

Real-Time Computing for a Parameterized Feedback Control Problem of Boussinesq Equations by POD and Deep Learning

Guang-Ri Piao¹ and Hyung-Chun Lee^{2,*}

¹Department of Mathematics, Yanbian University, Yanji 133002, China.

²Department of Mathematics, Ajou University, Suwon 16499, Korea.

Received 24 March 2023; Accepted (in revised version) 25 June 2023.

Abstract. An efficient real-time computational method for a feedback control problem of the Boussinesq equations is studied. We consider a simple and effective feedback control law based on the relationship between the control and adjoint variables in the optimality system. We investigate a closure type modeling in reduced order model (ROM) of this problem for real-time computing. In order to improve the existing well-known POD-ROM method, the deep learning technique, which is currently being actively researched, is studied and applied. Computational results presented show that the suggested methods work well.

AMS subject classifications: 65M10, 78A480

Key words: Optimal control, feedback control, Boussinesq, finite element, POD, LSTM.

1. Introduction

Many mathematicians and scientists have long worked on the mathematical analysis and computations of optimal control problems for fluid flows. Also, feedback control problem has been studied for efficient real-time computations. In this article, we study efficient computations for a linear feedback control problem of the Boussinesq equations describing viscous incompressible fluid flow coupled with thermodynamics. Dynamics and approximations for linear feedback controls for tracking velocities in Navier-Stokes in [14] and Bénard flows in [20, 21] were considered. Their goals were to steer over time a candidate velocity field \mathbf{u} and fluid temperature θ to a target velocity field $\mathbf{U} \in \mathbf{L}^2(\Omega)$ and fluid temperature $\Theta \in L^2(\Omega)$ by appropriately controlling the body forces of the velocity and temperature field. For real-time and efficient numerical computations, we study the reduced order modeling technique that has been researched for the past 30 years and the deep learning method that has been very actively researched recently.

*Corresponding author. Email addresses: grpiao@ybu.edu.cn (G.-R. Piao), hclee@ajou.ac.kr (H.-C. Lee)

First, we consider an optimal control problem for the non-dimensionalized Boussinesq equations, given $T > 0$, the target velocity \mathbf{U} , and the target temperature Θ , seek $(\mathbf{u}, \mathbf{f}, \theta, \tau)$ such that the cost functional

$$\begin{aligned} \mathcal{J}(\mathbf{u}, \mathbf{f}) = & \frac{1}{2} \int_0^T \int_{\Omega} (|\mathbf{u} - \mathbf{U}|^2 + |\theta - \Theta|^2) d\Omega dt + \int_0^T \int_{\Omega} \left(\frac{\alpha_1}{2} |\mathbf{f}|^2 + \frac{\alpha_2}{2} |\tau|^2 \right) d\Omega dt \\ & + \frac{\delta_1}{2} \int_{\Omega} |\mathbf{u}(T) - \mathbf{U}(T)|^2 d\Omega + \frac{\delta_2}{2} \int_{\Omega} |\theta(T) - \Theta(T)|^2 d\Omega \end{aligned} \quad (1.1)$$

is minimized subject to constraints

$$\begin{aligned} \mathbf{u}_t - \nu \Delta \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla p - \beta \theta \mathbf{g} &= \mathbf{f} && \text{in } (0, T) \times \Omega, \\ \nabla \cdot \mathbf{u} &= 0 && \text{in } (0, T) \times \Omega, \\ \theta_t - \kappa \Delta \theta + (\mathbf{u} \cdot \nabla) \theta &= \tau && \text{in } (0, T) \times \Omega, \\ \mathbf{u}|_{\partial\Omega} &= \mathbf{0}, \quad \mathbf{u}(0, \mathbf{x}) = \mathbf{u}_0(\mathbf{x}), \quad \theta(t, \mathbf{x})|_{\partial\Omega} = 0, \quad \theta(0, \mathbf{x}) = \theta_0(\mathbf{x}), \end{aligned} \quad (1.2)$$

where Ω is a bounded open set in \mathbb{R}^2 denoted by $\partial\Omega$. Here \mathbf{u} is the velocity vector, p is the pressure, θ is the temperature of the fluid, \mathbf{f} is a source field, τ is a heat source. The functions \mathbf{u}_0 and θ_0 are given, \mathbf{g} is a unit vector in the direction of gravitational acceleration, $\beta > 0$ is any positive number, $\nu > 0$ is the kinematic viscosity and $\kappa > 0$ is the thermal conductivity parameter. Using the Lagrange multipliers method, one can obtain the following optimality system: seek $(\mathbf{u}, p, \theta, \mathbf{w}, r, \psi)$ such that

$$\begin{aligned} \mathbf{u}_t - \nu \Delta \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla p - \beta \theta \mathbf{g} &= \mathbf{f} && \text{in } (0, T) \times \Omega, \\ \nabla \cdot \mathbf{u} &= 0 && \text{in } (0, T) \times \Omega, \\ \theta_t - \kappa \Delta \theta + (\mathbf{u} \cdot \nabla) \theta &= \tau && \text{in } (0, T) \times \Omega, \end{aligned} \quad (1.3a)$$

$$\begin{aligned} -\mathbf{w}_t, \mathbf{v} + \nu \Delta \mathbf{w} + (\mathbf{w} \cdot \nabla) \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{w} + \nabla r - \beta \psi \mathbf{g} &= \mathbf{u} - \mathbf{U} && \text{in } (0, T) \times \Omega, \\ \nabla \cdot \mathbf{w} &= 0 && \text{in } (0, T) \times \Omega, \\ -\psi_t + \kappa \Delta \psi + (\mathbf{w} \cdot \nabla) \psi &= \theta - \Theta && \text{in } (0, T) \times \Omega, \end{aligned} \quad (1.3b)$$

$$\mathbf{w} = -\alpha_1 \mathbf{f}, \quad \psi = -\alpha_2 \tau \quad (1.3c)$$

with the homogeneous boundary conditions, initial velocity and temperature (\mathbf{u}_0, θ_0) for the state equations, and the final conditions

$$\mathbf{w}(T, \mathbf{x}) = \delta_1 (\mathbf{u}(T) - \mathbf{U}(T)), \quad \psi(T, \mathbf{x}) = \delta_2 (\theta(T) - \Theta(T))$$

for the adjoint equations. The optimal system (1.3) is a system of nonlinear partial differential equations consisting of nonlinear state equations (1.3a), linear adjoint equations (1.3b), and an optimality condition (1.3c). The state equations are forward in time and the adjoint equations are backward in time. For these reasons, it is known that the numerical computation of the optimality system is almost impossible or the amount of computation is

prohibitive. In order to overcome the enormous amount of computation of the optimal system, some piecewise optimal control methods were considered [22] and references therein. Although the amount of computations was reduced through piecewise optimization control, a huge amount of computations was still required.

From the Eqs. (1.3c), we see that controls \mathbf{f} and τ are related linearly with the adjoint variables \mathbf{w} and ψ , respectively, which are solutions of the adjoint Eqs. (1.3b) with $\mathbf{f} = -(1/\alpha_1)\mathbf{w}$ and $\tau = -(1/\alpha_2)\psi$. Also, we see that (\mathbf{w}, ψ) is linearly dependent only on the source terms of the adjoint Eqs. (1.3b) $(\mathbf{u} - \mathbf{U}, \theta - \Theta)$ since the adjoint equations are linear equations. So, we see that

$$\mathbf{f} \propto \frac{1}{\alpha_1}(\mathbf{u} - \mathbf{U}) \quad \text{and} \quad \tau \propto \frac{1}{\alpha_2}(\theta - \Theta).$$

Usually, α_1 and α_2 are very small real numbers to get a small value of the first term in the right-hand side of (1.1). Thus, we may let

$$\mathbf{f} = \mathbf{F} - \gamma_1(\mathbf{u} - \mathbf{U}) \quad \text{and} \quad \tau = \mathcal{T} - \gamma_2(\theta - \Theta)$$

with some large values of γ_1 and γ_2 . The smaller α_1 and α_2 , the greater control \mathbf{f} and τ . Increasing controls is not a problem in numerical simulation, but it is expensive in actual control problems. Therefore, γ_1 and γ_2 can be set according to the actual problem. The controls are achieved by means of a linear feedback law relating the body forces to the velocity and temperature fields. $(\mathbf{F}, \mathcal{T})$ will be defined in (2.1). Thus, a linear feedback control problem of Boussinesq equations can be formulated by

$$\begin{aligned} \mathbf{u}_t - \nu \Delta \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla p - \beta \theta \mathbf{g} &= \mathbf{F} - \gamma_1(\mathbf{u} - \mathbf{U}) && \text{in } (0, T) \times \Omega, \\ \nabla \cdot \mathbf{u} &= 0 && \text{in } (0, T) \times \Omega, \\ \theta_t - \kappa \Delta \theta + (\mathbf{u} \cdot \nabla) \theta &= \mathcal{T} - \gamma_2(\theta - \Theta) && \text{in } (0, T) \times \Omega. \end{aligned} \quad (1.4)$$

The feedback control problem (1.4) is linear control, which can be explicitly solved. This would actually improve the efficiency of the proposed method. In addition, we point out that the optimal control would not result major computational cost could avoid more detailed discussions on the classic optimal control solvers. Feedback control can effect good velocity and temperature tracking and at the same time the solution can be obtained step-by-step in time at the cost of a single flow problem solve. Some efficient numerical algorithm for solving data driven feedback control problems was considered in [4] and references therein.

Proper orthogonal decomposition (POD) is a common technique for extracting the dominant mode that contributes the most to the energy of the entire system [7, 28]. POD combined with Galerkin projection (GP) has been used for many years to formulate ROMs for dynamic systems [5, 9, 10, 23, 29, 30]. 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).

Machine learning (ML) tools have had considerable success in the fluid mechanics community, identifying basic structures and mimicking dynamics [2, 8, 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 [12, 18]. In the course of conducting this study, we also experienced these points seriously. 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 article [8] and references therein. Recently, in [19], using the POD and LSTM, the efficient computational method of the linear feedback control problem of the Navier-Stokes equations has been studied.

