ELECTRONIC RESEARCH ARCHIVE
Volume **29**, Number **3**, August **2021**

doi:10.3934/era.2020128
pp. **2533–2552**

EFFICIENT COMPUTATIONS FOR LINEAR FEEDBACK CONTROL PROBLEMS FOR TARGET VELOCITY MATCHING OF NAVIER-STOKES FLOWS VIA POD AND LSTM-ROM

Hyung-Chun Lee*

Department of Mathematics, Ajou University
Suwon, 16499, Republic of Korea

Abstract. An efficient computing method for a target velocity tracking problem of fluid flows is considered. We first adopts the Lagrange multipliers method to obtain the optimality system, and then designs a simple and effective feedback control law based on the relationship between the control $\boldsymbol{f}$ and the adjoint variable $\boldsymbol{w}$ in the optimality system. We consider a reduced order modeling (ROM) of this problem for real-time computing. In order to improve the existing ROM method, the deep learning technique, which is currently being actively researched, is applied. We review previous research results and some computational results are presented.

1. **Introduction.** The problem of fluid flow control has long been the focus of fluid mechanics research and is still an important research topic for many researchers. One of the most important fluid control problems is target velocity matching or tracking issues for fluid flow. The issue of target velocity matching or tracking problems for fluid flows has been a very active area of research e.g. [1, 15, 17, 18, 19, 21, 22, 29, 31] as well as others. The computation of optimally controlled flows based on solving an optimality system is expensive in terms of CPU time and memory space. This is due to the fact that such an approach involves a coupled system of state and adjoint equations with initial and final conditions. The system has to be solved on the entire space-time domain and cannot be solved by marching in time. Therefore, a feedback control design and/or reduced order modeling (ROM) is required for efficient and real time computing.

Feedback control is a less expensive solution technique that gives qualitatively similar results to optimal control. Feedback control can effect good velocity tracking and at the same time the solution can be obtained step-by-step in time at the cost of a single flow solve. Mathematical theories and approximation techniques for optimal control problems for the Navier-Stokes equations have been developed in various areas of fluid dynamics; see, e.g., [1, 17, 21, 22]. Some numerical methods for solving control problems for unsteady flows have been proposed and tested; see, e.g., [17, 18, 19, 21, 22] and references therein. Many of these deal with distributed

2020 *Mathematics Subject Classification.* Primary: 76D55, 49M25, 49M41, 68T07; Secondary: 65M60.

Key words and phrases. Navier-Stokes equations, finite element, reduced order method, singular value decomposition, deep neural network, LSTM.

The author is supported by Basic Science Research Program through the National Research Foundation of Korea NRF-2016R1D1A1B03932219 and NRF-2019R1F1A1050231.

* Corresponding author.

controls and take different approaches to limiting the size of the control. Feedback controls for the Navier-Stokes equations have also been the subject of previous, mostly computational, studies; see, e.g., [11, 16] and references therein. In this article, we consider desired states $\boldsymbol{U}$ in $\mathbf{L}^2$ which are not necessarily divergence-free, which is a feature not seen in previous work.

Proper orthogonal decomposition (POD) is a common technique for extracting the dominant mode that contributes the most to the energy of the entire system [6]. POD combined with Galerkin projection (GP) have been used for many years to formulate ROMs for dynamic systems [5, 8, 9, 32, 24]. In such ROMs, the full-order set of equations is projected onto a reduced space, resulting in a dynamic system (modal coefficients) of much lower order than the full order model (FOM). However, in many situations there is a discrepancy between the governing equation and the observed system. This can be due to an approximation of the underlying phenomenon, incorrect parameterization, or insufficient information about the source terms contained in the system. This is especially evident in fluid flow systems where many complex phenomena and source terms interact in different ways. A common reduced order modeling (ROM) development procedure may be described by the following tasks:

1. Reduced space and basis identification.
2. Nonlinear dynamical system evolution in the reduced space.
3. Reconstruction in full-order space for assessments.

Machine learning (ML) tools have had considerable success in the fluid mechanics community, identifying basic structures and mimicking dynamics [3, 7, 25, 26]. However, modeling with ML, especially deep learning, has faced strict opposition from both academia and industry alike because it can produce non-physical results due to its black box nature and its lack of interpretability and generalization [10, 23]. In the course of conducting this study, we also experienced these points seriously. Even if the input and output data used to train the ML algorithm is physically accurate, the quantity interpolated with the ML approach can deviate significantly from a physically accurate solution. However, the fluid mechanics problem solving using deep learning is considered a promising research in the future, and we think it should be continued. A perspective on machine learning for advancing fluid dynamics can be found in a recent review articles [7] and references therein.

The plan of the rest of the paper is as follows. In Section 2, we define and discuss the linear feedback control. We derive a time-space discretized version of the feedback control of Navier-Stokes equations. In Section 3, we explain the POD reduced basis and Galerkin Projection ROM. In Section 4, we introduce a closure model using deep neural networks. Finally, in section 5 some numerical results will be given.

2. **A feedback control problem for Navier-Stokes equations.**

Let Ω be a bounded open set. We shall use the standard function spaces and their norms; for details see [2, 13]. For any nonnegative integer m, we define the Sobolev spaces $H^m(\Omega)$ by

$$H^m(\Omega) = \left\{u \in L^2(\Omega) : D^\alpha u \in L^2(\Omega) \text{ for } 0 \leq |\alpha| \leq m\right\},$$

where $D^\alpha u$ denotes the weak (or distributional) partial derivative and α is a multi-index, $|\alpha| = \sum_i \alpha_i$. Note that $H^0(\Omega) = L^2(\Omega)$. We equip $H^m(\Omega)$ with the norm

$$||u||_m^2 = \sum_{|\alpha|\le m} ||D^\alpha u||_0^2$$

The usual inner product associated with $H^m(\Omega)$ will be denoted by $(\cdot,\cdot)_m$. Let $H_0^m(\Omega)$ denote the closure of $C_0^\infty(\Omega)$ under the norm $||\cdot||_m$ and $H_0^{-m}(\Omega)$ be the dual space of $H_0^m(\Omega)$.

For vector-valued functions, we define the Sobolev space $\mathbf{H}^m(\Omega)$ (in all cases, boldface indicates vector-valued) by

$$\mathbf{H}^m(\Omega) = \{\boldsymbol{u} \,|\, u_i \in H^m(\Omega),\, i=1,2\},$$

where $\boldsymbol{u}=(u_1,u_2)$ and its associated norm by

$$||\boldsymbol{u}||_m = \left(\sum_{i=1}^{2} ||u_i||_m^2\right)^{1/2}.$$

We also define particular subspace

$$L_0^2(\Omega) = \left\{p \in L^2(\Omega) : \int_\Omega p\, d\Omega = 0\right\}.$$

We introduce the solenoidal spaces

$$\begin{aligned}
\boldsymbol{\mathcal{V}} &= \{\boldsymbol{u} \in \mathbf{C}_0^\infty(\Omega) : \nabla\cdot\boldsymbol{u} = 0\},\\
\mathbf{V} &= \{\boldsymbol{u} \in \mathbf{H}_0^1(\Omega) : \nabla\cdot\boldsymbol{u} = 0\},\\
\mathbf{W} &= \{\boldsymbol{u} \in \mathbf{L}^2(\Omega) : \nabla\cdot\boldsymbol{u} = 0\}.
\end{aligned}$$

The spaces $\mathbf{V}$ and $\mathbf{W}$ are closure of $\boldsymbol{\mathcal{V}}$ in $\mathbf{H}_0^1(\Omega)$ and $\mathbf{L}^2(\Omega)$, respectively. All subspaces are equipped with the norms inherited from the corresponding underlying spaces.

In order to define a weak form of the Navier-Stokes equations, we introduce the continuous bilinear form

$$a(\boldsymbol{u},\boldsymbol{v}) = 2\sum_{i,j=1}^{n}\int_\Omega D_{ij}(\boldsymbol{u})D_{ij}(\boldsymbol{v})d\Omega \quad \forall \boldsymbol{u},\boldsymbol{v} \in \mathbf{H}^1(\Omega), \tag{1}$$

$$b(\boldsymbol{v},q) = -\int_\Omega q\nabla\cdot\boldsymbol{v}d\Omega \quad \forall q \in L_0^2, \quad \forall \boldsymbol{v} \in \mathbf{H}^1(\Omega), \tag{2}$$

where $D(\boldsymbol{u}) = (1/2)(\nabla\boldsymbol{u} + \nabla\boldsymbol{u}^T)$ and the trilinear form

$$c(\boldsymbol{w};\boldsymbol{u},\boldsymbol{v}) = \sum_{i,j=1}^{n}\int_\Omega w_j\left(\frac{\partial u_i}{\partial x_j}\right)v_i d\Omega \quad \forall \boldsymbol{w},\boldsymbol{u},\boldsymbol{v} \in \mathbf{H}^1(\Omega). \tag{3}$$

A weak formulation of the Navier-Stokes equations is given by *given* $\boldsymbol{g} \in \mathbf{H}^{-1}(\Omega)$, *find* $(\boldsymbol{u},p) \in \mathbf{H}_0^1(\Omega)\times L_0^2(\Omega)$ *satisfying*

$$\begin{cases} \langle \boldsymbol{u}_t,\boldsymbol{v}\rangle + \nu a(\boldsymbol{u},\boldsymbol{v}) + c(\boldsymbol{u};\boldsymbol{u},\boldsymbol{v}) + b(\boldsymbol{v},p) = \langle \boldsymbol{g},\boldsymbol{v}\rangle,\\ b(\boldsymbol{u},q) = 0, \end{cases} \tag{4}$$

with initial velocity $\boldsymbol{u}_0$, where $\langle\cdot,\cdot\rangle$ denotes the duality paring between $\mathbf{H}_0^1(\Omega)$ and $\mathbf{H}_0^{-1}(\Omega)$.

