



# Learning-based robust stabilization for reduced-order models of 2D and 3D Boussinesq equations



Mouhacine Benosman<sup>a,\*</sup>, Jeff Borggaard<sup>b</sup>, Omer San<sup>c</sup>, Boris Kramer<sup>d</sup>

<sup>a</sup> Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA 02139, USA

<sup>b</sup> Interdisciplinary Center for Applied Mathematics, Virginia Tech, Blacksburg, VA 24061, USA

<sup>c</sup> Department of Mechanical and Aerospace Engineering, Oklahoma State University, 218 Engineering NorthStillwater, Oklahoma 74078-5016, USA

<sup>d</sup> Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

## ARTICLE INFO

### Article history:

Received 21 September 2016

Revised 17 April 2017

Accepted 24 April 2017

Available online 8 May 2017

### Keywords:

Reduced-order models

Closure models

Robust Lyapunov control

Extremum-seeking

Boussinesq

## ABSTRACT

We present some results on the stabilization of reduced-order models (ROMs) for thermal fluids. The stabilization is achieved using robust Lyapunov control theory to design a new closure model that is robust to parametric uncertainties. Furthermore, the free parameters in the proposed ROM stabilization method are optimized using a data-driven multi-parametric extremum seeking (MES) algorithm. The 2D and 3D Boussinesq equations provide challenging numerical test cases that are used to demonstrate the advantages of the proposed method.

© 2017 Elsevier Inc. All rights reserved.

## 1. Introduction

The *stable model reduction* problem is well-known in model reduction for partial differential equations (PDEs). Here we consider model reduction by performing Galerkin projection of the PDE onto a suitable low-dimensional set of modes to approximate the PDE solutions using a small system of ordinary differential equations (ODEs). However, using the low-dimensional basis necessitates the truncation of modes that contain small-scale structures, and may be responsible for retaining the main characteristics of the original PDE model, such as stability and prediction precision. Thus additional modeling terms are usually necessary.

In this paper, we focus on reduced-order models (ROMs) where modes are obtained using proper orthogonal decomposition (POD) [1], which has been widely used to obtain surrogate models of tractable size in fluid flow applications. However, it has been observed, e.g., [2–6], that POD-ROMs can lose stability. This loss of stability is due to the truncation of higher-order modes, which are responsible for the dissipation of energy and thus have a stabilizing effect. Maintaining stability is crucial for any ROM that is used for predictions over long time intervals.

We address the stable model reduction problem by adding a closure model term that takes the form of an additive viscosity. This term is added to the ROM equations to ensure the stability and accuracy of solutions. Closure models have classically been introduced based on physical intuition. Thus, their applicability is limited to those applications where significant research in closure models have been performed. In this work, we propose the use of robust control theory to constructively

\* Corresponding author.

E-mail address: [m\\_benosman@ieee.org](mailto:m_benosman@ieee.org) (M. Benosman).

design a new closure model that is robust to parametric uncertainties. There are several closure models motivated from physical modeling of fluids, e.g., constant eddy viscosity model, or time and space varying terms, such as Smagorinsky or dynamic subgrid-scale models e.g., [7–9]. However, there are two main conceptual differences with the closure model that we are proposing here. First of all, we propose a closure model that explicitly accounts for parametric uncertainties in the system. Indeed, we formulate the problem of ROM stabilization at the design step, by considering uncertainties in the ROM parameters, then using tools borrowed from robust control theory, we design a closure model which stabilizes the ROM for a specified range of parametric uncertainties. To our knowledge, this is the first class of closure model that is designed to be robust with respect to parametric uncertainties. All previously defined closure models, are either motivated from physics or simply add extra dissipation in the model. They do not take parametric uncertainties into account, and thus their ability to stability the model is not robust to uncertainties nor changes in the system's parameters. Secondly, in this work we propose to auto-tune the closure model using a data-driven optimization algorithm. This auto-tuning can be used in simulations to find the best closure model by tracking the true behavior of the system, then automatically choosing among a class of robust closure models obtained as described above. However, an important observation is that *this auto-tuning algorithm can be incorporated in real-time simulations*, by feeding actual measurements from the system into the closure model, and adapting its coefficients. In this way, we always ensure that the ROM is functioning at its optimal performance, regardless of changes or drifts that the system may experience over time. In other words, most closure models typically use static parameters, either chosen by intuition and experience, or are optimally tuned off-line. However, they are unable to auto-tune themselves on-line while the model is being evolved. In this work, the obtained closure model has free parameters that are auto-tuned with a data-driven MES algorithm to optimally match the predictions (or measurements) from the PDE model. The idea of using extremum-seeking to auto-tune closure models has been introduced by the authors in [10]. However, the difference with this work lies in the new constructive formulation of *robust* closure models, based on robust Lyapunov control theory. Furthermore, contrary to [10] where the authors considered the simple case of the one-dimensional Burgers' equation, here we study a wide range of formulations of the Boussinesq equations. These models are more challenging and more directly applicable to a number of important real-world control problems [11]. Our Boussinesq formulations include different equation scalings and primitive variable (velocity–temperature) vs. vorticity–stream function–temperature formulations.