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

2. Notations and Formulation of Feedback Control Problem

2.1. Notations and weak forms

We shall use the standard function spaces and their norms, for details see [1]. For any nonnegative integer m , we define the Sobolev space $H^m(\Omega)$ by

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

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

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

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 spaces 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

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

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

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

We also define a particular subspace and solenoidal spaces — viz.

$$\begin{aligned} L_0^2(\Omega) &:= \left\{ p \in L^2(\Omega) : \int_{\Omega} p \, dx = 0 \right\}, & \mathcal{V}(\Omega) &:= \{\mathbf{u} \in \mathbf{C}_0^\infty(\Omega) : \nabla \cdot \mathbf{u} = 0\}, \\ \mathbf{V}(\Omega) &:= \{\mathbf{u} \in \mathbf{H}_0^1(\Omega) : \nabla \cdot \mathbf{u} = 0\}, & \mathbf{W}(\Omega) &:= \{\mathbf{u} \in \mathbf{L}^2(\Omega) : \nabla \cdot \mathbf{u} = 0\}. \end{aligned}$$

The spaces $\mathbf{V}(\Omega)$ and $\mathbf{W}(\Omega)$ are closures of $\mathcal{V}(\Omega)$ 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. For the spaces $L^2(\Omega)$ and $\mathbf{L}^2(\Omega)$, we denote the inner product by (\cdot, \cdot) and the norm by $\|\cdot\|_0 = \|\cdot\|$. Given T , we introduce the notation $L^p((0, T) : H^m(\Omega))$ for the temporal-spatial function spaces defined on $(0, T) \times \Omega$ with the norm

$$\|u\|_{L^p((0, T); X)} = \left(\int_0^T \|u\|^p dt \right)^{\frac{1}{p}}.$$

In order to define the weak form of the Boussinesq equations, for all $\mathbf{u}, \mathbf{v}, \mathbf{w} \in \mathbf{H}^1(\Omega)$, $\theta, s \in H^1(\Omega)$, and $q \in L^2(\Omega)$, we introduce bilinear and trilinear forms by

$$\begin{aligned} a_0(\mathbf{u}, \mathbf{v}) &= \int_{\Omega} \nabla \mathbf{u} : \nabla \mathbf{v} \, dx, & \kappa a_1(\theta, s) &= \int_{\Omega} \nabla \theta \cdot \nabla s \, dx, \\ b_0(\mathbf{u}, \mathbf{v}, \mathbf{w}) &= \int_{\Omega} (\mathbf{u} \cdot \nabla) \mathbf{v} \cdot \mathbf{w} \, dx, & b_1(\mathbf{u}, \theta, s) &= \int_{\Omega} (\mathbf{u} \cdot \nabla \theta) s \, dx \quad \text{for all } \mathbf{u} \in \mathbf{H}^1(\Omega), \\ c(\mathbf{v}, q) &= - \int_{\Omega} q \nabla \cdot \mathbf{v} \, dx, & d(\theta, \mathbf{v}) &= \int_{\Omega} \theta \mathbf{g} \cdot \mathbf{v} \, dx. \end{aligned}$$

We start with a precise definition of the admissible target velocity and temperature. (\mathbf{U}, Θ) is said to be in the set of admissible target velocities and temperatures U_{ad} if \mathbf{U} is a divergence free vector field and Θ is a temperature field in the set $\{(\mathbf{u}, \theta) : \mathbf{u} \in C((0, T) : \mathbf{H}^2(\Omega) \cap \mathbf{H}_0(\Omega)), \theta \in C((0, T) : H^2(\Omega) \cap H_0^1(\Omega)) \mid \partial_t \mathbf{u} \in C((0, T) : \mathbf{H}^1(\Omega)), \partial_t \theta \in C((0, T) : H^1(\Omega))\}$. The corresponding body forces $(\mathbf{F}, \mathcal{F})$ are generated by

$$\begin{aligned} \mathbf{F}(t, \mathbf{x}) &= \frac{\partial \mathbf{U}(t, \mathbf{x})}{\partial t} - \nu \Delta \mathbf{U}(t, \mathbf{x}) + (\mathbf{U}(t, \mathbf{x}) \cdot \nabla) \mathbf{U}(t, \mathbf{x}) - \beta \Theta(t, \mathbf{x}) \mathbf{g}, \\ \mathcal{F}(t, \mathbf{x}) &= \frac{\partial \Theta(t, \mathbf{x})}{\partial t} - \kappa \Delta \Theta(t, \mathbf{x}) + (\mathbf{U}(t, \mathbf{x}) \cdot \nabla) \Theta(t, \mathbf{x}). \end{aligned} \tag{2.1}$$

Let $\mathbf{u} \in L^2((0, T) : \mathbf{H}_0^1(\Omega))$, $p \in L^2((0, T) : L_0^2(\Omega))$ and $\theta \in L^2((0, T) : H_0^1(\Omega))$ denote the state variables, i.e., the velocity, pressure and temperature fields, respectively. Let $\mathbf{f} \in L^2((0, T) : \mathbf{L}^2(\Omega))$ and $\tau \in L^2((0, T) : L^2(\Omega))$ denote the distributed controls. The state variables are constrained to satisfy the weak form of the Boussinesq equations (1.2) a.e. for t in $(0, T)$, i.e.

$$\begin{aligned} \left(\frac{\partial \mathbf{u}}{\partial t}, \mathbf{v} \right) + \nu a_0(\mathbf{u}, \mathbf{v}) + b_0(\mathbf{u}, \mathbf{u}, \mathbf{v}) + c(\mathbf{v}, p) - \beta d(\theta, \mathbf{v}) &= (\mathbf{f}, \mathbf{v}) & \text{for all } \mathbf{v} \in \mathbf{H}_0^1(\Omega), \\ c(\mathbf{u}, q) &= 0 & \text{for all } q \in L_0^2(\Omega), \\ \left(\frac{\partial \theta}{\partial t}, s \right) + \kappa a_1(\theta, s) + b_1(\mathbf{u}, \theta, s) &= (\tau, s) & \text{for all } s \in H_0^1(\Omega) \end{aligned}$$

with the homogeneous boundary conditions and initial velocity field $\mathbf{u}_0(\mathbf{x})$ and initial temperature field $\theta_0(\mathbf{x})$. (\cdot, \cdot) denotes the L^2 inner product of the functions over the domain Ω .

2.2. Feedback control

From the Eqs. (1.4), a linear feedback control problem of Boussinesq equations can be written in a weak form by

$$\begin{aligned} (\mathbf{u}_t, \mathbf{v}) + \nu a_0(\mathbf{u}, \mathbf{v}) + b_0(\mathbf{u}, \mathbf{u}, \mathbf{v}) + c(\mathbf{v}, p) - \beta d(\theta, \mathbf{v}) &= (\mathbf{F} - \gamma_1(\mathbf{u} - \mathbf{U}), \mathbf{v}), \\ c(\mathbf{u}, q) &= 0, \\ (\theta_t, s) + \kappa a_1(\theta, s) + b_1(\mathbf{u}, \theta, s) &= (\mathcal{T} - \gamma_2(\theta - \Theta), s), \end{aligned} \quad (2.2)$$

$\forall \mathbf{v} \in \mathbf{H}_0^1(\Omega)$, $\forall q \in L_0^2(\Omega)$, $\forall s \in H_0^1(\Omega)$ with initial conditions $\mathbf{u}_0(x) \in \mathbf{V}(\Omega)$ and $\theta_0(x) \in H_0^1(\Omega)$ and homogeneous boundary conditions. An admissible solution for our control problem can be defined as follows. Given T , $\mathbf{u}_0(x) \in \mathbf{V}(\Omega)$, $\theta_0(x) \in H_0^1(\Omega)$, and $(\mathbf{U}, \Theta) \in U_{ad}$, the solution (\mathbf{u}, p, θ) of (2.2) is called an admissible solution for the control problem if $(\mathbf{u}, p, \theta) \in L^2((0, T); \mathbf{H}_0^1(\Omega)) \times L^2((0, T); L_0^2(\Omega)) \times L^2((0, T); H_0^1(\Omega))$ and

$$\frac{d}{dt} (\|\mathbf{u} - \mathbf{U}\|^2 + \|\theta - \Theta\|^2) \leq 0, \quad \text{a.e. } t \in (0, T).$$

The set of all admissible solutions is notated as A_{ad} .

The goal of our feedback control problem is to have the controlled solution (\mathbf{u}, θ) match (\mathbf{U}, Θ) over time — i.e. we want the solution (\mathbf{u}, θ) to belong to the set of admissible solutions and $\|\mathbf{u} - \mathbf{U}\|^2 + \|\theta - \Theta\|^2 \rightarrow 0$ as t increases. The next theorem shows that this decay property holds and furthermore that this decay is exponential.

Theorem 2.1 (cf. Lee & Choi [20]). *If $(\mathbf{u}, p, \theta, \mathbf{f}, \tau)$ is a solution of the linear feedback control problem (2.2) with $\gamma_1 \geq M$ and $\gamma_2 \geq N$, then*

(i) *The solution $(\mathbf{u}, p, \theta, \mathbf{f}, \tau)$ belongs to A_{ad} , i.e.*

$$\frac{d}{dt} (\|\mathbf{u} - \mathbf{U}\|^2 + \|\theta - \Theta\|^2) \leq 0, \quad \text{a.e. } t \in (0, T).$$

(ii) *The following estimate:*

$$\|\mathbf{u} - \mathbf{U}\|^2 + \|\theta - \Theta\|^2 \leq (\|\mathbf{u}_0 - \mathbf{U}_0\|^2 + \|\theta_0 - \Theta_0\|^2) e^{-2\tilde{\gamma}t},$$

where

$$\tilde{\gamma} = \min\{\gamma_1 - M, \gamma_2 - N\}, \quad \gamma_1 - M \geq 0, \quad \gamma_2 - N \geq 0,$$

and

$$\begin{aligned} M &= \max \left\{ 0, \frac{1}{\nu} \|\nabla \mathbf{U}\|_{L^\infty((0, T); L^2(\Omega))}^2 + \frac{\beta}{2} + \frac{\nu^2}{8\kappa} - \frac{\nu C_0}{4} \right\}, \\ N &= \max \left\{ 0, \frac{1}{\nu^3} \|\nabla \Theta\|_{L^\infty((0, T); L^2(\Omega))}^4 + \frac{\beta}{2} - \frac{\kappa C_1}{2} \right\} \end{aligned}$$

holds a.e. in $(0, T)$.