We consider the target velocity fields $\boldsymbol{U} \in \mathcal{U}_{ad}$, where $\mathcal{U}_{ad}$ is defined by

$$\begin{cases} \boldsymbol{U} = \boldsymbol{U}(t,\boldsymbol{x}) \in C([0,T];\mathbf{H}^2(\Omega)\cap\mathbf{H}_0^1(\Omega)),\\ \nabla\cdot\boldsymbol{U}(t,\boldsymbol{x}) = 0 \quad \forall \boldsymbol{x} \in \Omega. \end{cases} \tag{5}$$

We will consider the target velocity matching problem in which we want to minimize

$$\int_0^T \int_\Omega |\boldsymbol{u} - \boldsymbol{U}|^2 \, d\Omega dt.$$

To do this, we define the following control and cost functional. Let $\boldsymbol{f} \in L^2((0,T); \mathbf{L}^2(\Omega))$ denote the distributed control. Given T, we define the functional

$$\mathcal{J}(\boldsymbol{u}, \boldsymbol{f}) = \int_0^T \int_\Omega \left(\frac{\alpha}{2} |\boldsymbol{u} - \boldsymbol{U}|^2 + \frac{\beta}{2} |\boldsymbol{f}|^2 \right) d\Omega dt + \frac{\delta}{2} \int_\Omega |\boldsymbol{u}(T) - \boldsymbol{U}(T)|^2 d\Omega. \tag{6}$$

The minimization of the $\int_0^T \int_\Omega |\boldsymbol{u} - \boldsymbol{U}|^2 d\Omega dt$ is the real goal of velocity matching problem, which is to keep the solution $\boldsymbol{u}$ to $\boldsymbol{U}$ over $(0,T) \times \Omega$. The $\int_0^T \int_\Omega \frac{\beta}{2} |\boldsymbol{f}|^2 d\Omega dt$ term is introduced in order to bound the control function and to prove the existence of the optimal control (regularization term). The $\int_\Omega |\boldsymbol{u}(T) - \boldsymbol{U}(T)|^2 d\Omega$ term is necessary in order to keep the solution $\boldsymbol{u}$ to $\boldsymbol{U}$ near the time T. An optimal control problem is defined by

seek $(\boldsymbol{u}, \boldsymbol{f}) \in L^2((0,T); \mathbf{H}_0^1(\Omega)) \times L^2((0,T); \mathbf{L}^2(\Omega))$ *such that the cost functional (6) is minimized subject to constraints*

$$\begin{cases} \langle \boldsymbol{u}_t, \boldsymbol{v} \rangle + \nu a(\boldsymbol{u}, \boldsymbol{v}) + c(\boldsymbol{u}; \boldsymbol{u}, \boldsymbol{v}) + b(\boldsymbol{v}, p) = \langle \boldsymbol{f}, \boldsymbol{v} \rangle \quad \forall \boldsymbol{v} \in \mathbf{H}_0^1(\Omega), \\ b(\boldsymbol{u}, q) = 0 \quad \forall L_0^2(\Omega), \end{cases} \tag{7}$$

with initial velocity $\boldsymbol{u}_0 \in \mathbf{H}_0^1(\Omega)$.

Using the Lagrange multipliers method, one can obtain the optimality system: *seek* $\boldsymbol{u} \in L^2((0,T); \mathbf{H}_0^1(\Omega)), p \in L^2((0,T); \mathbf{L}_0^2(\Omega)), \boldsymbol{w} \in L^2((0,T); \mathbf{H}_0^1(\Omega))$ *and* $r \in L^2((0,T); \mathbf{L}_0^2(\Omega))$ such that

$$\begin{cases} \langle \boldsymbol{u}_t, \boldsymbol{v} \rangle + \nu a(\boldsymbol{u}, \boldsymbol{v}) + c(\boldsymbol{u}; \boldsymbol{u}, \boldsymbol{v}) + b(\boldsymbol{v}, p) = \langle \boldsymbol{f}, \boldsymbol{v} \rangle \quad \forall \boldsymbol{v} \in \mathbf{H}_0^1(\Omega), \\ b(\boldsymbol{u}, q) = 0 \quad \forall L_0^2(\Omega), \end{cases} \tag{8}$$

with initial velocity $\boldsymbol{u}_0 \in \mathbf{H}_0^1(\Omega)$ and the adjoint equations

$$\begin{cases} -\langle \boldsymbol{w}_t, \boldsymbol{v} \rangle + \nu a(\boldsymbol{w}, \boldsymbol{v}) + c(\boldsymbol{w}; \boldsymbol{u}, \boldsymbol{v}) + c(\boldsymbol{u}; \boldsymbol{w}, \boldsymbol{v}) + b(\boldsymbol{v}, r) \\ \qquad\qquad = \alpha \langle (\boldsymbol{u} - \boldsymbol{U}), \boldsymbol{v} \rangle \quad \forall \boldsymbol{v} \in \mathbf{H}_0^1(\Omega), \\ b(\boldsymbol{w}, q) = 0 \quad \forall L_0^2(\Omega), \end{cases} \tag{9}$$

with final velocity $\boldsymbol{w}(T, \boldsymbol{x}) = \delta(\boldsymbol{u}(T) - \boldsymbol{U}(T))$ *and the optimality condition*

$$\boldsymbol{w} = -\beta \boldsymbol{f}. \tag{10}$$

Several treatments of similar optimal control problems can be found in the literature, most notably in [1, 17]. The numerical treatment of the velocity tracking problem is also an outstanding problem and other algorithms have been proposed. For example, a quasi-optimal control has been studied in [21, 22].

Remark 1. Thus the optimality system is a system of the coupled nonlinear partial differential equations (8)-(10). We already mentioned that solving an optimality system is very expensive in terms of CPU time and memory space. This is due to the fact that such an approach involves a coupled system of state and adjoint equations with initial and final conditions. The system has to be solved on the entire space-time domain and cannot be solved by marching in time. A gradient algorithm is needed and so the final algorithm is complex, involving multiple flow solutions, and the convergence may be slow.

2.1. **Feedback control.** From (10), we see that a control $\boldsymbol{f}$ is related linearly with the adjoint variable $\boldsymbol{w}$ which is a solution of the adjoint equations (9) with $\boldsymbol{f} = -(1/\beta)\boldsymbol{w}$. Also, we see that $\boldsymbol{w}$ is linearly dependent only on the source term of the adjoint equations (9) $\alpha(\boldsymbol{u}-\boldsymbol{U})$ since the adjoint equations are linear equations. So, we see that

$$\boldsymbol{f} \ \propto \ \frac{\alpha}{\beta}(\boldsymbol{u}-\boldsymbol{U}). \tag{11}$$

Usually, β is very small real number to get a small value of the first term $\int_0^T \int_\Omega |\boldsymbol{u}-\boldsymbol{U}|^2 d\Omega dt$ in (6).

$$\boldsymbol{f} = \gamma(\boldsymbol{u}-\boldsymbol{U}). \tag{12}$$

with very large value of γ. Remember that γ is closely related to β. The smaller β, the greater control $\boldsymbol{f}$. Increasing control is not a problem in numerical simulation, but it is expensive in actual control problems. Therefore, γ can be set according to the actual problem.

Now, we consider the feedback control problem

$$\begin{cases} \boldsymbol{u}_t + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} - \nu\nabla^2\boldsymbol{u} + \nabla p = \gamma(\boldsymbol{u}-\boldsymbol{U}) \quad \text{in } (0,T)\times\Omega, \\ \qquad\qquad\qquad\qquad\qquad\qquad \nabla\cdot\boldsymbol{u} = 0 \quad \text{in } (0,T)\times\Omega, \\ \quad \boldsymbol{u}(t,\boldsymbol{x}) = \boldsymbol{0} \quad \text{in } (0,T)\times\partial\Omega, \quad \boldsymbol{u}(0,\boldsymbol{x}) = \boldsymbol{u}_0 \quad \text{in } \Omega. \end{cases} \tag{13}$$

Remark 2. In [19], the control is achieved by means of a linear feedback law relating the body force to the velocity field, i.e., $\boldsymbol{f} = \boldsymbol{F} - \gamma(\boldsymbol{u}-\boldsymbol{U})$ where $\boldsymbol{U}$ is in the set of the admissible target velocity $\boldsymbol{U}_{ad}$ if $\boldsymbol{U}$ is a divergence free vector field in the set $\{\boldsymbol{c} : \boldsymbol{c} \in C((0,T);\mathbf{H}^2(\Omega)\cap\mathbf{H}_0^1(\Omega)) : \partial_t\boldsymbol{c} \in C((0,T);\mathbf{H}^1(\Omega))\}$. The corresponding body force generated by $\boldsymbol{U}$ is defined as

$$\boldsymbol{F}(t,\boldsymbol{u}) = \boldsymbol{U}_t(t,\boldsymbol{x}) - \nu\nabla^2\boldsymbol{U}(t,\boldsymbol{x}) + (\boldsymbol{U}(t,\boldsymbol{x})\cdot\nabla)\,\boldsymbol{U}(t,\boldsymbol{x}).$$

In this article, $\boldsymbol{U} \in L((0,T);\mathbf{L}^2(\Omega))$ so $\boldsymbol{F}$ can not be defined and we do not include $\boldsymbol{F}$ in the control. We will compare our results with those in [19]. Our computational result is almost the same as that of [19] depending on the value of γ (see Figure 3).

