley, C.W.; Marsden, J.E. Reconstruction equations and the Karhunen–Loève expansion for systems with symmetry. *Phys. D Nonlinear Phenom.* **2000**, *142*, 1–19.
55. Gerbeau, J.F.; Lombardi, D. Approximated Lax pairs for the reduced order integration of nonlinear evolution equations. *J. Comput. Phys.* **2014**, *265*, 246–269.
56. Bernard, F.; Iollo, A.; Riffaud, S. Reduced-order model for the BGK equation based on POD and optimal transport. *J. Comput. Phys.* **2018**, *373*, 545–570.
57. Cagniart, N.; Maday, Y.; Stamm, B. Model order reduction for problems with large convection effects. In *Contributions to Partial Differential Equations and Applications*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 131–150.
58. Nair, N.J.; Balajewicz, M. Transported snapshot model order reduction approach for parametric, steady-state fluid flows containing parameter-dependent shocks. *Int. J. Num. Meth. Engng.* **2019**, *117*, 1234–1262.

59. Ohlberger, M.; Rave, S. Nonlinear reduced basis approximation of parameterized evolution equations via the method of freezing. *C. R. Math.* **2013**, *351*, 901–906.
60. Reiss, J.; Schulze, P.; Sesterhenn, J.; Mehrmann, V. The shifted proper orthogonal decomposition: A mode decomposition for multiple transport phenomena. *SIAM J. Sci. Comput.* **2018**, *40*, A1322–A1344.
61. Rim, D.; Moe, S.; LeVeque, R.J. Transport reversal for model reduction of hyperbolic partial differential equations. *SIAM-ASA J. Uncertain.* **2018**, *6*, 118–150.
62. Welper, G. h and hp-adaptive interpolation by transformed snapshots for parametric and stochastic hyperbolic PDEs. *arXiv* **2017**, arXiv:1710.11481.
63. Carlberg, K.; Tuminaro, R.; Boggs, P. Preserving Lagrangian structure in nonlinear model reduction with application to structural dynamics. *SIAM J. Sci. Comput.* **2015**, *37*, B153–B184.
64. Lall, S.; Krysl, P.; Marsden, J.E. Structure-preserving model reduction for mechanical systems. *Phys. D Nonlinear Phenom.* **2003**, *184*, 304–318.
65. Afkham, B.M.; Hesthaven, J.S. Structure preserving model reduction of parametric Hamiltonian systems. *SIAM J. Sci. Comput.* **2017**, *39*, A2616–A2644.
66. Breiten, T.; Benner, P. Interpolation-based $\mathcal{H}_2$-model reduction of bilinear control system. *SIAM J. Matrix Anal. Appl* **2012**, *33*, 859–885.
67. Chaturantabut, S.; Beattie, C.; Gugercin, S. Structure-preserving model reduction for nonlinear port-Hamiltonian systems. *SIAM J. Sci. Comput.* **2016**, *38*, B837–B865.
68. Gong, Y.; Wang, Q.; Wang, Z. Structure-preserving Galerkin POD reduced-order modeling of Hamiltonian systems. *Comput. Methods Appl. Mech. Engrg.* **2017**, *315*, 780–798.
69. Peng, L.; Mohseni, K. Symplectic model reduction of Hamiltonian systems. *SIAM J. Sci. Comput.* **2016**, *38*, A1–A27.
70. Chan, J. Entropy stable reduced order modeling of nonlinear conservation laws. *arXiv* **2019**, arXiv:1909.09103.
71. Farhat, C.; Avery, P.; Chapman, T.; Cortial, J. Dimensional reduction of nonlinear finite element dynamic models with finite rotations and energy-based mesh sampling and weighting for computational efficiency. *Int. J. Num. Meth. Eng.* **2014**, *98*, 625–662.