Our work extends existing results in the field. Indeed, stable model reduction of Navier–Stokes flow models by adding a nonlinear viscosity term to the reduced-order model is considered in [12]. In [13,14], incompressible flows are stabilized by an iterative search of the projection modes that satisfy a local Lyapunov stability condition. An optimization-based approach for the POD modes of linear models, which solely focused on matching the outputs of the models, is derived in [4,6]. Kalb and Deane [3] added error correction terms to the reduced-order model for improved accuracy and stabilization. Moreover, the authors in [2] calibrated the POD model by solving a quadratic optimization problem based on three different weighted error norms. Stable model reduction for the Navier–Stokes and Boussinesq equations using turbulence closure models was presented in [7–9]. These closure models modify some stability-enhancing coefficients of the reduced-order ODE model using either constant additive terms, such as the constant eddy viscosity model, or time and space varying terms, such as Smagorinsky or dynamic subgrid-scale models. The amplitudes of the additional terms are tuned in such a way to accurately stabilize the reduced-order models to the data at hand.

However, such closure models do not take into account parametric uncertainties in the model, and their tuning is not always straightforward. Our work<sup>1</sup> addresses these issues and proposes, using robust Lyapunov control theory, a new closure model in Section 3 that addresses parametric uncertainties. Furthermore, we achieve optimal auto-tuning of this closure model using a learning-based approach. This is demonstrated using the 2D and 3D Boussinesq equations in Section 4. To set the stage, the following section establishes our notation.

## 2. Basic notation and definitions

For a vector  $q \in \mathbb{R}^n$ , the transpose is denoted by  $q^*$ . The Euclidean vector norm for  $q \in \mathbb{R}^n$  is denoted by  $\|\cdot\|$  so that  $\|q\| = \sqrt{q^*q}$ . The Frobenius norm of a tensor  $A \in \mathbb{R}^{m \times n_1}$ , with elements  $a_i = a_{i_1 \dots i_k}$ , is defined as  $\|A\|_F \triangleq \sqrt{\sum_{i=1}^n |a_i|^2}$ . The Kronecker delta function is defined as:  $\delta_{ij} = 0$ , for  $i \neq j$  and  $\delta_{ii} = 1$ . For a symmetric matrix  $D \in \mathbb{R}^{n \times n}$ ,  $\lambda_{\max}(D)$  denotes its maximum eigenvalue. The 2D and 3D Boussinesq equations are, respectively, solved on the rectangle  $x \in \Omega = (0, 8) \times (0, 1)$  or the unit cube  $x \in \Omega = (0, 1)^3$  for the simulation time interval  $t \in (0, t_f)$ . We shall abbreviate the time derivative by  $\dot{f}(t, x) = \frac{\partial}{\partial t} f(t, x)$ , and consider the following Hilbert spaces:  $\mathcal{H} = L^2(\Omega)$ ,  $\mathcal{V} = H_{\text{div}}^1(\Omega) \subset (\mathcal{H})^d$ ,  $d = 2, 3$  for velocity and  $\mathcal{T} = H^1(\Omega) \subset \mathcal{H}$  for temperature. Thus,  $\mathcal{V}$  is the space of divergence-free vector fields on  $\Omega$  with components in  $H^1(\Omega)$ . Fixed Dirichlet boundary conditions are also specified in the definition of the sets  $\mathcal{V}$  and  $\mathcal{T}$ . We define the inner product  $\langle \cdot, \cdot \rangle_{\mathcal{H}}$  and the associated norm  $\|\cdot\|_{\mathcal{H}}$  on  $\mathcal{H}$  as  $\langle f, g \rangle_{\mathcal{H}} = \int_{\Omega} f(x)g(x)dx$ , for  $f, g \in \mathcal{H}$ , and  $\|f\|_{\mathcal{H}}^2 = \int_{\Omega} |f(x)|^2 dx$ . A function  $T(t, x)$  is in  $L^2([0, t_f]; \mathcal{H})$  if for each  $0 \leq t \leq t_f$ ,  $T(t, \cdot) \in \mathcal{H}$ , and  $\int_0^{t_f} \|T(t, \cdot)\|_{\mathcal{H}}^2 dt < \infty$  with analogous definitions for the vector valued functions in  $(\mathcal{H})^d$ ,  $d = 2, 3$ . To generalize the discussion below, we consider the abstract Hilbert space  $\mathcal{Z}$ , and later specialize to  $\mathcal{Z} = \mathcal{V} \oplus \mathcal{T}$  when considering our Boussinesq equation examples. Finally, in the remainder of this paper we consider the stability of dynamical systems in the sense of Lagrange, e.g., [16]: a system  $\dot{q} = f(t, q)$  is said to be Lagrange stable if for every initial condition  $q_0$  associated with the time instant  $t_0$ , there exists  $\epsilon(q_0)$ , such that  $\|q(t)\| < \epsilon, \forall t \geq t_0 \geq 0$ .