Then, a linear feedback control problem of Navier-Stokes equations can be formulated by

$$\begin{cases} (\boldsymbol{u}_t,\boldsymbol{v}) + \nu a(\boldsymbol{u},\boldsymbol{v}) + c(\boldsymbol{u};\boldsymbol{u},\boldsymbol{v}) + b(\boldsymbol{v},p) - \gamma(\boldsymbol{u},\boldsymbol{v}) = -\gamma(\boldsymbol{U},\boldsymbol{v}) \quad \forall\boldsymbol{v}\in\mathbf{H}_0^1(\Omega), \\ b(\boldsymbol{u},q) = 0 \quad \forall L_0^2(\Omega), \end{cases} \tag{14}$$

with initial velocity $\boldsymbol{u}_0 \in \mathbf{H}_0^1(\Omega)$

2.2. **The Navier-Stokes system and its finite element approximation.** A typical finite element approximation of (14) is defined as follows: we first choose conforming finite element subspaces $\mathbb{V}_h \subset \mathbf{H}^1(\Omega)$ and $S_h \subset L^2(\Omega)$ and then define $\mathbb{V}_{h,0} = \mathbb{V}_h \cap \mathbf{H}_0^1(\Omega)$, and $S_{h,0} = S_h \cap L_0^2(\Omega)$. One then seeks $\boldsymbol{u}_h(t,\cdot) \in \mathbb{V}_{h,0}$ and $p_h \in S_{h,0}$ such that

$$\begin{cases} \left(\dfrac{\partial\boldsymbol{u}_h}{\partial t},\boldsymbol{v}_h\right) + \nu a(\boldsymbol{u}_h,\boldsymbol{v}_h) + c(\boldsymbol{u}_h;\boldsymbol{u}_h,\boldsymbol{v}_h) + b(\boldsymbol{v}_h,p_h) - \gamma(\boldsymbol{u}_h,\boldsymbol{v}_h) \\ \qquad\qquad\qquad\qquad = -\gamma(\boldsymbol{U},\boldsymbol{v}_h) \quad \text{for all } \boldsymbol{v}_h \in \mathbb{V}_{h,0}, \\ \quad b(\boldsymbol{u}_h,q_h) = 0 \quad \text{for all } q_h \in S_{h,0}, \quad \boldsymbol{u}_h(0,\boldsymbol{x}) = \boldsymbol{u}_{0,h}(\boldsymbol{x}) \quad \text{in } \Omega, \end{cases} \tag{15}$$

where $\boldsymbol{u}_{0,h}(\boldsymbol{x}) \in \mathbb{V}_{h,0}$ is an approximation, e.g., a projection, of $\boldsymbol{u}_0(\boldsymbol{x})$. Discretization is completed by choosing a time-marching method such as the backward Euler scheme. See, e.g., [13] for details about finite element discretizations of the Navier-Stokes system.

Let $\sigma_N = \{t_n\}_{n=0}^N$ be a partition of $[0, T]$ into equal intervals $\Delta t = T/N$ with $t_0 = 0$ and $t_N = T$. Then a time discretization of (15) is given by

$$\begin{cases} \left(\dfrac{\boldsymbol{u}_h^{(n)} - \boldsymbol{u}_h^{(n-1)}}{\Delta t}, \boldsymbol{v}_h\right) + \nu a(\boldsymbol{u}_h^{(n)}, \boldsymbol{v}_h) + c(\boldsymbol{u}_h^{(n)}; \boldsymbol{u}_h^{(n)}, \boldsymbol{v}_h) + b(\boldsymbol{v}_h, p_h^{(n)}) \\ \qquad -\gamma(\boldsymbol{u}_h^{(n)}, \boldsymbol{v}_h) = -\gamma(\boldsymbol{U}, \boldsymbol{v}_h) \quad \text{for all } \boldsymbol{v}_h \in \mathbb{V}_{h,0}, \\ b(\boldsymbol{u}_h^{(n)}, q_h) = 0 \quad \text{for all } q_h \in S_{h,0}, \quad \boldsymbol{u}_h^{(0)}(\boldsymbol{x}) = \boldsymbol{u}_{0,h}(\boldsymbol{x}) \quad \text{in } \Omega. \end{cases} \tag{16}$$

Since the trilinear term is nonlinear, a linearization of the discrete equations (16) must be introduced in order to solve it by a iteration method. We solve the following linearized equations:

$$\begin{cases} \dfrac{1}{\Delta t}\left(\boldsymbol{u}_h^{(n)}(k), \boldsymbol{v}_h\right) + \nu a\left(\boldsymbol{u}_h^{(n)}(k), \boldsymbol{v}_h\right) + \sigma \tilde{c}\left(\boldsymbol{u}_h^{(n)}(k); \boldsymbol{u}_h^{(n)}(k-1), \boldsymbol{v}_h\right) \\ \quad + \tilde{c}\left(\boldsymbol{u}_h^{(n)}(k-1); \boldsymbol{u}_h^{(n)}(k), \boldsymbol{v}_h\right) + b\left(\boldsymbol{v}_h, p_h^{(n)}(k)\right) - \gamma\left(\boldsymbol{u}_h^{(n)}(k), \boldsymbol{v}_h\right) \\ \qquad = \dfrac{1}{\Delta t}\left(\boldsymbol{u}_h^{(n-1)}(k), \boldsymbol{v}_h\right) + \sigma \tilde{c}\left(\boldsymbol{u}_h^{(n)}(k-1); \boldsymbol{u}_h^{(n)}(k-1), \boldsymbol{v}_h\right) \\ \qquad\qquad -\gamma\left(\boldsymbol{U}, \boldsymbol{v}_h\right) \quad \text{for all } \boldsymbol{v}_h \in \mathbb{V}_{h,0}, \\ b\left(\boldsymbol{u}_h^{(n)}(k), q_h\right) = 0 \quad \text{for all } q_h \in S_{h,0}, \quad \boldsymbol{u}_h^{(0)}(\boldsymbol{x}) = \boldsymbol{u}_{0,h}(\boldsymbol{x}) \quad \text{in } \Omega, \end{cases} \tag{17}$$

where $\tilde{c}(\boldsymbol{u}; \boldsymbol{v}, \boldsymbol{w})$ is the modified trilinear form (see [12]) to preserve the antisymmetry of the trilinear form $c(\boldsymbol{u}; \boldsymbol{v}, \boldsymbol{w})$ on the finite element spaces

$$\tilde{c}(\boldsymbol{u}; \boldsymbol{v}, \boldsymbol{w}) = \frac{1}{2}\Big\{c(\boldsymbol{u}; \boldsymbol{v}, \boldsymbol{w}) - c(\boldsymbol{u}; \boldsymbol{w}, \boldsymbol{v})\Big\} \quad \forall \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w} \in \mathbf{H}_0^1(\Omega).$$

Given the velocity $\boldsymbol{u}_h^{(n-1)}$ and an integer m, one can generate the sequence $\{\boldsymbol{u}_h^{(n)}(k), p_h^{(n)}\}$ for $k = 1, 2, \dots$ by solving the linear problem with $\sigma = 0$ for $k \le m$ and $\sigma = 1$ for $k > m$. The algorithm terminates when the maximum value of

$$\frac{|\boldsymbol{u}_h^{(n)}(k) - \boldsymbol{u}_h^{(n-1)}(k)|}{|\boldsymbol{u}_h^{(n)}(k)|}$$

is less or equal to τ on Ω. The global convergence of the simple iteration method ($\gamma = 0$), and the rapid convergence of Newton's method ($\gamma = 1$) are combined in this numerical algorithm. The value of m is set to be two or three, depending on the time step Δt.

3. **Reduced order method using proper orthogonal decomposition.**

Let us briefly introduce the POD method following the formulation of Sirovich [30], Rathinam [28] and Burkardt et. al. [8, 9]. POD provides a method for finding the best approximating subspace to a given set of data. Originally POD was used as data representation technique. Proper orthogonal decomposition (POD), also known as Karhunen–Lóeve decomposition or principal component analysis, provides a technique for analyzing multidimensional data. This method essentially provides an orthonormal basis for representing the given data in a certain least squares optimal sense.

3.1. POD reduced-order bases. Given a discrete set of snapshot vectors $W = \{\boldsymbol{w}_n\}_{n=1}^N$ belonging to $\mathbb{R}^J$, where $N < J$, we form the $J \times N$ snapshot matrix $\mathbb{A}$ whose columns are the snapshot vectors $\boldsymbol{w}_n$:

$$\mathbb{A} = \big(\boldsymbol{w}_1 \ \boldsymbol{w}_2 \ \cdots \ \boldsymbol{w}_N\big).$$

Let

$$\mathbb{U}^T \mathbb{A} \mathbb{V} = \begin{pmatrix} \Sigma & 0 \\ 0 & 0 \end{pmatrix},$$

where $\mathbb{U}$ and $\mathbb{V}$ are $J \times J$ and $N \times N$ orthogonal matrices, respectively, and $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_{\widetilde{N}})$ with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\widetilde{N}}$ be the singular value decomposition of $\mathbb{A}$. Here, $\widetilde{N}$ is the rank of $\mathbb{A}$, i.e., the dimension of the snapshot set W, which would be less than N whenever the snapshot set is linearly dependent. It is well known [14] that if

$$\mathbb{U} = \big(\boldsymbol{\phi}_1 \ \boldsymbol{\phi}_2 \ \cdots \ \boldsymbol{\phi}_J\big) \qquad \text{and} \qquad \mathbb{V} = \big(\boldsymbol{\psi}_1 \ \boldsymbol{\psi}_2 \ \cdots \ \boldsymbol{\psi}_N\big),$$

then

$$\mathbb{A}\boldsymbol{\psi}_i = \sigma_i \boldsymbol{\phi}_i \qquad \text{and} \qquad \mathbb{A}^T \boldsymbol{\phi}_i = \sigma_i \boldsymbol{\psi}_i \qquad \text{for } i = 1, \ldots, \widetilde{N}$$

so that also

$$\mathbb{A}^T \mathbb{A}\boldsymbol{\psi}_i = \sigma_i^2 \boldsymbol{\psi}_i \qquad \text{and} \qquad \mathbb{A}\mathbb{A}^T \boldsymbol{\phi}_i = \sigma_i^2 \boldsymbol{\phi}_i \qquad \text{for } i = 1, \ldots, \widetilde{N}$$

so that σ_i^2, $i = 1, \ldots, \widetilde{N}$, are the nonzero eigenvalues of $\mathbb{A}^T\mathbb{A}$ (and also of $\mathbb{A}\mathbb{A}^T$) arranged in nondecreasing order. Note that the matrix $\mathbb{C} = \mathbb{A}^T\mathbb{A}$ is simply the correlation matrix for the set of snapshot vectors $W = \{\boldsymbol{w}_n\}_{n=1}^N$, i.e., we have that $\mathbb{C}_{mn} = \boldsymbol{w}_m^T \boldsymbol{w}_n$.