Fig. 4 in Section 5 shows that the controlled states (\mathbf{u}, θ) in numerical simulations converge exponentially to desired states (\mathbf{U}, Θ) . Here we set $\gamma = \gamma_1 = \gamma_2$ without loss of generality.

3. POD-Galerkin Reduced Order Model (GP-ROM)

3.1. Finite element approximation

A typical finite element approximation [13] of (2.2) is defined as follows: we first choose conforming finite element subspaces $\mathbb{V}_h \subset \mathbf{H}^1(\Omega)$, $S_h \subset L^2(\Omega)$ and $X_h \subset H^1(\Omega)$ and then define $\mathbb{V}_{h,0} = \mathbf{H}_0^1(\Omega)$, $S_{h,0} = L_0^2(\Omega)$ and $X_{h,0} = H_0^1(\Omega)$. One then seeks $\mathbf{u}_h(t, \cdot) \in \mathbb{V}_{h,0}$, $p_h \in S_{h,0}$ and $\theta_h(t, \cdot) \in X_{h,0}$ such that

$$\begin{aligned}
& \left(\frac{\partial \mathbf{u}_h}{\partial t}, \mathbf{v}_h \right) + \nu a_0(\mathbf{u}_h, \mathbf{v}_h) + b_0(\mathbf{u}_h; \mathbf{u}_h, \mathbf{v}_h) + c(\mathbf{v}_h, p_h) \\
& \quad - \beta d(\theta_h, \mathbf{v}_h) + \gamma_1(\mathbf{u}_h, \mathbf{v}_h) \\
& = (\mathbf{F} + \gamma_1 \mathbf{U}, \mathbf{v}_h) \quad \text{for all } \mathbf{v}_h \in \mathbb{V}_{h,0}, \\
& \quad c(\mathbf{u}_h, q_h) = 0 \quad \text{for all } q_h \in S_h, \\
& \left(\frac{\partial \theta_h}{\partial t}, \psi_h \right) + \kappa a_1(\theta_h, \psi_h) + b_1(\mathbf{u}_h, \theta_h, \psi_h) + \gamma_2(\theta_h, \psi_h) \\
& = (\mathcal{F} + \gamma_2 \Theta, \psi_h) \quad \text{for all } \psi_h \in X_{h,0}, \\
& \quad \mathbf{u}_h(0, \mathbf{x}) = \mathbf{u}_{h,0}(\mathbf{x}), \quad \theta_h(0, \mathbf{x}) = \theta_{h,0}(\mathbf{x}),
\end{aligned} \tag{3.1}$$

where $\mathbf{u}_{0,h}(\mathbf{x}) \in \mathbb{V}_{h,0}$ and $\theta_{h,0}(\mathbf{x}) \in X_h$ are approximations — viz. the projections of $\mathbf{u}_0(\mathbf{x})$ and $\theta_0(\mathbf{x})$ onto each finite element space, respectively. Discretization is completed by choosing a time-marching method such as the backward Euler scheme.

Let $\{t_n\}_{n=1}^N$ be a partition of $[0, T]$ into equal intervals $\Delta t = T/N$ and $t_n = t_0 + n\Delta t$ with $t_0 = 0$ and $t_N = T$. Then, a fully discretized version of (3.1) is given by

$$\begin{aligned}
& \frac{1}{\Delta t} \left(\mathbf{u}_h^{(n)} - \mathbf{u}_h^{(n-1)}, \mathbf{v}_h \right) + \nu a_0 \left(\mathbf{u}_h^{(n)}, \mathbf{v}_h \right) + b_0 \left(\mathbf{u}_h^{(n)}; \mathbf{u}_h^{(n)}, \mathbf{v}_h \right) + c \left(\mathbf{v}_h, p_h^{(n)} \right) \\
& \quad - \beta d \left(\theta_h^{(n)}, \mathbf{v}_h \right) + \gamma_1 \left(\mathbf{u}_h^{(n)}, \mathbf{v}_h \right) \\
& = (\mathbf{F}^{(n)} + \gamma_1 \mathbf{U}^{(n)}, \mathbf{v}_h) \quad \text{for all } \mathbf{v}_h \in \mathbb{V}_{h,0}, \\
& \quad c \left(\mathbf{u}_h^{(n)}, q_h \right) = 0 \quad \text{for all } q_h \in S_h, \\
& \frac{1}{\Delta t} \left(\theta_h^{(n)} - \theta_h^{(n-1)}, \psi_h \right) + \kappa a_1 \left(\theta_h^{(n)}, \psi_h \right) + b_1 \left(\mathbf{u}_h^{(n)}, \theta_h^{(n)}, \psi_h \right) + \gamma_2 \left(\theta_h^{(n)}, \psi_h \right) \\
& = (\mathcal{F}^{(n)} + \gamma_2 \Theta^{(n)}, \psi_h) \quad \text{for all } \psi_h \in X_{h,0}
\end{aligned}$$

for $N = 1, 2, \dots, N$, initial velocity $\mathbf{u}_h^{(0)}(\mathbf{x}) = \pi^h \mathbf{u}_0(\mathbf{x})$ and temperature $\theta_h^{(0)}(\mathbf{x}) = \pi^h \theta_0(\mathbf{x})$. Here, $\mathbf{U}^{(n)} = \mathbf{U}(t_n, \mathbf{x})$, $\Theta^{(n)} = \Theta(t_n, \mathbf{x})$ and

$$\begin{aligned}
(\mathbf{F}^{(n)}, \mathbf{v}_h) &= \frac{1}{\Delta t} \left(\mathbf{U}_h^{(n)} - \mathbf{U}_h^{(n-1)}, \mathbf{v}_h \right) + \nu a_0 \left(\mathbf{U}_h^{(n)}, \mathbf{v}_h \right) + b_0 \left(\mathbf{U}_h^{(n)}; \mathbf{U}_h^{(n)}, \mathbf{v}_h \right) - \beta d \left(\Theta_h^{(n)}, \mathbf{v}_h \right), \\
(\mathcal{G}_h^{(n)}, \psi_h) &= \frac{1}{\Delta t} \left(\Theta_h^{(n)} - \Theta_h^{(n-1)}, \psi_h \right) - \kappa a_1 \left(\Theta_h^{(n)}, \psi_h \right) + b_1 \left(\Theta_h^{(n)}, \Theta_h^{(n)}, \psi_h \right).
\end{aligned}$$

3.2. GP-ROM

We now apply a POD basis to define a reduced-order model for our feedback system. Let $\{\varphi(\gamma_1)\}_{k=1}^{K_u}$ and $\{\xi(\gamma_2)\}_{k=1}^{K_\theta}$ be a K_u - and K_θ - dimensional POD basis corresponding to the snapshot set $\{\mathbf{u}_n(\gamma_1)\}_{n=1}^N$ and $\{\theta_n(\gamma_2)\}_{n=1}^N$, respectively, where $K_u \ll N$ and $K_\theta \ll N$. Let

$$\mathbb{U}_u(\gamma_1) = \text{span}\{\varphi_k(\gamma_1)\}_{k=1}^{K_u} \subset \mathbb{V}_{h,0}, \quad \mathbb{X}_\theta(\gamma_2) = \text{span}\{\xi_k(\gamma_2)\}_{k=1}^{K_\theta} \subset X_{h,0}.$$

From now on, we will omit γ_1 and γ_2 on the assumption that there is no confusion. We will use γ when we want to emphasize the use of γ . For a given reduced spaces \mathbb{U}_u and \mathbb{X}_θ , GP-ROM finds the approximation of the velocity and temperature fields spanned by the low-dimensional spaces,

$$\mathbf{u} \approx \mathbf{u}_h^{POD}(t, \mathbf{x}) \equiv \sum_{k=1}^{K_u} a_k(t) \varphi_k(\mathbf{x}), \quad \theta \approx \theta_h^K(t, \mathbf{x}) \equiv \sum_{k=1}^{K_\theta} b_k(t) \xi_k(\mathbf{x}),$$

where $\{a_k(t)\}_{k=1}^{K_u}$ and $\{b_k(t)\}_{k=1}^{K_\theta}$ 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

$$\begin{aligned} & \sum_{k=1}^{K_u} \frac{d}{dt} a_k(t) (\varphi_k, \varphi_\ell) + \nu \sum_{k=1}^{K_u} a_k(t) \int_{\Omega} \nabla \varphi_k : \nabla \varphi_\ell \, d\Omega - \beta \sum_{k=1}^{K_u} b_k(t) (\xi_k \mathbf{g}, \varphi_\ell) \\ & + \left(\sum_{m=1}^{K_u} a_m(t) \varphi_m \cdot \nabla \sum_{k=1}^{K_u} a_k(t) \varphi_k, \varphi_\ell \right) + \gamma_1 \sum_{k=1}^{K_u} a_k(t) (\varphi_k, \varphi_\ell) \\ = & \sum_{k=1}^{K_u} h_k(t) (\varphi_k, \varphi_\ell) + \gamma_1 \sum_{k=1}^{K_u} c_k(t) (\varphi_k, \varphi_\ell), \\ & \sum_{k=1}^{K_u} a_k(0) (\varphi_k, \varphi_\ell) = (\mathbf{u}_0, \varphi_\ell), \quad \sum_{k=1}^{K_u} c_k(t) (\varphi_k, \varphi_\ell) = (\mathbf{U}, \varphi_\ell), \\ & \sum_{k=1}^{K_\theta} \frac{d}{dt} b_k(t) (\xi_k, \xi_j) + \kappa \sum_{k=1}^{K_\theta} b_k(t) (\nabla \xi_k, \nabla \xi_j) + \left(\sum_{m=1}^{K_\theta} a_m(t) \varphi_m \cdot \nabla \sum_{k=1}^{K_\theta} b_k(t) \xi_k, \xi_j \right) \\ & + \gamma_2 \sum_{k=1}^{K_\theta} b_k(t) (\xi_k, \xi_j) \\ = & \sum_{k=1}^{K_\theta} o_k(t) (\xi_k, \xi_j) + \gamma_2 \sum_{k=1}^{K_\theta} d_k(t) (\xi_k, \xi_j), \\ & \sum_{k=1}^{K_\theta} b_k(0) (\xi_k, \xi_j) = (\theta_0, \xi_j), \quad \sum_{k=1}^{K_\theta} d_k(t) (\xi_k, \xi_j) = (\Theta, \xi_j) \end{aligned}$$

for $\ell = 1, \dots, K_u$ and $j = 1, \dots, K_\theta$.