<sup>1</sup> A preliminary version of this work has been posted on arXiv [15].

### 3. Lyapunov-based robust stable model reduction of PDEs

#### 3.1. Reduced-order PDE approximation

We consider a stable dynamical system modeled by a nonlinear partial differential equation of the form

$$\dot{z}(t) = \mathcal{F}(z(t)), \quad z(0) \in \mathcal{Z}, \quad (1)$$

where  $\mathcal{Z}$  is an infinite-dimensional Hilbert space. Solutions to this PDE can be approximated in a finite dimensional subspace  $\mathcal{Z}^n \subset \mathcal{Z}$  through expensive numerical discretization, which can be impractical for multi-query settings such as analysis and design, and even more so for real-time applications such as prediction and control. In many systems, including fluid flows, solutions of the PDE may be well-approximated using only a few suitable (optimal) basis functions [1].

This gives rise to reduced-order modeling through Galerkin projection, which can be broken down into three main steps: one first discretizes the PDE using a finite, but large, number of basis functions, such as piecewise quadratic (for finite element methods), higher-order polynomials (spectral methods), or splines. In this paper we use the well-established finite element method (FEM), and refer the reader to the large literature, e.g., [17], for details. We denote the approximation of the PDE solution by  $z_n(t, \cdot) \in \mathcal{Z}^n$ , where  $\mathcal{Z}^n$  is an  $n$ -dimensional finite element subspace of  $\mathcal{Z}$ .

Secondly, one determines a small set of spatial basis vectors  $\phi_i(\cdot) \in \mathcal{Z}^n$ ,  $i = 1, \dots, r$ ,  $r \ll n$ , that approximates the discretized PDE solution with respect to a pre-specified criterion, i.e.,

$$P_n z(t, x) \approx \Phi q(t) = \sum_{i=1}^r q_i(t) \phi_i(x). \quad (2)$$

Here,  $P_n$  is the projection of  $z(t, \cdot)$  onto  $\mathcal{Z}^n$ , and  $\Phi$  is a matrix containing the basis vectors  $\phi_i(\cdot)$  as column vectors. Note that the dimension  $n$ , coming from the high-fidelity discretization of the PDE described above, is generally very large, in contrast to the dimension  $r$  of the optimal basis set. Thirdly, a Galerkin projection yields a ROM for the coefficient functions  $q(\cdot)$  of the form

$$\dot{q}(t) = F(q(t)), \quad q(0) \in \mathbb{R}^r. \quad (3)$$

The function  $F: \mathbb{R}^r \rightarrow \mathbb{R}^r$  is obtained using the weak form of the original PDE and Galerkin projection.