In the reduced-order modeling context, given a set of snapshots $W = \{\boldsymbol{w}_n\}_{n=1}^N$ belonging to $\mathbb{R}^J$, the POD reduced-basis of dimension $K \leq N < J$ is the set $\{\boldsymbol{\phi}_k\}_{k=1}^K$ of vectors also belonging to $\mathbb{R}^J$ consisting of the first K left singular vectors of the snapshot matrix $\mathbb{A}$. Thus, one can determine the POD basis by computing the (partial) singular value decomposition of the $J \times N$ matrix $\mathbb{A}$. Alternately, one can compute the (partial) eigensystem $\{\sigma_k^2, \boldsymbol{\psi}_i\}_{i=1}^K$ of the $N \times N$ correlation matrix $\mathbb{C} = \mathbb{A}^T\mathbb{A}$ and then set $\boldsymbol{\phi}_k = \frac{1}{\sigma_k}\mathbb{A}\boldsymbol{\psi}_k$, $k = 1, \ldots, K$.

The K-dimensional POD basis has the obvious property of orthonormality. It also has several other important properties which we now mention. Let $\{\boldsymbol{s}_k\}_{k=1}^K$ be an arbitrary set of K orthonormal vectors in $\mathbb{R}^J$ and let $\Pi\boldsymbol{w}$ denote the projection of a vector $\boldsymbol{w} \in \mathbb{R}^J$ onto the subspace spanned by that set. Further, let

$$\mathcal{E}(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_K) = \sum_{n=1}^N |\boldsymbol{w}_n - \Pi\boldsymbol{w}_n|^2,$$

i.e., $\mathcal{E}$ is the sum of the squares of the error between each snapshot vector $\boldsymbol{w}_n$ and its projection $\Pi\boldsymbol{w}_n$ onto the span of $\{\boldsymbol{s}_k\}_{k=1}^K$. Then, it can be shown that

$$\left\{ \begin{array}{l} \textit{the POD basis } \{\boldsymbol{\phi}_k\}_{k=1}^K \textit{ minimizes } \mathcal{E} \textit{ over all possible} \\ K\textit{-dimensional orthonormal sets in } \mathbb{R}^J. \end{array} \right. \tag{18}$$

In fact, often the POD basis corresponding to a set of snapshots $W = \{\boldsymbol{w}_n\}_{n=1}^N$ and then its relation to the singular value decomposition of the matrix $\mathbb{A}$ or to the eigenvalue decomposition of $\mathbb{A}^T\mathbb{A}$ are derived properties. We note that $\mathcal{E}(\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_K)$

is referred to as the "POD energy" or "error in the POD basis". Also, it can be shown that

$$\mathcal{E}(\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_K) = \sum_{k=K+1}^{\widetilde{N}} \sigma_k^2, \tag{19}$$

i.e., the error in the POD basis is simply the sum of the squares of the singular values corresponding to the neglected POD modes.

3.2. **Galerkin Projection-ROM (GP-ROM).** We now show how a POD basis is used to define a reduced-order model for the Navier-Stokes system. For the sake of brevity, we only discuss the case for which the snapshot set is viewed as a set of finite element coefficient vectors; the case for which the snapshot set is a set of finite element functions proceeds in an almost identical manner.

Let $\{\boldsymbol{\phi}\}_{k=1}^K$ be a K-dimensional POD basis corresponding to the snapshot set $\{\boldsymbol{w}_n\}_{n=1}^N$ and let

$$\mathbf{U}_K = \operatorname{span}\{\boldsymbol{\phi}_k\}_{i=1}^K \subset \mathbb{V}_h.$$

We then determine $\boldsymbol{u}_h^K(t,\cdot) \in \mathbf{U}_K$ from the discrete problem

$$\begin{cases} \left(\dfrac{\partial \boldsymbol{u}_h^K}{\partial t}, \boldsymbol{v}\right) + \nu a(\boldsymbol{u}_h^K, \boldsymbol{v}) + c(\boldsymbol{u}_h^K; \boldsymbol{u}_h^K, \boldsymbol{v}) - \gamma(\boldsymbol{u}_h^K, \boldsymbol{v}) = -\gamma(\boldsymbol{U}, \boldsymbol{v}) \ \forall \boldsymbol{v} \in \mathbf{U}_K \\ \hfill (\boldsymbol{u}_h^K(0,\boldsymbol{x}), \boldsymbol{v}) = (\boldsymbol{u}_{0,h}(\boldsymbol{x}), \boldsymbol{v}) \forall \boldsymbol{v} \in \mathbf{U}_K. \end{cases} \tag{20}$$

For a given reduced space $\mathbf{U}_K$, GP-ROM finds the approximation of the velocity field spanned by the low-dimensional space,

$$\boldsymbol{u} \approx \boldsymbol{u}_h^K(t,\boldsymbol{x}) \equiv \sum_{k=1}^K a_k(t) \boldsymbol{\phi}_k(\boldsymbol{x}) \tag{21}$$

where $\{a_k(t)\}_{k=1}^K$ are the sought time-varying coefficients. The GP-ROM can be obtained by projecting the FOM (Full order finite element model) onto the POD space:

$$\begin{cases} \displaystyle\sum_{k=1}^K \frac{d}{dt} a_k(t)(\boldsymbol{\phi}_k, \boldsymbol{\phi}_\ell) + 2\nu \sum_{k=1}^K a_k(t)\big(D(\boldsymbol{\phi}_k), D(\boldsymbol{\phi}_\ell)\big) \\ \displaystyle\qquad + \left(\sum_{m=1}^K a_m(t)\boldsymbol{\phi}_m \cdot \nabla \sum_{k=1}^K a_k(t)\boldsymbol{\phi}_k \,,\, \boldsymbol{\phi}_\ell\right) - \gamma \sum_{k=1}^K a_k(t)(\boldsymbol{\phi}_k, \boldsymbol{\phi}_\ell) \\ = -\gamma \ \sum_{k=1}^K b_k(\boldsymbol{\phi}_k, \boldsymbol{\phi}_\ell), \\ \displaystyle\sum_{k=1}^K a_k(0)(\boldsymbol{\phi}_k, \boldsymbol{\phi}_\ell) = (\boldsymbol{u}_0, \boldsymbol{\phi}_\ell), \quad \sum_{k=1}^K b_k(\boldsymbol{\phi}_k, \boldsymbol{\phi}_l) = (\boldsymbol{U}, \boldsymbol{\phi}_\ell), \end{cases} \tag{22}$$

for $\ell = 1, \ldots, K$. Equivalently, we have the system of nonlinear ordinary differential equations that determine the coefficient functions $\{a_k(t)\}_{k=1}^K$:

$$\begin{cases} \mathbb{G}\dfrac{d}{dt}\boldsymbol{a}(t) + \mathbb{K}\boldsymbol{a}(t) + (\boldsymbol{a}(t))^T \mathbb{N}\boldsymbol{a}(t) - \gamma\,\mathbb{G}\boldsymbol{a}(t) = -\gamma\,\mathbb{G}\boldsymbol{b} \\ \mathbb{G}\boldsymbol{a}(0) = \boldsymbol{a}_0, \end{cases} \tag{23}$$

or

$$\begin{cases} \dfrac{d}{dt}\boldsymbol{a}(t) + (\mathbb{G}^{-1}\mathbb{K} - \gamma\mathbb{I})\boldsymbol{a}(t) + \mathbb{G}^{-1}\boldsymbol{a}(t)^T \mathbb{N}\boldsymbol{a}(t) = -\gamma\,\boldsymbol{b} \\ \boldsymbol{a}(0) = \mathbb{G}^{-1}\boldsymbol{a}_0, \end{cases} \tag{24}$$

where the Gram matrix $\mathbb{G}$, stiffness matrix $\mathbb{K}$, convection tensor $\mathbb{N}$, and solution vector $\boldsymbol{a}(t)$ are respectively given by

$$\begin{aligned}
&\mathbb{G}_{\ell k} = \int_\Omega \boldsymbol{\phi}_k \cdot \boldsymbol{\phi}_\ell \, d\Omega \, \propto \, \mathbb{I}_K, \qquad \mathbb{K}_{\ell k} = 2\nu \int_\Omega D(\boldsymbol{\phi}_k) : D(\boldsymbol{\phi}_\ell)\, d\Omega, \\
&\mathbb{N}_{\ell m k} = \int_\Omega (\boldsymbol{\phi}_m \cdot \nabla)\boldsymbol{\phi}_k \cdot \boldsymbol{\phi}_\ell \, d\Omega, \qquad \text{and} \qquad (\boldsymbol{a})_k = a_k(t)
\end{aligned} \tag{25}$$

for $k, \ell, m = 1, \ldots, K$. Note that all of these matrices and tensors are full; however, since K will be chosen small, this does not cause any computational inefficiencies.

4. **Closure model using least squares projection and LSTM.**

It is easy to see that the computational cost of GP-ROM (24) is $O(K^3)$, which limits the number of modes to be used in the ROM. Several efforts have been devoted to introduce stabilization and closure techniques to account for the effects of truncated modes on ROM's dynamics. In this study, we utilize a LSTM closure model to improve the accuracy and efficiency for our computations.

The model reduction error $\mathcal{E}^{\rm ROM}(t, \boldsymbol{x})$ is equal to

$$\begin{aligned}
\mathcal{E}^{\rm ROM}(t, \boldsymbol{x}) &= \boldsymbol{u}^{\rm FOM}(t, \boldsymbol{x}) - \boldsymbol{u}^{\rm ROM}(t, \boldsymbol{x}) \\
&= \boldsymbol{u}^{\rm FOM}(t, \boldsymbol{x}) - \boldsymbol{u}^{\rm Proj}(t, \boldsymbol{x}) + \boldsymbol{u}^{\rm Proj}(t, \boldsymbol{x}) - \boldsymbol{u}^{\rm ROM}(t, \boldsymbol{x}) \\
&= \mathcal{E}_{\boldsymbol{U}_K{}^\perp}(t, \boldsymbol{x}) + \mathcal{E}_{\boldsymbol{U}_K}(t, \boldsymbol{x}).
\end{aligned} \tag{26}$$

The first term $\mathcal{E}_{\boldsymbol{U}_K{}^\perp}$ is the projection error that is a result of neglecting the components of $\boldsymbol{u}^{\rm FOM}(t, \boldsymbol{x})$ that lie in the space orthogonal to $\mathbf{U}_K$. This error is represented as $\mathcal{E}_{\boldsymbol{U}_K{}^\perp}$ in Fig. 1. The second term $\mathcal{E}_{\boldsymbol{U}_K}$ is the reduced order modeling error that results from solving a different dynamical system than the original. This error vector lies in the subspace $\mathbf{U}_K$ spanned by the reduced order basis $\{\boldsymbol{\phi}_k\}_{i=1}^K$. This error represents $\mathcal{E}_{\boldsymbol{U}_K}$ in Fig. 1. For more details, one can refer the articles [3, 4] .