Equivalently, we have a system of nonlinear ordinary differential equations, which determines the coefficient functions $\{a_k(t)\}_{k=1}^{K_u}$ and $\{b_k(t)\}_{k=1}^{K_\theta}$

$$\mathbb{G} \frac{d}{dt} \mathbf{a}(t) = -\nu \mathbb{K} \mathbf{a}(t) - (\mathbf{a}(t))^T \mathbb{N} \mathbf{a}(t) + \beta \mathbb{L} \mathbf{b}(t) - \gamma_1 \mathbb{G} (\mathbf{a}(t) - \mathbf{c}(t)) + \mathbb{G} \mathbf{h}(t), \quad (3.2)$$

$$\mathbb{W} \frac{d}{dt} \mathbf{b}(t) = -\kappa \mathbb{P} \mathbf{b}(t) - (\mathbf{a}(t))^T \mathbb{Q} \mathbf{b}(t) - \gamma_2 \mathbb{W} (\mathbf{b}(t) - \mathbf{d}(t)) + \mathbb{W} \mathbf{o}(t), \quad (3.3)$$

where the Gram matrices \mathbb{G} and \mathbb{W} , stiffness matrices \mathbb{K} and \mathbb{P} , convection tensors \mathbb{N} and \mathbb{P} , given vectors $\mathbf{c}(t)$, $\mathbf{d}(t)$, $\mathbf{h}(t)$ and $\mathbf{o}(t)$ and solution vectors $\mathbf{a}(t)$ and $\mathbf{b}(t)$ are respectively given by

$$\begin{aligned} \mathbb{G}_{k\ell} &= (\varphi_k, \varphi_\ell), & \mathbb{K}_{k\ell} &= \int_{\Omega} \nabla \varphi_k : \nabla \varphi_\ell d\Omega, & \mathbb{N}_{k\ell m} &= ((\varphi_k \cdot \nabla) \varphi_\ell, \varphi_m), \\ \mathbb{L}_{k\ell} &= (\xi_k \mathbf{g}, \varphi_\ell), & \mathbb{W}_{k\ell} &= (\xi_k, \xi_\ell), & \mathbb{P}_{k\ell} &= (\nabla \xi_k \cdot \nabla \xi_\ell), & \mathbb{Q}_{k\ell m} &= ((\varphi_k \cdot \nabla) \xi_\ell, \xi_m), \\ c_k(t) &= (\mathbf{U}(t), \varphi_k), & d_k(t) &= (\Theta(t), \xi_k), & h_k(t) &= (\mathbf{F}(t), \varphi_k), & o_k(t) &= (\mathcal{T}(t), \xi_k). \end{aligned}$$

The initial conditions for the ROM are have the form

$$a_k(0) = (\mathbf{u}_0, \varphi_k), \quad b_k(0) = (\theta_0, \xi_k).$$

POD basis functions satisfy the orthogonality conditions

$$\int_{\Omega} \varphi_i(\mathbf{x}) \varphi_j(\mathbf{x}) d\mathbf{x} = \delta_{ij}, \quad \int_{\Omega} \xi_i(\mathbf{x}) \xi_j(\mathbf{x}) d\mathbf{x} = \delta_{ij},$$

where δ_{ij} is 1 if $i = j$ and 0 otherwise. Let

$$q_k(t) = \begin{cases} a_k(t) & \text{for } k = 1, \dots, K_u, \\ b_{k-K_u}(t) & \text{for } k = K_u + 1, \dots, K_u + K_\theta. \end{cases}$$

Then, (3.2) can be written by

$$\dot{q}_i(t) = \sum_{j=1}^{K_u} \mathbb{D}_{ij} q_j(t) + \sum_{j=1}^{K_u} \sum_{k=1}^{K_u} \mathbb{C}_{ijk} q_k(t) q_j(t) - \gamma_1 (q_i(t) - c_i(t)) + h_i(t), \quad (3.4)$$

where

$$\mathbb{D}_{ij} = \begin{cases} -\nu \mathbb{K}_{ij} & \text{for } j = 1, \dots, K_u, \\ -\beta \mathbb{L}_{ij} & \text{for } j = K_u + 1, \dots, K_u + K_\theta, \end{cases}$$

and

$$\mathbb{C}_{ijk} = \mathbb{N}_{jki}$$

for $i, k = 1, \dots, K_u$. Also, (3.3) can be written by

$$\dot{q}_i(t) = \sum_{j=1}^{K_\theta} \mathbb{D}_{ij} q_j(t) + \sum_{j=K_u+1}^{K_u+K_\theta} \sum_{k=1}^{K_u} \mathbb{C}_{ijk} q_k(t) q_j(t) - \gamma_2 (q_i(t) - d_i(t)) + o_i(t), \quad (3.5)$$

where

$$\mathbb{D}_{ij} = -\kappa \mathbb{P}_{ij} \quad \text{for } j = K_u + 1, \dots, K_u + K_\theta$$

for $j = K_u + 1, \dots, K_u + K_\theta$, and

$$\mathbb{C}_{ijk} = \mathbb{Q}_{kji}$$

for $j = K_u + 1, \dots, K_u + K_\theta$, $k = 1, \dots, K_u$, and $i = K_u + 1, \dots, K_u + K_\theta$. Here we use the shifted index for $\xi(\mathbf{x})$ like that

$$\xi_{K_u+j}(\mathbf{x}) \Leftarrow \xi_j(\mathbf{x}) \quad \text{for } j = 1, \dots, K_\theta.$$

Combining (3.4) and (3.5) leads to a dynamical system for the POD coefficients of the form

$$\dot{\mathbf{q}}(t) = \mathbb{D} \mathbf{q}(t) + [\mathbb{C} \mathbf{q}(t)] \mathbf{q}(t) \quad (3.6)$$

with initial conditions

$$q_i(0) = \begin{cases} a_i(0) & \text{for } i = 1, \dots, K_u, \\ b_{i-K_u}(0) & \text{for } i = K_u + 1, \dots, K_u + K_\theta. \end{cases}$$

4. Closure Type Model Using Least Squares Projection and LSTM

4.1. Closure model using least squares projection

Current ROMs cannot be used in complex, realistic settings, since they require too many modes (degrees of freedom) [24]. The drastic ROM truncation is one of the most important reasons for the ROMs' numerical inaccuracy. To overcome this, closure type modeling is studied using ideas from the previous articles [6, 11, 15, 17] and references therein. One needs to model the effect of the discarded ROM modes on the ROM dynamics — i.e. on the time evolution of the resolved ROM modes

$$\dot{\mathbf{q}}(t) = \mathbb{D} \mathbf{q}(t) + [\mathbb{C} \mathbf{q}(t)] \mathbf{q}(t) + \mathbf{C}(\mathbf{q}(t)),$$

where $\mathbf{C}(\mathbf{q}(t))$ is a low-dimensional term that models the effect of the discarded ROM modes on the active modes.

From now on, we let $K = K_u = K_\theta$ for simplicity. It is easy to see that the computational cost of GP-ROM (3.6) is $\mathcal{O}(K^3)$, 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.

Let $\mathbf{w} = (\mathbf{u}, \theta)$. Then the model reduction error $\mathcal{E}^{\text{ROM}}(t, \mathbf{x}, \gamma)$ is equal to

$$\begin{aligned} \mathcal{E}^{\text{ROM}}(t, \mathbf{x}, \gamma) &= \mathbf{w}^{\text{FOM}}(t, \mathbf{x}, \gamma) - \mathbf{w}^{\text{ROM}}(t, \mathbf{x}, \gamma) \\ &= \mathbf{w}^{\text{FOM}}(t, \mathbf{x}, \gamma) - \mathbf{w}^{\text{Proj}}(t, \mathbf{x}, \gamma) + \mathbf{w}^{\text{Proj}}(t, \mathbf{x}, \gamma) - \mathbf{w}^{\text{ROM}}(t, \mathbf{x}, \gamma) \\ &= \mathcal{E}_{\mathbf{w}(\gamma)^\perp}(t, \mathbf{x}) + \mathcal{E}_{\mathbf{w}(\gamma)}(t, \mathbf{x}). \end{aligned}$$

The first term $\mathcal{E}_{\mathbf{W}(\gamma)^\perp}$ is the least-squares projection error that is a result of neglecting the components of $\mathbf{w}^{\text{FOM}}(t, \mathbf{x}, \gamma)$ that lie in the space orthogonal to $\mathbf{W}(\gamma)$. This error is represented as $\mathcal{E}_{\mathbf{W}(\gamma)^\perp}$ in Fig. 1. The second term $\mathcal{E}_{\mathbf{W}(\gamma)}$ is the reduced order modeling error that results from solving a different dynamical system than the original. This error vector lies in the subspace $\mathbf{W}(\gamma)$ spanned by the reduced order basis $\{(\varphi_i, \xi_i)\}_{i=1}^K$. This error represents $\mathcal{E}_{\mathbf{W}(\gamma)}$ in Fig. 1.

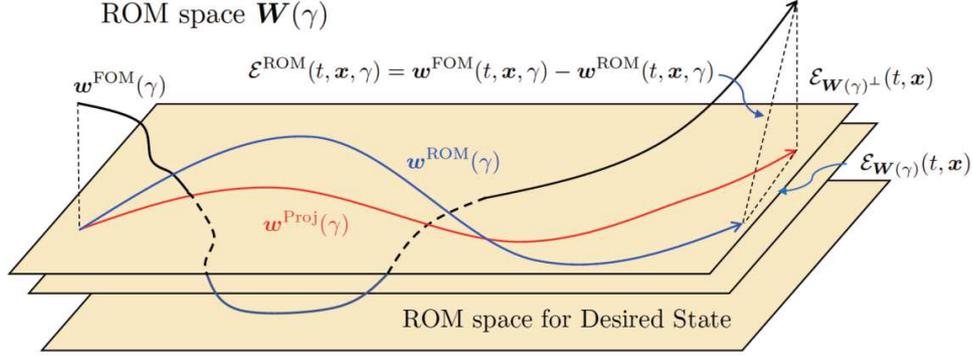


Figure 1: Discrepancy of the snapshot trajectories between the parameterized FOM, ROM, desired state and least squares projected solutions after Galerkin projection.