One significant challenge in this approach lies in the selection of the 'optimal' basis matrix  $\Phi$ , the criterion of optimality used, and selecting the dimension of this basis. There are many approaches to compute those basis functions for nonlinear systems. For example, some of the most common methods are proper orthogonal decomposition (POD) [18], dynamic mode decomposition (DMD) [19], and reduced-basis methods (RBM) [20]. Another challenge for nonlinear problems, in particular, is to model the influence of the discarded basis functions on the retained basis functions  $\Phi$  in the dynamical system for  $q(t)$ .

**Remark 1.** We present the idea of closure models in the framework of POD. However, the derivation is not limited to a particular basis. Indeed, these closure models can be applied to ROMs constructed from other basis functions, such as, DMD. The motivation comes from the fact that any low-dimensional basis necessarily removes the ability to represent the smallest scale structures in the flow and these structures are responsible for energy dissipation. The missing dissipation often must be accounted for with an additional modeling term to ensure accuracy and stability of the ROM.

#### 3.2. Proper orthogonal decomposition for ROMs

POD-based models are most known for retaining a maximal amount of energy in the reduced-order model [1,18]. The POD basis is computed from a collection of  $s$  time snapshots

$$S = \{z_n(t_1, \cdot), \dots, z_n(t_s, \cdot)\} \subset \mathcal{Z}^n, \quad (4)$$

of the dynamical system, usually obtained from a discretized approximation of the PDE model in  $n$  dimensions. The  $\{t_i\}_{i=1}^s$  are time instances at which snapshots are recorded, and do not have to be uniform. The *correlation matrix*  $K$  is then defined as

$$K_{ij} = \frac{1}{s} \langle z_n(t_i, \cdot), z_n(t_j, \cdot) \rangle_{\mathcal{H}}, \quad i, j = 1, \dots, s. \quad (5)$$

The normalized eigenvalues and eigenvectors of  $K$  are denoted by  $\lambda_i$  and  $v_i$ , respectively. Note that the  $\lambda_i$  are also referred to as the *POD eigenvalues*. The  $i$ th *POD basis function* is computed as

$$\phi_i(x) = \frac{1}{\sqrt{s} \sqrt{\lambda_i}} \sum_{j=1}^s [v_i]_j z_n(t_j, x), \quad i = 1, \dots, r, \quad (6)$$

where  $r \leq \min\{s, n\}$  is the number of retained POD basis functions and depends upon the application. The POD basis functions are orthonormal:

$$\langle \phi_i, \phi_j \rangle_{\mathcal{H}} = \int_{\Omega} \phi_i(x)^* \phi_j(x) dx = \delta_{ij}, \quad (7)$$

where  $\delta_{ij}$  denotes the Kronecker delta function.

In this new basis, the solution of the PDE (1) can then be approximated by

$$z_n^{pod}(t, \cdot) = \sum_{i=1}^r q_i(t)\phi_i(\cdot) \in \mathcal{Z}^n, \tag{8}$$

where  $q_i$ ,  $i = 1, \dots, r$  are the POD projection coefficients. To find the coefficients  $q_i(t)$ , the (weak form of the) model (1) is projected onto the  $r$ th-order POD subspace  $\mathcal{Z}^r \subseteq \mathcal{Z}^n \subset \mathcal{Z}$  using a Galerkin projection in  $\mathcal{H}$ . In particular, both sides of Eq. (1) are multiplied by the POD basis functions, where  $z(t)$  is replaced by  $z_n^{pod}(t) \in \mathcal{Z}^n$ , and then both sides are integrated over  $\Omega$ . Using the orthonormality of the POD basis (7) leads to an ODE of the form (3). A projection of the initial condition for  $z(0)$  can be used to determine  $q(0)$ . The Galerkin projection preserves the structure of the nonlinearities of the original PDE.

### 3.3. Closure models for ROM stabilization

We continue to present the problem of stable model reduction in its general form, without specifying a particular type of PDE. However, we now assume an affine dependence of the general PDE (1) on a single physical parameter  $\mu$ ,

$$\dot{z}(t) = \mathcal{F}(z(t), \mu), \quad z(0) = z_0 \in \mathcal{Z}, \quad \mu \in \mathbb{R}, \tag{9}$$

as well as

**Assumption 1.** The solutions of the original PDE model (9) are assumed to be in  $L^2([0, \infty); \mathcal{Z})$ ,  $\forall \mu \in \mathbb{R}$ .