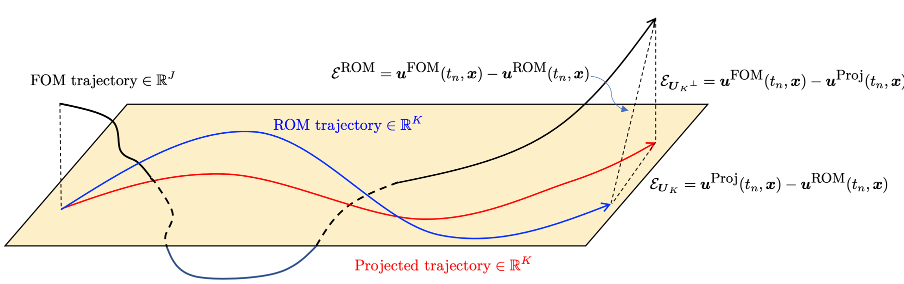

FIGURE 1. *Discrepancy of the trajectories between the FOM, ROM and least squares projected solutions after Galerkin projection*

The GP-ROM equations (24) with linear and nonlinear operators can be written as:

$$\dot{\boldsymbol{a}} = -\left(\mathcal{L} - \gamma \mathcal{I}\right)\boldsymbol{a} - \boldsymbol{a}^T \mathcal{N} \boldsymbol{a} - \gamma \boldsymbol{b}, \tag{27}$$

where $\mathcal{L}$, $\mathcal{I}$ and $\mathcal{N}$ are the linear, identity and nonlinear operators in (22), respectively. The above equation (27) can be written more explicitly

$$\frac{da_k(t)}{dt} = -\sum_{i=1}^{K}(\mathcal{L}_k^i - \gamma)a_i(t) - \sum_{i=1}^{K}\sum_{j=1}^{K}\mathcal{N}_k^{ij} a_i(t) a_j(t) - \gamma b_k(t), \;\; k = 1, 2, \ldots, K, \tag{28}$$

which can be numerically solved using time-stepping integration,

$$a_k(t_{n+1}) = a_k(t_n) - \Delta t \sum_{q=0}^{s} \beta_q \, G(a_k(t_{n-q})) - \beta_q \gamma b_k(t) \quad n = 0, \dots, J-1, \tag{29}$$

where s and β_q depend on the numerical scheme. Since this problem is a non-stiff problem, we usually use the fourth-order Adams-Bashforth method, e.g., ode113 in Matlab. From (29), we see that

$$G(a_k(t_n)) = -\sum_{i=1}^{K} \left(\mathcal{L}_k^i - \gamma\right) a_i(t_n) - \sum_{i=1}^{K}\sum_{j=1}^{K} \mathcal{N}_k^{ij} a_i(t_n) a_j(t_n). \tag{30}$$

To reduce the error $\mathcal{E}^{\mathrm{ROM}}$, we use a closure model which can be written as:

$$\frac{da_k(t)}{dt} = -\sum_{i=1}^{K}(\mathcal{L}_k^i - \gamma) a_i(t) - \sum_{i=1}^{K}\sum_{j=1}^{K} \mathcal{N}_k^{ij} a_i(t) a_j(t) - \gamma b_k(t) + C_k(a_1, \dots, a_K, \alpha_1, \dots, \alpha_K), \tag{31}$$

where $\alpha_k(t)$ is the Least-Squares (LS) projection modal coefficients which can be obtained by:

$$\alpha_k(t_n) = \langle \boldsymbol{u}(t_n, \boldsymbol{x}), \boldsymbol{\phi}_k \rangle, \quad n = 1, \dots, J, \tag{32}$$

where the angle parentheses refer to the Euclidean inner product in $\mathbb{R}^K$. The LS projection modal coefficients include the hidden physics and its interaction with the dynamical core of the system. For simplicity, we assume that C_k depend only on a_k and α_k for each k. The correction function C_k can be defined in various way, especially and naturally,

$$C_k(t_{n+1}) = \alpha_k(t_n) - a_k(t_n) \tag{33}$$

and

$$C_k(t_{n+1}) = \alpha_k(t_{n+1}) - \Big[\alpha_k(t_n) + \Delta t \sum_{q=0}^{s} \beta_q \, G(\alpha_k(t_{n-q}))\Big], \tag{34}$$

which are used in [25] and [26], respectively. In theory, they are different but we can not find the differences in computational performance. So, we will study more on these problem in the later articles. In this article, we adopt the equation (33) to the correction step in each time step t_n. One can use any of the suitable machine learning algorithm to learn this correction term C_k such as ResNET and LSTM network, so on. In this article, we employ LSTM neural network algorithm to learn the mapping from LS projection modal coefficients to the correction term

$$\{\alpha_1, \dots, \alpha_K\} \in \mathbb{R}^K \mapsto \{C_1, \dots, C_K\} \in \mathbb{R}^K.$$

5. Computational procedure and numerical results.

5.1. Full order finite element solutions.

Let $\sigma_N = \{t_n\}_{n=0}^N$ be a partition of $[0, T]$ into equal intervals $\Delta t = T/N$ with $t_0 = 0$ and $t_N = T$. The finite element spaces are chosen to be piecewise quadratic for the velocity and linear for the pressure, i.e., the Taylor-Hood finite element pair, based on a rectangular mesh. We use FreeFEM++ [20].

In this test we are interested in the convergence history for the parameters involved and so a simple stationary target velocity $\boldsymbol{U} = (U, V)$ is chosen, where

$$\begin{aligned} U(x,y) &= 10\frac{d\phi}{dy}(0.4,x,y), \quad V(x,y) = -10\frac{d\phi}{dx}(0.4,x,y), \\ \phi(t,x,y) &= (1-\cos(2\pi tx))(1-x)^2(1-\cos(2\pi ty))(1-y)^2. \end{aligned} \tag{35}$$

For the comparison purpose with the results in [19], we set up the initial velocity as follows:

$$\begin{cases} u_0(x,y) = -10U(x,y), \\ v_0(x,y) = -10V(x,y). \end{cases} \tag{36}$$

This initial velocity rotates in the opposite direction to the target velocity $\boldsymbol{U}$. For this calculation, we use $\Delta t = 0.0001$, $h = 1/64$ and $T = 1.0$. The number of unknowns (degree of freedom) is 49923.

The flow evolution is given in Fig. 2. The controlled fluid is on the left and the desired flow is on the right. All the figures are normalized. At the beginning, we have a reduction in the magnitude, then we have a change in shape, and finally a change in magnitude again. This evolution is typical of linear feedback control. As shown in Fig. 2, the change in shape is so quick that it is difficult to see the full evolution without using very small time steps. The error $||\boldsymbol{u}(t) - \boldsymbol{U}(t)||$ rapidly goes to zero and a good match is reached at $t = 0.5$. To compare computational results for our feedback law with those in [19], we compute $||\boldsymbol{u}(t) - \boldsymbol{U}(t)||_{\mathbf{L}^2(\Omega)}$ for $\gamma = 10, 30$. Fig. 3 shows the L^2 norm $||\boldsymbol{u}(t) - \boldsymbol{U}(t)||_{\mathbf{L}^2(\Omega)}$ between the calculated discretized controlled flow (here denoted by $\boldsymbol{u}$ instead of $\boldsymbol{u}_h$) and the exact target flow $\boldsymbol{U}$. We see that our feedback law is almost same effect to that of [19] .

5.2. **POD-ROM.** Now, we set up the Reynolds number as $Re = 10$ and the initial velocity for our computational experiments as follows:

$$\begin{cases} u_0(x,y) = -U(x,y), \\ v_0(x,y) = -V(x,y). \end{cases} \tag{37}$$

We generate the data snapshots for $\gamma = [10, 20, 30, 40]$ with a time-step of 0.001 from time $t = 0$ to $T = 0.25$. The $||\boldsymbol{u}(t) - \boldsymbol{U}(t)||_{\mathbf{L}^2(\Omega)}$ for each cases of γ is shown in Fig. 4.

The POD reduced basis is determined from the corresponding snapshot set as described in Section 3. Note that each basis function satisfies the discretized continuity equation, i.e., it is discretely solenoidal. The eight-dimensional POD basis functions are displayed in Fig. 5 (Upper two rows). We see that the difference of basis between those of $\gamma = 10$ and $\gamma = 40$.

We can compute the energy retained by POD basis functions using a relative information content (RIC) formula as given below:

$$\mathrm{RIC}(K) = \frac{\sum_{i=1}^{K} \sigma_j^2}{\sum_{i=1}^{\tilde{N}} \sigma_j^2}, \tag{38}$$

where σ's are singular values and $\tilde{N}$ is the total number of singular values. RIC represents the fraction of information of the the total data that can be recovered using K basis functions.