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

$$\dot{\mathbf{q}}(t) = \mathbb{D} \mathbf{q}(t) + [\mathbb{C} \mathbf{q}(t)] \mathbf{q}(t) + \mathbf{C}(\mathbf{q}, \boldsymbol{\alpha}),$$

where $\alpha_k(t)$ is the least-squares (LS) projection modal coefficients which can be obtained by

$$\alpha_k(t) = \begin{cases} (\mathbf{u}(t, \mathbf{x}), \boldsymbol{\varphi}_k) & \text{for } k = 1, \dots, K, \\ (\boldsymbol{\theta}(t, \mathbf{x}), \boldsymbol{\xi}_k) & \text{for } k = K + 1, \dots, 2K. \end{cases}$$

The LS projection modal coefficients may include some information of the hidden physics and its interaction with the dynamical core of the system. For simplicity, we assume that C_k depends only on q_k and α_k for each k . The correction function C_k can be defined in various way, especially and naturally,

$$C_k(t) = f(\alpha_k(t) - q_k(t)).$$

Appropriate machine learning algorithms can be used to learn this correction term C_k such as ResNET and LSTM networks, so on. In this article, we employ LSTM neural network algorithm to learn the mapping from LS projection modal coefficients to the correction term

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

4.2. LSTM-ROM

In this subsection, we briefly introduce LSTM-ROM. Please refer to [19] for more details. 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

$$\text{Correction} := C_k^{(n)} = f\left(\alpha_k^{(n)} - q_k^{(n)}\right).$$

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 are used as input features to the LSTM network, the parameter γ governing the system's behavior is taken implicitly into account. Fig. 2 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 [2, 25–27] and references therein. Related computer programs can be obtained from GitHub (<https://github.com/>), namely ETC-ROM Master, Hybrid-Modeling Master, UROM Master and mnni-rom Master, ROM-FOM-Coupling-main, so on. Once the model is trained, we could correct the GP modal coefficients with LSTM-based correction to approximate true projection modal coefficients.

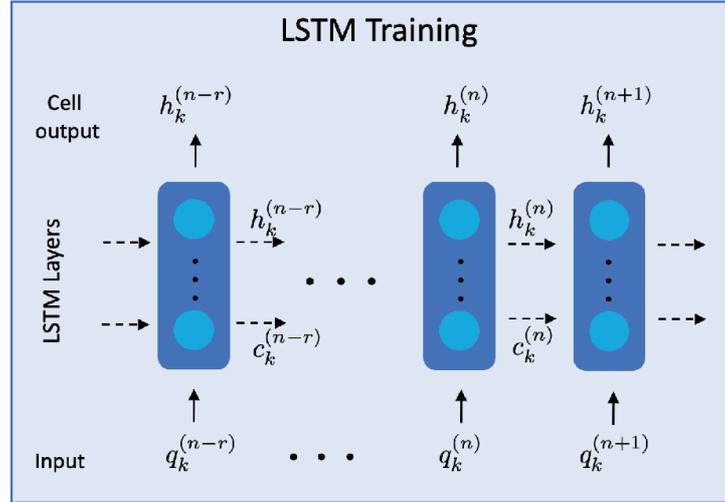


Figure 2: LSTM Cell (left) and LSTM Training where the number of lookbacks $r = 5$.

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, temperature and linear for the pressure — i.e. the Taylor-Hood finite element pair for Navier-Stokes equations, based on a rectangular mesh with $\Delta \mathbf{x} = 1/32$. We use FreeFEM++ [16]. After the snapshot is created, all computations are performed using Python 3.8. We compute the desired states by solving the equations

$$\begin{aligned} \frac{\partial \mathbf{U}}{\partial t} - \nu \Delta \mathbf{U} + \nabla \mathbf{U}^T + (\mathbf{U} \cdot \nabla) \mathbf{U} + \nabla p - \beta \Theta \mathbf{g} &= \mathbf{0} & \text{in } (0, T) \times \Omega, \\ \nabla \cdot \mathbf{U} &= 0 & \text{in } (0, T) \times \Omega, \\ \mathbf{U}|_{\partial\Omega} &= \mathbf{0}, \quad \mathbf{U}(0, \mathbf{x}) = \mathbf{U}_0(\mathbf{x}), \\ \frac{\partial \Theta}{\partial t} - \nabla \cdot (\kappa \nabla \Theta) + (\mathbf{U} \cdot \nabla) \Theta &= 10 \sin(\pi \mathbf{x}) \sin(1.0 + 4t) \pi \mathbf{y} & \text{in } (0, T) \times \Omega, \\ \Theta|_{\partial\Omega} &= 0, \quad \Theta(0, \mathbf{x}) = \Theta_0(\mathbf{x}), \end{aligned}$$

where $\nu = 10$, $\beta = 100$ and $\kappa = 1$. (\mathbf{U}_0, Θ_0) is chosen like in Fig. 3. Once we obtain the desired states (\mathbf{U}, Θ) , we compute the distributed control \mathcal{T} using the Eq. (2.1). In this example, we set $\mathbf{F} = \mathbf{0}$.

Now, generate snapshot set for each $\gamma = 10$, $\gamma = 20$ and $\gamma = 30$ using the following equations:

$$\begin{aligned} &\left(\frac{\partial \mathbf{u}_h}{\partial t}, \mathbf{v}_h \right) + \nu a_0(\mathbf{u}_h, \mathbf{v}_h) + b_0(\mathbf{u}_h; \mathbf{u}_h, \mathbf{v}_h) + c(\mathbf{v}_h, p_h) - \beta d(\theta_h, \mathbf{v}_h) + \gamma_1(\mathbf{u}_h, \mathbf{v}) \\ &= (\gamma_1 \mathbf{U}, \mathbf{v}_h) \quad \text{for all } \mathbf{v}_h \in \mathbb{V}_{h,0}, \\ &c(\mathbf{u}_h, q_h) = 0 \quad \text{for all } q_h \in S_h, \\ &\left(\frac{\partial \theta_h}{\partial t}, \psi_h \right) + \kappa a_1(\theta_h, \psi_h) + b_1(\mathbf{u}_h, \theta_h, \psi_h) + \gamma_2(\theta_h, \psi_h) \\ &= (\mathcal{T} + \gamma_2 \Theta, \psi_h) \quad \text{for all } \psi_h \in X_{h,0} \end{aligned}$$

with the initial conditions

$$\mathbf{u}(0, \mathbf{x}) = -\mathbf{U}_0(\mathbf{x}), \quad \theta(0, \mathbf{x}) = -\Theta_0(\mathbf{x}).$$

One can see the initial conditions for the desired states and controlled states in Fig. 3. This initial velocity rotates in the opposite direction to the target initial velocity \mathbf{U}_0 .

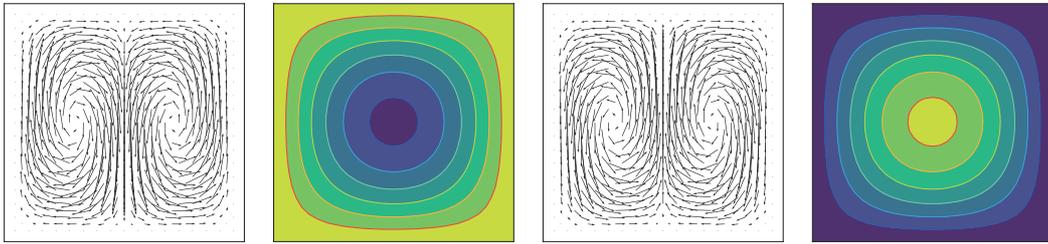


Figure 3: From the left, the initial velocity and temperature for controlled flows, and the velocity and temperature for desired flows.

For this computation, we use $\Delta t = 0.001$, $h = 1/32$ and $T = 0.2$. The number of unknowns (degree of freedom) is 16,900. In this test we are also interested in the convergence history for the parameters involved. In Fig. 4, one can see the exponential convergence of the controlled solution to the desired solutions in time. The solutions over time are shown in Fig. 5.

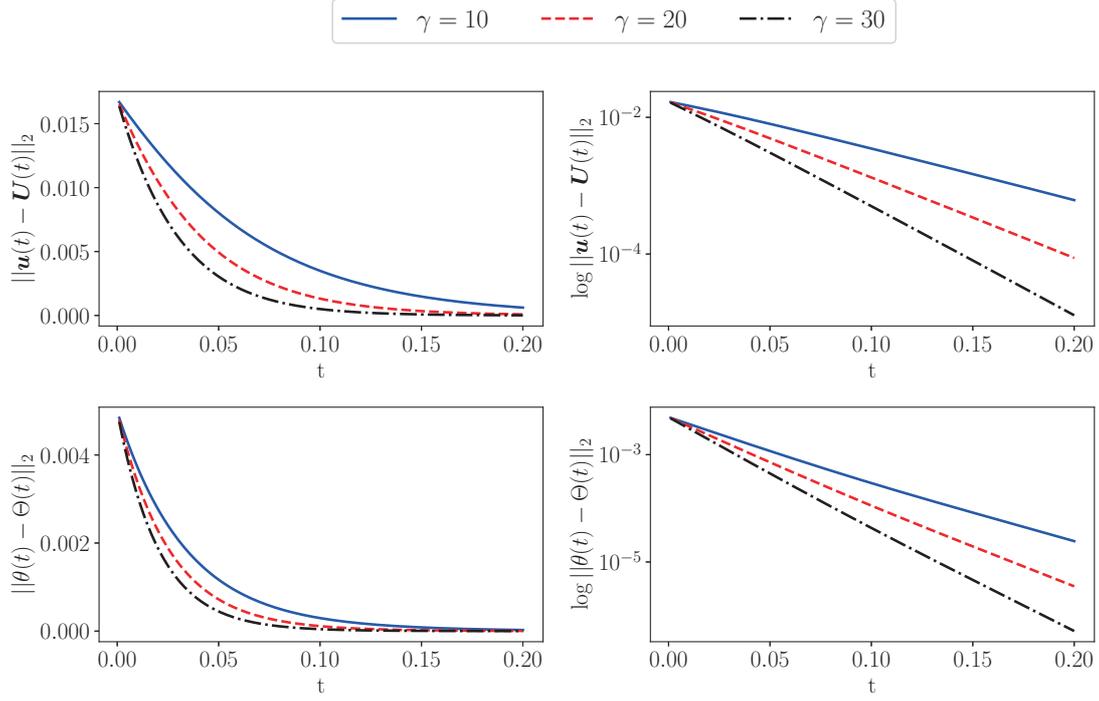


Figure 4: $\|u(t) - U(t)\|_{L^2}$ and $\|\theta(t) - \Theta(t)\|_{L^2}$ in different parameter γ .