72. Farhat, C.; Chapman, T.; Avery, P. Structure-preserving, stability, and accuracy properties of the energy-conserving sampling and weighting method for the hyper reduction of nonlinear finite element dynamic models. *Int. J. Num. Meth. Eng.* **2015**, *102*, 1077–1110.
73. Loiseau, J.C.; Brunton, S.L. Constrained sparse Galerkin regression. *J. Fluid Mech.* **2018**, *838*, 42–67.
74. Mohebujjaman, M.; Rebholz, L.G.; Xie, X.; Iliescu, T. Energy balance and mass conservation in reduced order models of fluid flows. *J. Comput. Phys.* **2017**, *346*, 262–277.
75. Mohebujjaman, M.; Rebholz, L.G.; Iliescu, T. Physically-constrained data-driven correction for reduced order modeling of fluid flows. *Int. J. Num. Meth. Fluids* **2019**, *89*, 103–122.
76. Yano, M. Discontinuous Galerkin reduced basis empirical quadrature procedure for model reduction of parametrized nonlinear conservation laws. *Adv. Comput. Math.* **2019**, *45*, 2287–2320.
77. Kunisch, K.; Volkwein, S. Galerkin proper orthogonal decomposition methods for parabolic problems. *Numer. Math.* **2001**, *90*, 117–148.
78. Barrault, M.; Maday, Y.; Nguyen, N.C.; Patera, A.T. An 'Empirical Interpolation' Method: Application to Efficient Reduced-Basis Discretization of Partial Differential Equations. *C. R. Acad. Sci. Paris Ser. I* **2004**, *339*, 667–672.
79. Chaturantabut, S.; Sorensen, D.C. Nonlinear model reduction via discrete empirical interpolation. *SIAM J. Sci. Comput.* **2010**, *32*, 2737–2764.
80. Cummins, P.F. Inertial gyres in decaying and forced geostrophic turbulence. *J. Mar. Res.* **1992**, *50*, 545–566.
81. Greatbatch, R.J.; Nadiga, B.T. Four-gyre circulation in a barotropic model with double-gyre wind forcing. *J. Phys. Oceanogr.* **2000**, *30*, 1461–1471.
82. Holm, D.D.; Nadiga, B.T. Modeling mesoscale turbulence in the barotropic double-gyre circulation. *J. Phys. Oceanogr.* **2003**, *33*, 2355–2365.
83. Nadiga, B.T.; Margolin, L.G. Dispersive-dissipative eddy parameterization in a barotropic model. *J. Phys. Oceanogr.* **2001**, *31*, 2525–2531.
84. Rahman, S.M.; Ahmed, S.E.; San, O. A dynamic closure modeling framework for model order reduction of geophysical flows. *Phys. Fluids* **2019**, *31*, 046602.

85. San, O.; Maulik, R. Extreme learning machine for reduced order modeling of turbulent geophysical flows. *Phys. Rev. E* **2018**, *97*, 042322.
86. Dellnitz, M.; Junge, O.; Lo, M.W.; Marsden, J.E.; Padberg, K.; Preis, R.; Ross, S.D.; Thiere, B. Transport of Mars-crossing asteroids from the quasi-Hilda region. *Phys. Rev. Lett.* **2005**, *94*, 231102.
87. Grover, P.; Ross, S.D.; Stremler, M.A.; Kumar, P. Topological chaos, braiding and bifurcation of almost-cyclic sets. *Chaos* **2012**, *22*, 043135.
88. Nolan, P.J.; Serra, M.; Ross, S.D. Finite-time Lyapunov exponents in the instantaneous limit and material transport. *Nonlinear Dyn.* **2020**, *100* 3825–3852.
89. Serra, M.; Sathe, P.; Rypina, I.; Kirincich, A.; Ross, S.D.; Lermusiaux, P.; Allen, A.; Peacock, T.; Haller, G. Search and rescue at sea aided by hidden flow structures. *Nat. Commun.* **2020**, *11*, 2525.