Fig. 6 displays the convergence of relative information content with respect to the number of POD basis functions used to represent the reduced order system for

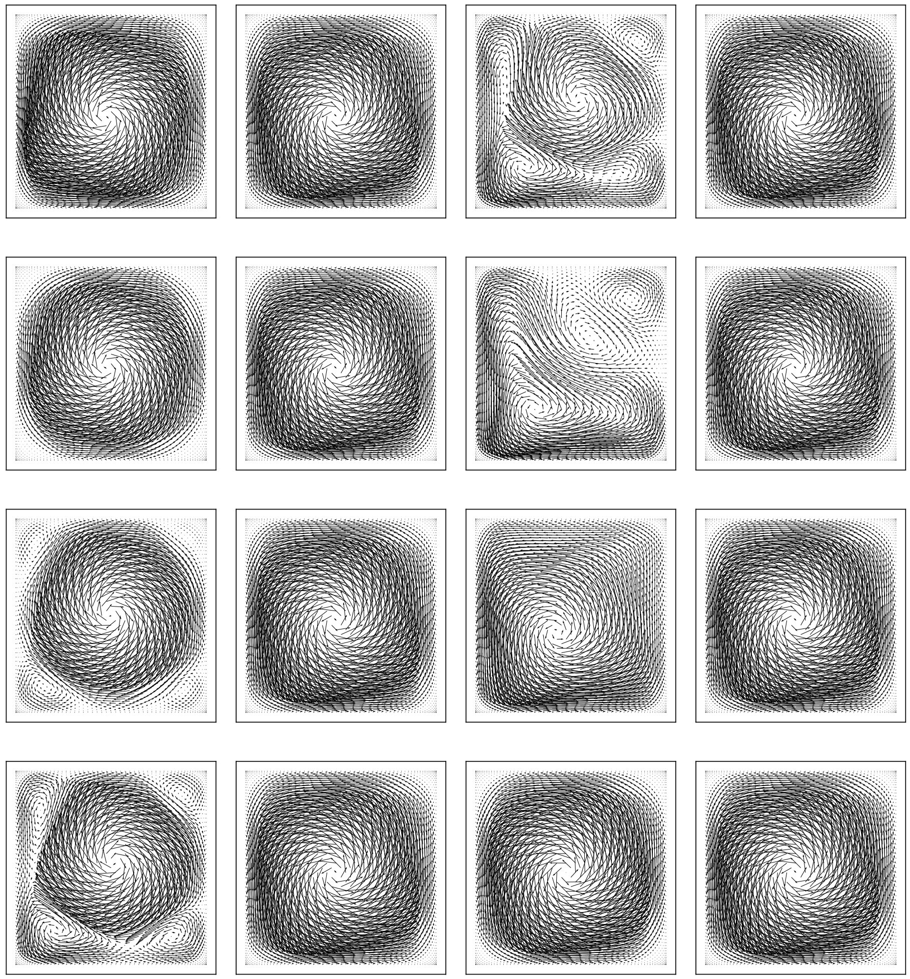

FIGURE 2. Controlled (left) and target (right) flows at $t = 0.0, 0.050, 0.057, 0.059$ (left pair of columns) and $t = 0.060, 0.061, 0.064, 0.5$ (right pair of columns) for $\gamma = 10$.

different control number γ. For the snapshot set determined as described in Section 2.2 with $\gamma = [10, 20, 30, 40]$, the first 8 singular values of the corresponding snapshot matrix are listed in Table 1.

	singular vales	RIC($\gamma = 40$)		singular vales	RIC($\gamma = 40$)
1	4.83907e+02	99.2476%	2	2.41493e+00	99.8174%
3	3.98155e-01	99.9207%	4	2.02669e-01	99.9731%
5	6.01101e-02	99.9909%	6	1.65390e-02	99.9964%
7	6.20453e-03	99.9985%	8	2.51213e-03	99.9994%

TABLE 1. The first 8 singular values of the snapshot matrix

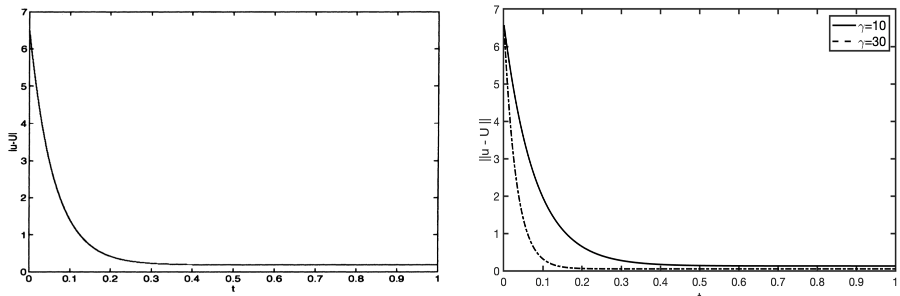

FIGURE 3. $||\boldsymbol{u}(t)-\boldsymbol{U}(t)||_2$, Compare between Figure 1 in [19] (left) and our results (right) where $Re = 1$

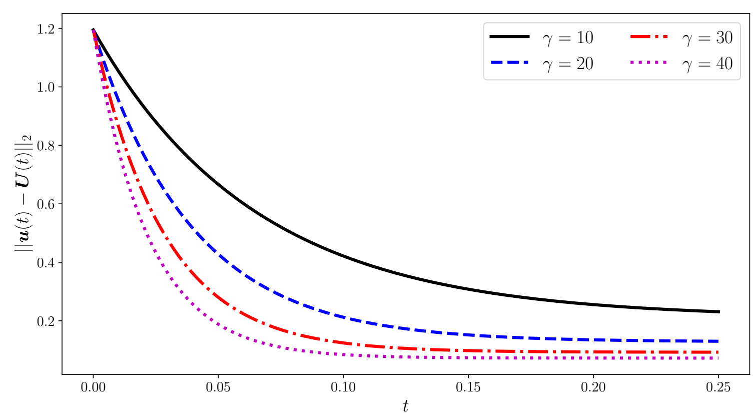

FIGURE 4. $||\boldsymbol{u} - \boldsymbol{U}||_{\mathbf{L}^2(\Omega)}$ for different control values γ.

We retained eight basis functions (i.e., $K = 8$) as they captured more than 99.997% of the energy for all control numbers $\gamma = [10, 20, 30, 40]$ (see Fig. 6 and Table 1 for $\gamma = 40$). It can be seen that the singular values for the feedback control problem decreases rapidly. We consider here, a "small-dimensional" POD basis can capture most of the information contained in the snapshot set.

Now, given the desired state $\boldsymbol{U}$, we use K-dimensional system of of nonlinear ordinary differential equations (24) to determine reduced-order solutions of the Navier-Stokes system. Approximations of solutions $\boldsymbol{a}_k^{(n)}$ of the system of ordinary differential equation (27) are determined using a fourth order Adams-Moulton method. To quantify the performance of different frameworks, we defined the root mean squared error (RMSE) between the FOM velocity field and the velocity field computed with different ROM frameworks. The RMSE is defined as:

$$\mathrm{RMSE}(t_n) = \sqrt{\frac{1}{n_x n_y}\sum_{i=1}^{n_x}\sum_{j=1}^{n_y}\Big(\boldsymbol{u}^{\mathrm{FOM}}(x_i, y_j, t_n) - \boldsymbol{u}^{\mathrm{ROM}}(x_i, y_j, t_n)\Big)^2} \tag{39}$$

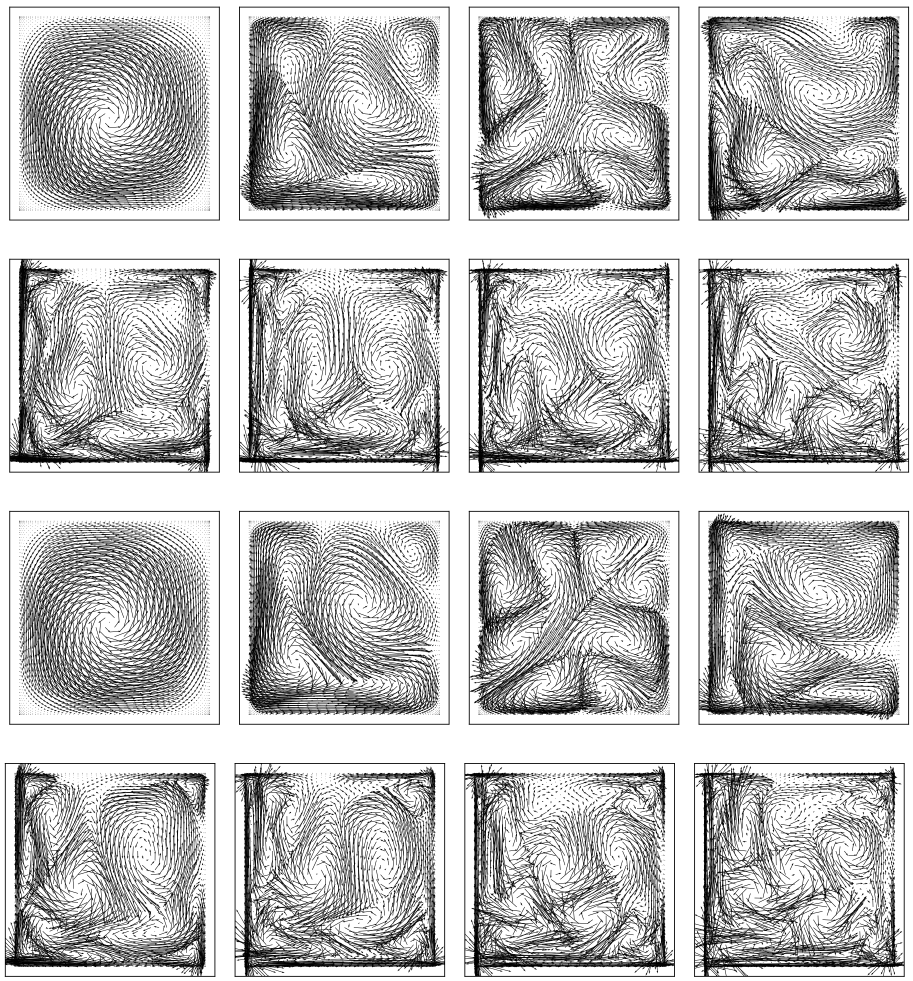

FIGURE 5. The POD reduced basis of cardinality 8 ($\gamma = 10$ Upper two rows and $\gamma = 40$ Down two rows). Basis 1,2,3,4 from the left to the right (Top) and Basis 5,6,7,8 from the left to the right (Bottom) in each two rows.

and

$$\text{RMSE}_T = \frac{1}{n_t}\sum_{n=1}^{n_t} \text{RMSE}(t_n). \tag{40}$$

Fig. 7 shows the RMSE(t) over time for four cases $\gamma = [10, 20, 30, 40]$. The LS Projection error is $\mathcal{E}_{\boldsymbol{U}_K{}^\perp}(t, \boldsymbol{x}) = \boldsymbol{u}^{\text{FOM}}(t, \boldsymbol{x}) - \boldsymbol{u}^{\text{Proj}}(t, \boldsymbol{x})$ and GP-ROM error is $\mathcal{E}^{\text{ROM}}(t, \boldsymbol{x}) = \boldsymbol{u}^{\text{FOM}}(t, \boldsymbol{x}) - \boldsymbol{u}^{\text{ROM}}(t, \boldsymbol{x})$. We calculate the LS projection modal coefficients by orthogonally projecting the state variables onto the reduced spaces. We will use the differences between RSME of LS projection and that of GP-ROM in the LSTM-ROM study, which will be studied in the next subsection.