5.2. POD ROM

The three set of POD reduced basis for each $\gamma = 10, 20, 30$ are determined from the corresponding snapshot set as described in Section 3. Note that each basis function satisfies the discretized continuity equations — i.e. it is discretely solenoidal. If one perform least-squares projection of each snapshot to corresponding ROM space, the errors for velocity field and temperature field are about 10^{-6} and 10^{-7} , respectively. However, if the desired state is projected on this ROM space, as shown in Fig. 6, it can be seen that their errors are more than 10 times larger than the errors in snapshots. In this case, the controlled state converges to a different desired state.

There are several ways to overcome the above problem. In this article, the basis is derived from the extended snapshot set $\widetilde{W}(\gamma) = W(\gamma) \cup Z$ where Z is the snapshot set of the desired state. In our opinion, this method is the simplest method. In this case, the projection errors are all about the same as shown in Fig. 7.

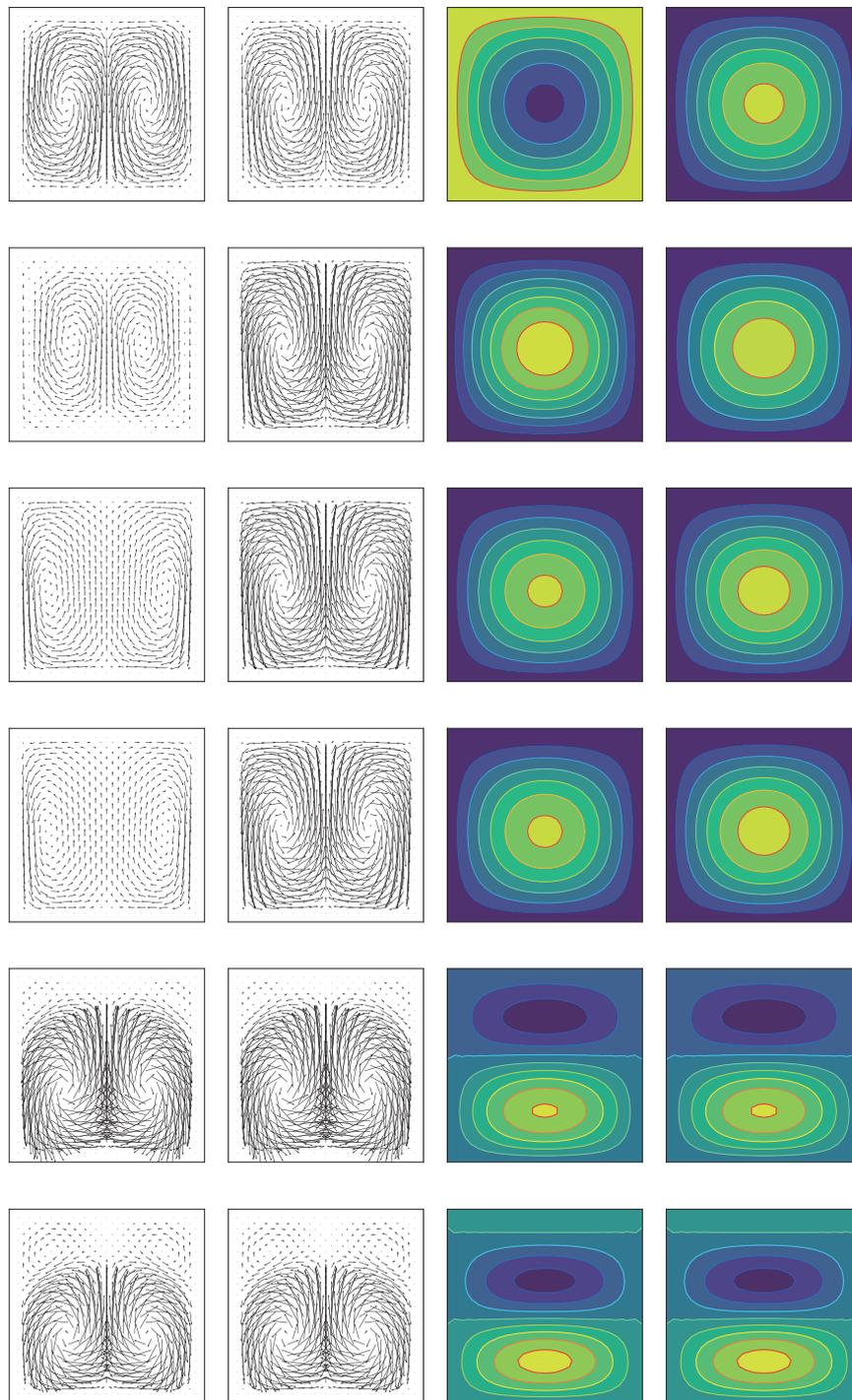


Figure 5: Controlled (left) and target (right) flows (left pair of columns) and temperatures (right pair of columns), $t = 0.0, 0.009, 0.015, 0.02, 0.08, 0.2$; $\gamma = 10$.

The eight-dimensional POD basis functions (velocity and temperature) are displayed in Fig. 8. We retained eight basis functions for each velocity and temperature fields — i.e. $K_u = K_\theta = 8$, as they captured more than 99.99997% of the energy for all control numbers $\gamma = [10, 20, 30]$. In the case of temperature, more energy is concentrated in the first eight bases than in the case of velocity. We consider here, a small-dimensional POD basis can capture most of the information contained in the snapshot set.

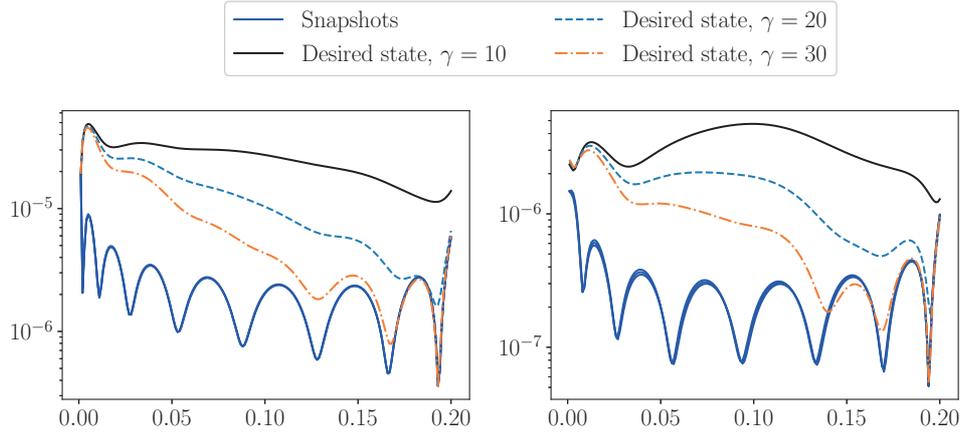


Figure 6: Least-squares projection error. Left: Velocity. Right: Temperature.

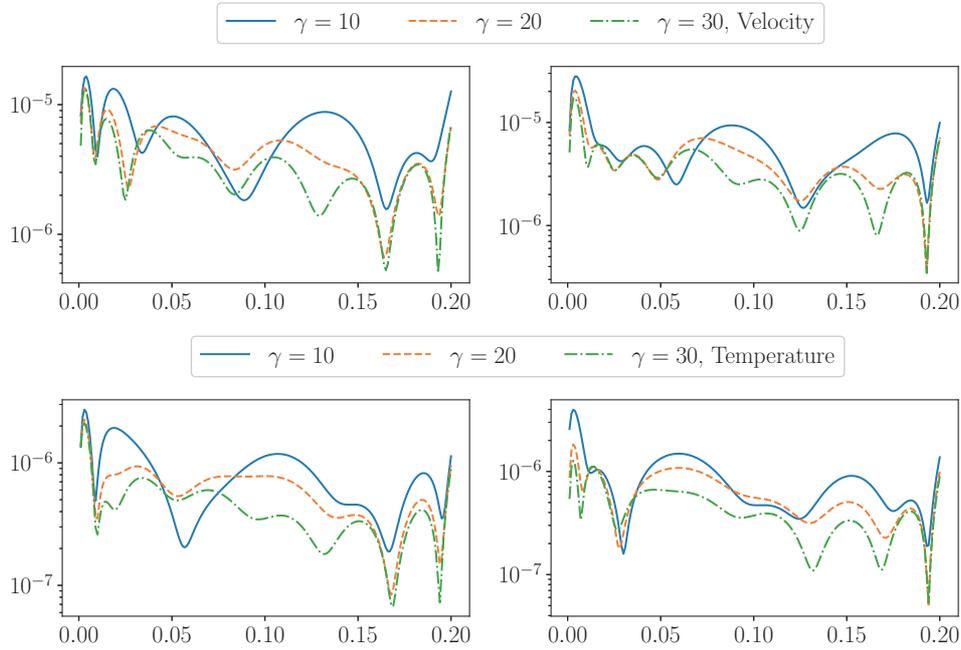


Figure 7: Least-squares projection error. Top: Velocity. Bottom: Temperature. Left: Snapshot set to its own ROM space. Right: Desired state to each ROM space.

Now, given the desired state (U, Θ) , we use K -dimensional system of nonlinear ordinary differential equations (3.6) to determine reduced-order solutions of the Boussinesq system. Approximations of solutions $q_k^{(n)}$ of the system of ordinary differential equation (3.6) are determined using a fourth order Runge-Kutta method.

To quantify the performance of different frameworks, we defined the root mean squared error (RMSE) between the FOM solution and the solutions computed with different ROM frameworks. The RMSE is defined as

$$\text{RMSE}(t_n) = \sqrt{\frac{1}{N} \sum_{i=1}^N ((\mathbf{u}, \theta)^{\text{FOM}}(\mathbf{x}_i, t_n) - (\mathbf{u}, \theta)^{\text{ROM}}(\mathbf{x}_i, t_n))^2},$$

where N represents the spatial resolution — i.e. the number of the interior finite element node.

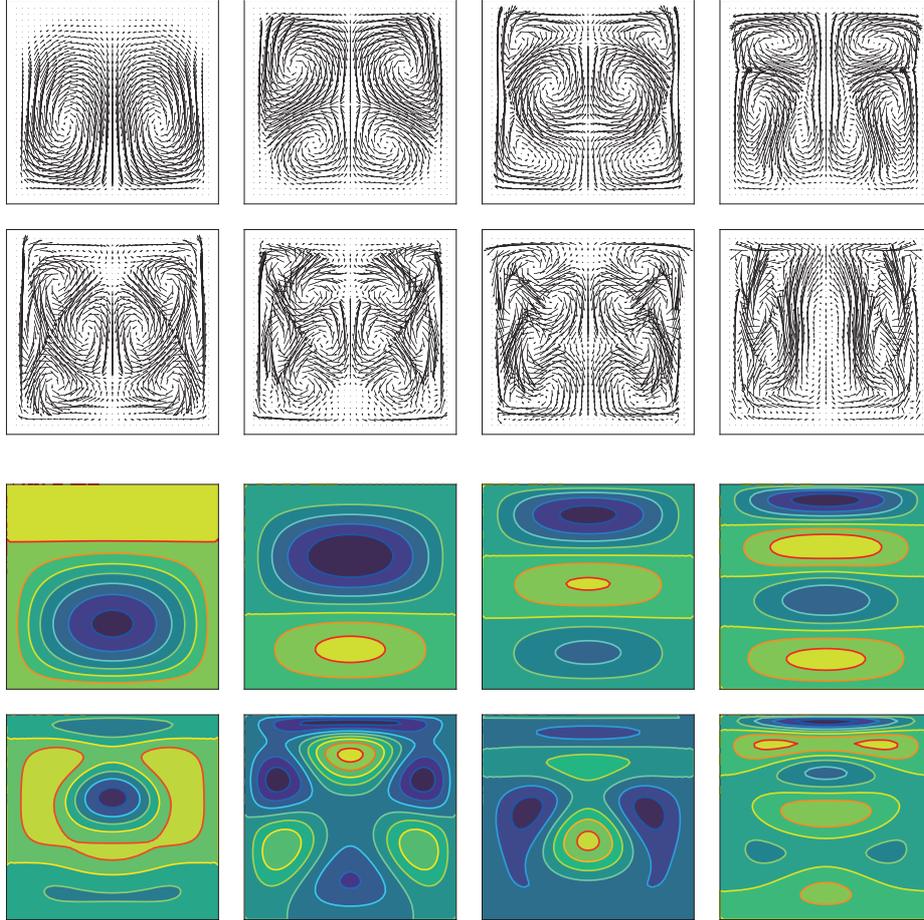


Figure 8: The POD reduced basis of cardinality 8 for velocity and temperature (basis 1,2,3,4 from the left to the right (top) and basis 5,6,7,8 from the left to the right (bottom) in each two rows.

Fig. 9 shows the RMSE(t) over time for three cases $\gamma = [10, 20, 30]$. The LS projection error is

$$\mathcal{E}_{\mathcal{W}(\gamma)^\perp}(t, \mathbf{x}, \gamma) = \mathbf{u}^{\text{FOM}}(t, \mathbf{x}, \gamma) - \mathbf{u}^{\text{Proj}}(t, \mathbf{x}, \gamma)$$

and GP-ROM error is

$$\mathcal{E}^{\text{ROM}}(t, \mathbf{x}, \gamma) = \mathbf{u}^{\text{FOM}}(t, \mathbf{x}, \gamma) - \mathbf{u}^{\text{ROM}}(t, \mathbf{x}, \gamma).$$

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.

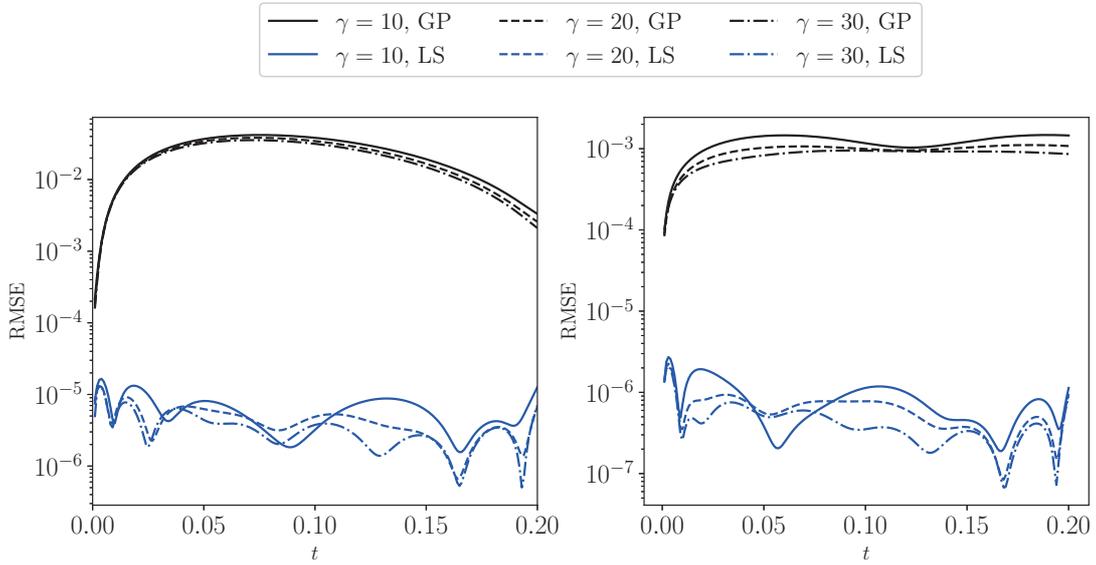


Figure 9: Upper 3 lines: RMSEs of Galerkin ROM. Lower 3 lines: LS projection. Top: Velocity. Bottom: Temperature.

5.3. LSTM-ROM

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

It was carried out according to the method suggested in Section 4.2. A supervised learning framework is applied to model the correction term C with information obtained from FOM and projection data (see Fig. 10). We train our LSTM neural network to learn the mapping from GP modal coefficients to the correction term. Since GP modal coefficients are 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 80 cells. The required settings can be seen in Table 1.

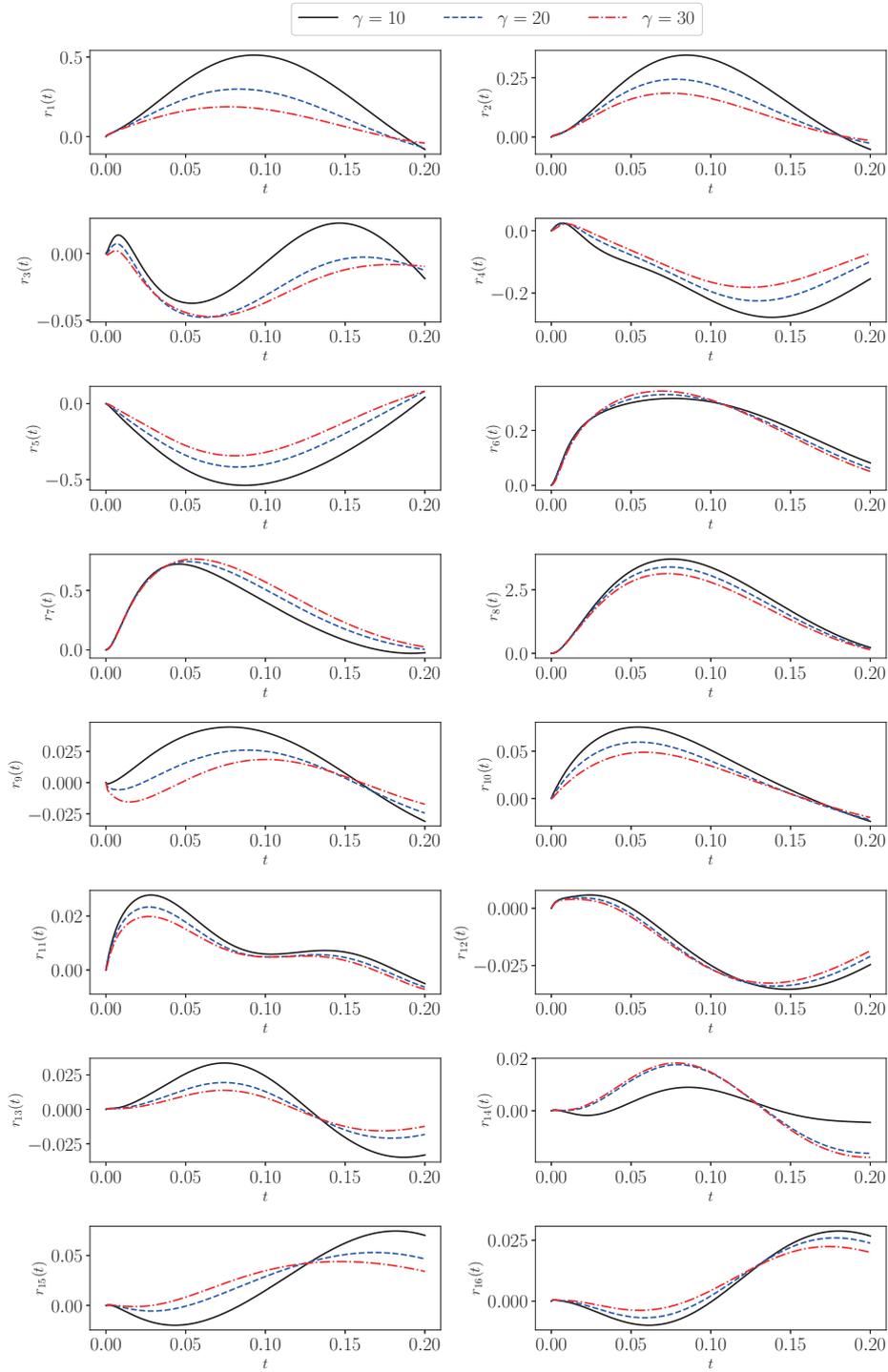


Figure 10: Difference between the coefficients of LS projection and coefficients of ROM solutions $r_k(t) = \alpha_k(t) - q_k(t)$.

Table 1: A list of hyperparameters utilized to train the LSTM network for numerical experiments.