5.3. **LSTM-ROM.** In this subsection, we test GP-LSTM closure model for the out-of-sample condition $\gamma = 50$. For testing our GP-LSTM closure model, we have generated four cases in different values of parameters such that $\gamma = [10, 20, 30, 40]$.

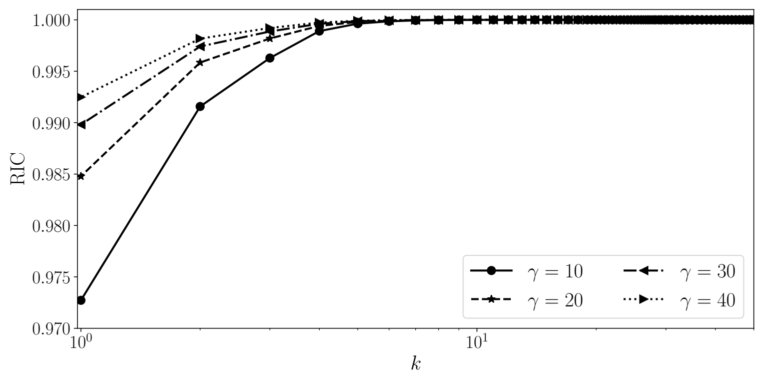

FIGURE 6. The singular values of the snapshot data matrix (right) for different control values γ.

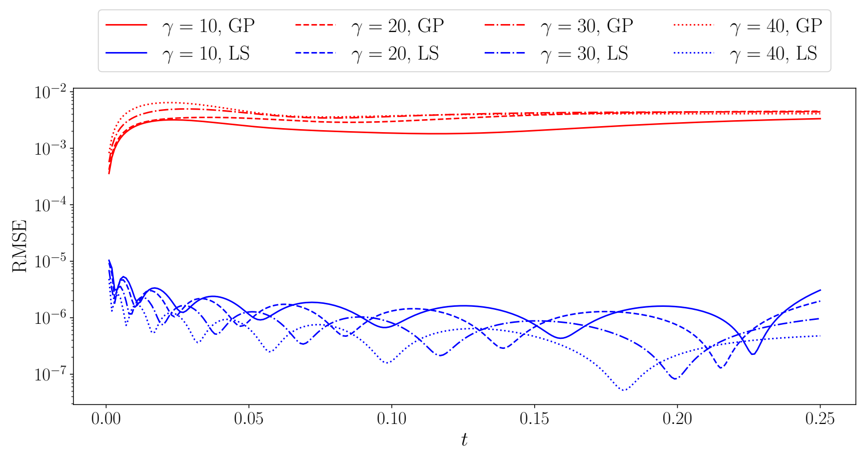

FIGURE 7. RMSE of Galerkin ROM (Upper 4 lines) and LS Projection (Lower 4 lines)

The LS projection modal coefficients include the hidden physics and its interaction with the dynamical core of the system. We can then define the correction term as:

$$\text{Correction} := C_k^{(n)} = \alpha_k^{(n)} - a_k^{(n+1)} \tag{41}$$

A supervised learning framework is applied to model the correction term C with information obtained from FOM and projection data. We train our LSTM neural network to learn the mapping from GP modal coefficients to the correction term. Since GP modal coefficients is used as input features to the LSTM network, the parameter γ governing the system's behavior is taken implicitly into account. We train the LSTM network with two hidden layers and 120 cells.

Fig. 8 shows a sketch of Long Short-Term Memory unit and an architecture of LSTM network training. For more details for LSTM architecture, one can refer to the articles [27] and references therein. Programs obtained from GitHub, namely

ETC-ROM Master, Hybrid-Modeling Master, UROM Master and mnni-rom Master, are adopted, modified and used in the learning process.

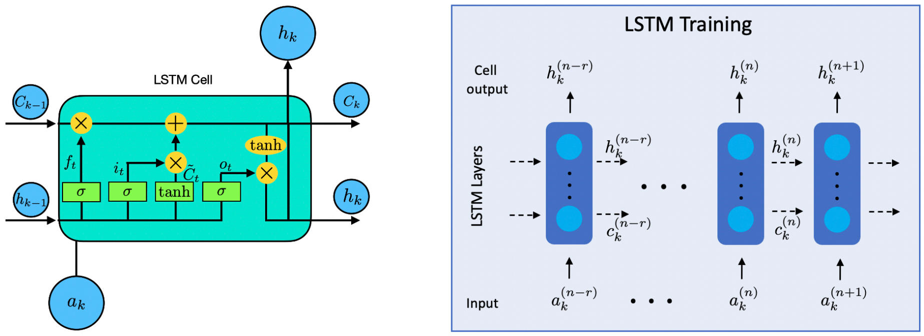

FIGURE 8. LSTM Cell (left) and LSTM Training where $r = 5$.

Once the model is trained, we could correct the GP modal coefficients with LSTM-based correction to approximate true projection modal coefficients.

$$a_k^{(n+1)} = a_k^{(n+1)} - \Delta t \left(\sum_{i=1}^{K} (\mathcal{L}_k^i - \gamma) a_i^{(n)} + \sum_{i=1}^{K}\sum_{j=1}^{K} \mathcal{N}_k^{ij} a_i^{(n)} a_j^{(n)} + \gamma b_k^{(n)} + C_k^{(n+1)} \right)$$

We give the numerical procedure as follows:

1. Data Generation: For $\gamma = 10, 20, 30, 40$ to use in learning algorithm and $\gamma = 50$ to use a test.
2. Basis Construction: For $\gamma = 10, 20, 30, 40$.
3. The velocity modal coefficients $a_k^{(n)}$ of Galerkin projection reduced order modeling calculation for $\gamma = 10, 20, 30, 40$ using each corresponding basis.
4. LSTM Training using $C_k^{(n+1)}$ and $a_k^{(n+1)}$. A summary of the adopted hyperparameters is presented in Table 2. The training and validation loss during the learning procedure are in Fig. 9.
5. Prediction for $\gamma = 50$ using the basis which is obtained from the snapshot sets with $\gamma = 40$ (see Fig. 10).
6. Reconstruction and compare with GP-ROM and LS Projection (see Fig. 11).

In step 5, interpolation techniques such as Grassman manifold interpolation [3, 4, 25], or the discrete empirical interpolation method, (DEIM) are typically applied to postprocess the results. For this work, however, such interpolations were not applied, so as to focus on the effects obtained from the Deep Learning technique. Thus, we use the same basis which is obtained from the snapshot sets with $\gamma = 40$. The POD modal coefficients predicted by these approaches are in Figure 10. As shown in the figure 10, it can be seen that the error of GP-LSTM is much smaller than that of GP-ROM.

Finally, we report the "online" computing time for our numerical experiments. We show the computational time as well as the RMSE of reconstructed fields at the final time $t = 0.25$ for the test cases at $\gamma = 50$ in Table 3. All calculations were performed on an iMac 3.6Ghz 8 Core. We observe that computing time of GP-LSTM is about twice that of GP-ROM(8) and the error of GP-LSTM(8) is

Variables	Hyperparameters
Number of hidden layers	2
Number of neurons in each hidden layer	120
Number of lookbacks	5
Batch size	32
Epochs	1000
Activation functions in the LSTM layers	tanh
Validation data set	20%
Loss function	MSE
Optimizer	ADAM

TABLE 2. A list of hyperparameters utilized to train the LSTM network for numerical experiments

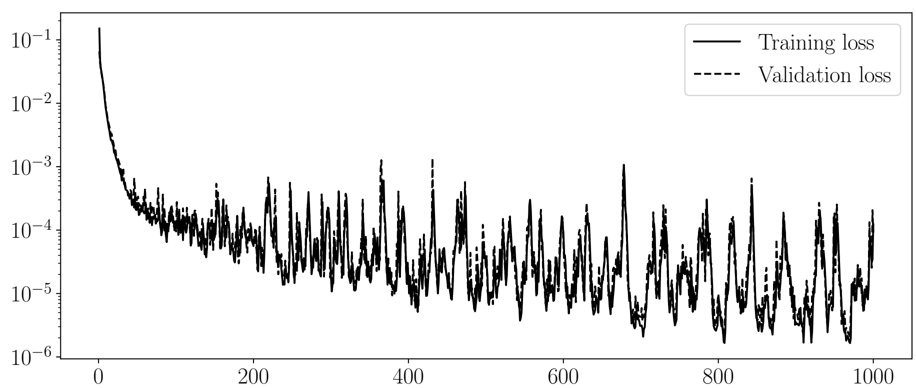

FIGURE 9. Training and validation loss

smaller (about 1/10) than that of GP-ROM(8). If we use as much data as and use interpolations like in other papers, for example ([3, 25, 26]), one can reduce the error even more. However, in this papers, LSTM is applied in the simplest way, to see the possibility of applying the deep learning technique to the optimal control problem or UQ problem of fluid flows.

Framework	Times	RMSE(0.25)
FOM	1695.7	0
LS-Proj		1.34053e-05
GP-ROM(8)	0.3268	3.77711e-03
GP-ROM (16)	1.7012	1.28362e-03
GP-LSTM(8)	0.6372	3.20970e-04

TABLE 3. CPU time (in second) comparison for the different ROM frameworks investigated in this study and RMSE at $t = 0.25$.

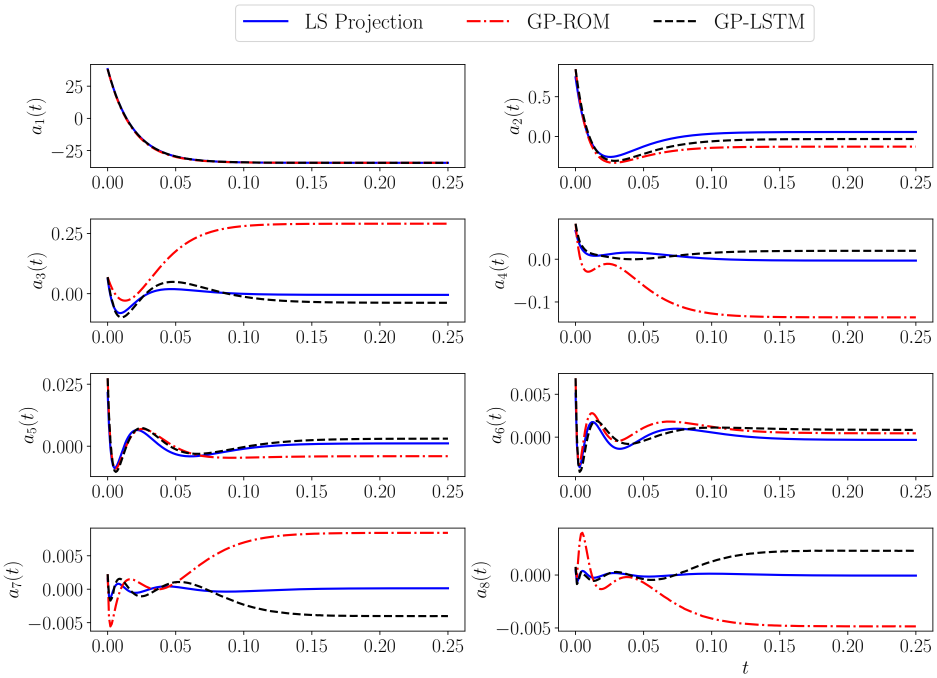

FIGURE 10. Temporal evolution of the 8 POD modal coefficients predicted by LS Projection, GP-ROM and GP-LSTM for $\gamma = 50$ using the basis from $\gamma = 40$.

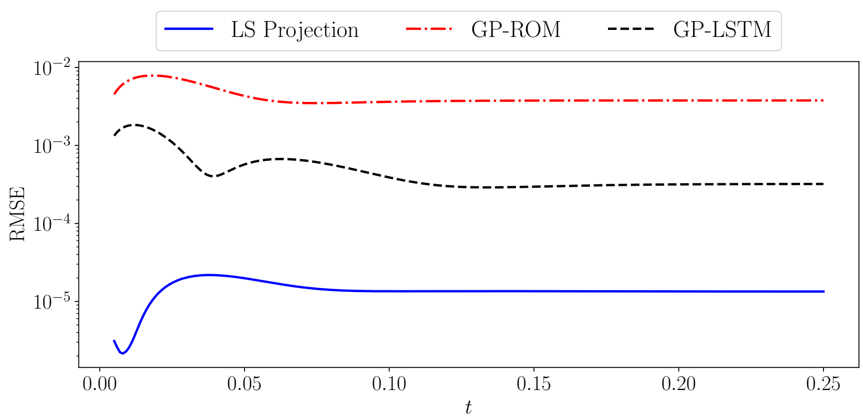

FIGURE 11. RMSE of ROMs predicted by LS Projection, GP-ROM and GP-LSTM for $\gamma = 50$ using the basis of $\gamma = 40$.

6. **Concluding remarks.** In this study, a simple feedback rule is developed in order to reduce the amount of computation. It has been shown that our feedback law works very well. It is believed that mathematical proof will be possible without much difficulty. For real-time computation, a numerical experiment was performed by adopting a GP-ROM. In order to increase the GP-ROM's accuracy, the deep

learning method, especially the LSTM method, which has been actively developed recently, was studied and applied. The ROM using deep learning such as LSTM performed in this study is considered to be worth continuing research. However, it is difficult to study systematically because the mathematical theory is not supported. We intend to apply GP-LSTM and GP-ResNet methods to the next studies, the optimal control problems and the uncertainty quantification problems of fluid flows.

Acknowledgments. We thank the reviewer for the comments that helped improve the paper.

REFERENCES

[1] F. Abergal and R. Temam, On some control problems in fluid mechanics, *Theoret. Comput. Fluid Dynamics*, **1** (1990), 303–325.

[2] R. A. Adams, *Sobolev Spaces*, Academic Press, 1975.

[3] S. E. Ahmed, O. San, A. Rasheed and T. Iliescu, A long short-term memory embedding for hybrid uplifted reduced order models, *Phys. D*, **409** (2020), 132471, 16 pp.

[4] D. Amsallem, *Interpolation on Manifolds of CFD-Based Fluid and Finite Element-Based Structural Reduced-Order Models for On-Line Aeroelastic Predictions*, Ph. D. Thesis, Stanford University, 2010.

[5] P. Benner, S. Gugercin and K. Willcox, A survey of projection-based model reduction methods for parametric dynamical systems, *SIAM Rev.*, **57** (2015), 483–531.

[6] G. Berkooz and P. Holmes and J. L. Lumley, The proper orthogonal decomposition in the analysis of turbulent flows, *Annual review of fluid mechanics*, **25** (1993), 539–575.

[7] S. Brunton and B. Noack and P. Koumoutsakos, Machine Learning for Fluid Mechanics, *Annu. Rev. Fluid Mech.*, **52** (2020), 477–508.

[8] J. Burkardt and M. Gunzburger and H.-C. Lee, Centroidal voronoi tessellation-based reduced-order modeling of complex systems, *SIAM J. Sci. Comput.*, **28** (2006), 459–484.

[9] J. Burkardt and M. Gunzburger and H.-C. Lee, POD and CVT-based reduced-order modeling of Navier-Stokes flows, *Comput. Methods Appl. Mech. Engrg.*, **196** (2006), 337–355.

[10] J. H. Faghmous, A. Banerjeeand, S. Shekharand, M. Steinbach, V. Kumar, A. R. Ganguly and N. Samatova, Theory-guided data science for climate change, *Computer*, **47** (2014), 74–78.

[11] J. Ffowcs-Williams and B. Zhao, Active control of vortex shedding, *J. Fluids Struct.*, **S3** (1989), 115–122.

[12] V. Girault and P. Raviart, *Navier-Stokes Equations*, North-Hollan, Amsterdam, 1979.

[13] V. Girault and P.-A. Raviart, *Finite Element Methods for Navier-Stokes Equations*, Springer-Verlag, Berlin, 1986.

[14] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University, Baltimore, 1996.

[15] M. D. Gunzburger and L. S. Hou, Finite-dimensional approximation of a class of constrained nonlinear optimal control problems, *SIAM J. Control Optim.*, **34** (1996), 1001–1043.

[16] M. Gunzburger and H.-C. Lee, Active control of vortex shedding, *J. Appl. Mech.*, **63** (1996), 828–835.

[17] M. D. Gunzburger and S. Manservisi, Analysis and approximation of the velocity tracking problem for Navier-Stokes flows with distributed control, *SIAM J. Numer. Anal.*, **37** (2000), 1481–1512.

[18] M. D. Gunzburger and S. Manservisi, The velocity tracking problem for Navier-Stokes flows with boundary control, *SIAM J. Control Optim.*, **39** (2000), 594–634.

[19] M. D. Gunzburger and S. Manservisi, Analysis and approximation for linear feedback control for tracking the velocity in Navier-Stokes flows, *Comput. Methods Appl. Mech. Engrg.*, **189** (2000), 803–823.

[20] F. Hecht, New development in FreeFem++, *J. Numer. Math.*, **20** (2012), 251–265.

[21] L. S. Hou and Y. Yan, Dynamics for controlled Navier-Stokes systems with distributed controls, *SIAM J. Control Optim.*, **35** (1997), 654–677.

[22] L. S. Hou and Y. Yan, Dynamics and approximations of a velocity tracking problem for the Navier-Stokes flows with piecewise distributed controls, *SIAM J. Control Optim.*, **35** (1997), 1847–1885.

[23] A. Karpatne, G. Atluri, J. H. Faghmous, M. Steinbach, A. Banerjee, A. Ganguly, S. Shekhar, N. Samatova and V. Kumar, Theory-guided data science: A new paradigm for scientific discovery from data, *IEEE Trans. Knowl. Data Eng.*, **29** (2017), 2318–2331.

[24] A. J. Majda and D. Qi, Strategies for reduced-order models for predicting the statistical responses and uncertainty quantification in complex turbulent dynamical systems, *SIAM Rev.*, **60** (2018), 491–549.

[25] S. Pawar, S. Ahmed, O. San and A. Rasheed, An evolve-then-correct reduced order model for hidden fluid dynamics. Mathematics, *Mathematics*, **8** (2020), 570.

[26] S. Pawar, S. E. Ahmed, O. San and A. Rasheed, Data-driven recovery of hidden physics in reduced order modeling of fluid flows, preprint, `arXiv:1910.13909`

[27] M. Rahman, S. Pawar, O. San, A. Rasheed and T. Iliescu, A non-intrusive reduced order modeling framework for quasi-geostrophic turbulence, preprint, `arXiv:1906.11617`

[28] M. Rathinam and L. R. Petzold, A new look at proper orthogonal decomposition, *SIAM J. Numer. Anal.*, **41** (2003), 1893–1925.

[29] L. Scarpa, Analysis and optimal velocity control of a stochastic convective Cahn-Hilliard equation, preprint, `arXiv:2007.14735`

[30] L. Sirovich, Turbulence and the dynamics of coherent structures, part i: Coherent structures; part ii: symmetries and transformations; part iii: Dynamics and scaling, *Quart. Appl. Math.*, **45** (1987), 561–590.

[31] M. Strazzullo, Z. Zainib, F. Ballarin and G. Rozza, Reduced order methods for parametrized non-linear and time dependent optimal flow control problems, towards applications in biomedical and environmental sciences, preprint, `arXiv:1912.07886`

[32] Kailai Xu, Bella Shi and Shuyi Yin, *Deep Learning for Partial Differential Equations*, Stanford University, 2018.

Received August 2020; revised September 2020.

E-mail address: `hclee@ajou.ac.kr`