Variables	Hyperparameters
Number of hidden layers	2
Number of neurons in each hidden layer	80
Number of lookbacks	5
Batch size	32
Epochs	800
Activation functions in the LSTM layers	tanh
Validation data set	20%
Loss function	MSE
Optimizer	ADAM

Once the model is trained, we could correct the GP modal coefficients with LSTM-based correction to approximate true projection modal coefficients. We give the numerical procedure as follows:

1. Data generation: For $\gamma = 10, 20, 30$ to use in learning algorithm and $\gamma = 40$ to use a test.
2. Basis construction: For $\gamma = 10, 20, 30$.
3. The velocity and temperature modal coefficients $q_k^{(n)}$ of Galerkin projection reduced order modeling calculation for $\gamma = 10, 20, 30$ using each corresponding basis.
4. LSTM training using $C_k^{(n+1)}$ and $q_k^{(n+1)}$. A summary of the adopted hyper-parameters is presented in Table 1. The training and validation loss during the learning procedure are in Fig. 11.
5. Prediction for $\gamma = 40$ using the basis which is obtained from the snapshot sets with $\gamma = 30$.

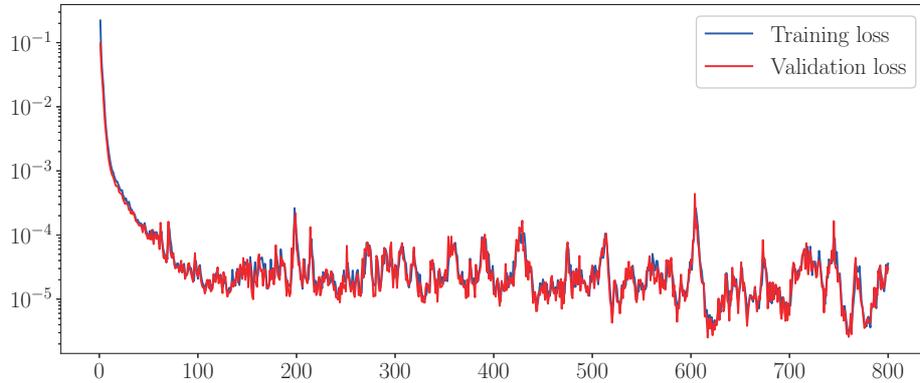


Figure 11: Training and validation loss.

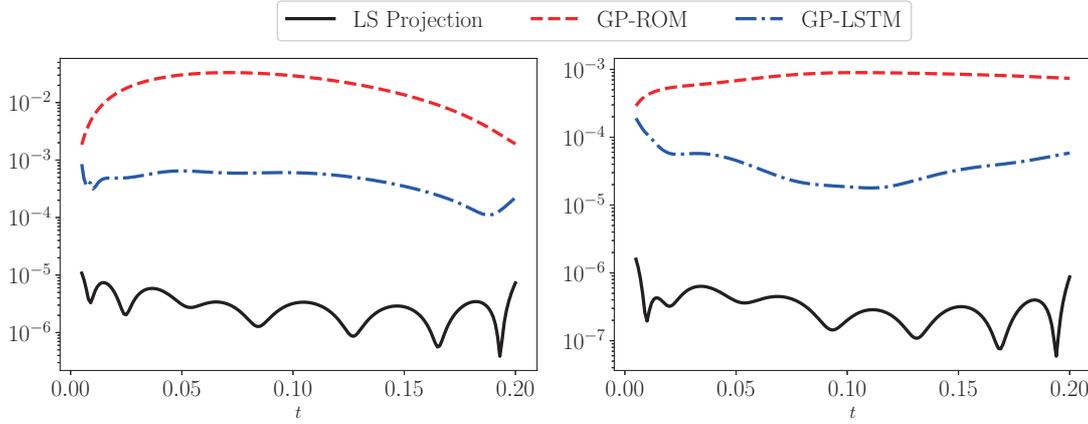


Figure 12: RMSE of ROMs predicted by LS projection, GP-ROM and GP-LSTM for $\gamma = 40$ using the basis of $\gamma = 30$, $0.005 \leq t \leq 0.2$. Left for u and right for θ .

6. Reconstruction and compare with GP-ROM, LS projection and GP-LSTM ROM (see Fig. 12).

In step 5, interpolation techniques such as Grassman manifold interpolation [2, 3, 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.

Finally, we report the online computing time for our numerical experiments. We show the computational time as well as the RMSE of reconstructed fields in average for the test cases at $\gamma = 40$ in Table 2. All calculations were performed on an iMac 3.6Ghz 10 Core. We observe that computing time of GP-LSTM is about twice that of GP-ROM and the error of GP-LSTM is much smaller (about 1/50 for the velocity field and about 1/150 for the temperature field) than that of GP-ROM. From Fig. 12, it can be seen that the results using GP-LSTM have significantly higher accuracy than the results using GP-ROM. If we use as much data as and use interpolations like in other papers, for example [2, 25, 26], one can reduce the error even more. However, in this article, 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.

Table 2: CPU time (in second) comparison for the different ROM frameworks investigated in this study and the average of RMSE.

Framework	Times	RMSE Velocity	(Average) Temperature
FOM	12.14	0	0
LS-Projection		3.22154634e-06	3.63202367e-07
GP-ROM	0.0485	1.93027855e-02	7.78547558e-03
GP-LSTM	0.0837	3.81106095e-04	4.72451491e-05

6. Concluding Remarks

In this study, a simple feedback rule is considered 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

This research was supported by NRF-2019R1F1A1050231 and the National Science Foundation of China (11961073).

References

- [1] R. Adams, *Sobolev Spaces*, Academic Press (1975).
- [2] S. Ahmed, O. San, A. Rasheed and T. Iliescu, *A long short-term memory embedding for hybrid uplifted reduced order models*, *Physica D* **409**, (2020).
- [3] 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).
- [4] R. Archibald, F. Bao, J. Yong and T. Zhou, *An efficient numerical algorithm for solving data driven feedback control problems*, *J. Sci. Comput.* **85(2)**, 58 (2020).
- [5] P. Benner, S. Gugercin and K. Willcox, *A survey of projection-based model reduction methods for parametric dynamical systems*, *SIAM Review* **57(4)**, 483–531 (2015).
- [6] M. Benosman, J. Borggaard, O. San and B. Kramer, *Learning-based robust stabilization for reduced-order models of 2D and 3D Boussinesq equations*, *Appl. Math. Model.* **49**, 162–181 (2017).
- [7] G. Berkooz, P. Holmes and J.L. Lumley, *The proper orthogonal decomposition in the analysis of turbulent flows*, *Annu. Rev. Fluid Mech.* **25**, 539–575 (1993).
- [8] S. Brunton, B. Noack and P. Koumoutsakos, *Machine learning for fluid mechanics*, *Annu. Rev. Fluid Mech.* **52**, 477–508 (2020).
- [9] J. Burkardt, M. Gunzburger and H.-C. Lee, *Centroidal voronoi tessellation-based reduced-order modeling of complex systems*, *SIAM J. Sci. Comput.* **28**, 459–484 (2006).
- [10] J. Burkardt, M. Gunzburger and H.-C. Lee, *POD and CVT-based reduced-order modeling of Navier-Stokes flows*, *Comput. Methods Appl. Mech. Engrg.* **196**, 337–355 (2006).
- [11] M.D. Chekroun, H. Liu and J.C. McWilliams, *Variational approach to closure of nonlinear dynamical systems: Autonomous case*, *J. Stat. Phys.* 1–88 (2019).
- [12] 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**, 74–78 (2014).

- [13] V. Girault and P.A. Raviart, *Finite element method for Navier-Stokes equations: Theory and Algorithms*, Springer (1986).
- [14] M.D. Gunzberger and S. Manservigi, *Analysis and approximation for linear feedback control for tracking the velocity in Navier-Stokes flows*, *Comput. Methods Appl. Mech. Engrg.* **189**, 803–823 (2000).
- [15] J. Harlim, S.W. Jiang, S. Liang and H. Yang, *Machine learning for prediction with missing dynamics*, <http://arxiv.org/abs/1910.05861> (2019).
- [16] F. Hecht, *New development in FreeFem++*, *J. Numer. Math.* **20**, 251–265 (2012).
- [17] P. Holmes, J.L. Lumley and G. Berkooz, *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*, Cambridge (1996).
- [18] 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**, 2318–2331 (2017).
- [19] H.-C. Lee, *Efficient computations for linear feedback control problems for target velocity matching of Navier-Stokes flows via POD and LSTM-ROM*, *ERA*, **29(3)**, 2533–2552 (2021).
- [20] H.-C. Lee and Y. Choi, *Analysis and approximation of linear feedback control problems for the Boussinesq equations*, *CAMWA* **51**, 829–848 (2006).
- [21] H.-C. Lee and B.C. Shin, *Dynamics for linear feedback controlled two-dimensional Bénard equations with distributed controls*, *Appl. Math. Optim.* **44(2)**, 163–175 (2001).
- [22] H.-C. Lee and B.C. Shin, *Piecewise optimal distributed controls for 2D Boussinesq equations*, *Math. Meth. Appl. Sci.* **23**, 227–254 (2000).
- [23] A. Majda, *Strategies for reduced-order models for predicting the statistical responses and uncertainty quantification in complex turbulent dynamical systems*, *SIAM Review*, **60(3)**, 491–549 (2015).
- [24] C. Moua, B.L. Koca, O. Sanb, L.G. Rebholz and T. Iliescu, *Data-driven variational multiscale reduced order models*, *Comput. Methods Appl. Mech. Engrg.* **373** (2021).
- [25] S. Pawar, S. Ahmed, O. San and A. Rasheed, *An evolve-then-correct reduced order model for hidden fluid dynamics*, *Mathematics*, **8(4)**, 570 (2020).
- [26] S. Pawar, S. Ahmed, O. San and A. Rasheed, *Data-driven recovery of hidden physics in reduced order modeling of fluid flows*, *Physics of Fluids* **32(3)**, 036602 (2020).
- [27] M. Rahman, S. Pawar, O. San, A. Rasheed and T. Iliescu, *A non-intrusive reduced order modeling framework for quasi-geostrophic turbulence*, *Phys. Rev. E* **100**, 053306 (2019).
- [28] 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(3)**, 561–590 (1987).
- [29] 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*, [arXivmath.NA/1912.07886](https://arxiv.org/abs/1912.07886)
- [30] K. Xu, B. Shi and S. Yin, *Deep Learning for Partial Differential Equations*, Stanford University (2018).