We further assume that the parameter  $\mu$  is critical for the stability and accuracy of the model, i.e., changing the parameter can either make the model unstable, or lead to inaccurate predictions. Since we are interested in fluid dynamics problems, we can consider  $\mu$  as a viscosity coefficient. The corresponding reduced-order POD model takes the form (3) and (8):

$$\dot{q}(t) = F(q(t), \mu). \tag{10}$$

The issue with this Galerkin POD-ROM (denoted POD-ROM-G) is that the norm of  $q$ , and hence  $z_n^{pod}$ , might become unbounded at a finite time, even if the solution of (9) is bounded (Lagrange stable).

The main idea behind the closure modeling approach is to replace the viscosity coefficient  $\mu$  in (10) by a virtual viscosity coefficient  $\mu_{cl}$ , whose form is chosen to stabilize the solutions of the POD-ROM (10). Furthermore, a penalty term  $H(\cdot)$  is added to the original POD-ROM-G, as follows

$$\dot{q}(t) = F(q(t), \mu_{cl}) + H(q(t)). \tag{11}$$

The term  $H(\cdot)$  is chosen depending on the structure of  $F(\cdot, \cdot)$  to stabilize the solutions of (11). To make the paper more self-contained, we give in the next section some examples of the most common closure models.

### 3.4. Closure model examples

We present here some existing closure models, often motivated from the physics of the system. These examples illustrate the principles behind closure modeling, and motivate our proposed method. Throughout,  $r$  denotes the total number of modes retained in the ROM.

-ROM-H model: the first eddy viscosity model, known as the Heisenberg ROM (ROM-H) is simply given by the constant viscosity coefficient

$$\mu_{cl} = \mu + \mu_e, \tag{12}$$

where  $\mu$  is the nominal value of the viscosity coefficient in (9), and  $\mu_e$  is the additional constant term added to compensate for the damping effect of the truncated modes.

-ROM-R model: this model is a variation of the first one, introduced in [21]. In this model,  $\mu_{cl}$  is dependent on the mode index, and the viscosity coefficients for each mode are given by

$$\mu_{cl} = \mu + \mu_e \frac{i}{r}, \tag{13}$$

with  $\mu_e$  being the viscosity amplitude, and  $i$  the mode index.

-ROM-RQ model: this model proposed in [8], is a quadratic version of the ROM-R, which we refer to as ROM-RQ. It is given by the coefficients

$$\mu_{cl} = \mu + \mu_e \left(\frac{i}{r}\right)^2, \tag{14}$$

where the variables are defined similarly to (13).

–ROM-RS model: this model proposed in [8], is a root-square version of the ROM-R; we use ROM-RS to refer to it. It is given by

$$\mu_{cl} = \mu + \mu_e \sqrt{\frac{i}{r}}, \tag{15}$$

where the coefficients are defined as in (13).

–ROM-T model: known as spectral vanishing viscosity model, is similar to the ROM-R in the sense that the amount of induced damping changes as function of the mode index. This concept has been introduced by Tadmor [22], and so these closure models are referred to as ROM-T. These models are given by

$$\begin{aligned} \mu_{cl} &= \mu, & \text{for } i \leq m \\ \mu_{cl} &= \mu + \mu_e, & \text{for } i > m \end{aligned} \tag{16}$$

where  $i$  denotes the mode index, and  $m \leq r$  is the index of modes above which a nonzero damping is introduced.

–ROM-SK model: introduced by Sirisup and Karniadakis [23], falls into the class of vanishing viscosity models. We use ROM-SK to refer to it; it is given by

$$\begin{aligned} \mu_{cl} &= \mu + \mu_e e^{\frac{-(i-r)^2}{(i-m)^2}}, & \text{for } i \leq m \\ \mu_{cl} &= \mu, & \text{for } i > m, m \leq r \end{aligned} \tag{17}$$

–ROM-CLM model: this model has been introduced in [24,25], and is given by

$$\mu_{cl} = \mu + \mu_e \alpha_0^{-1.5} (\alpha_1 + \alpha_2 e^{-\frac{\alpha_3 r}{i}}), \tag{18}$$

where  $i$  is the mode index, and  $\alpha_0, \alpha_1, \alpha_2, \alpha_3$  are positive gains (see [24,26] for some insight about their tuning).

### 3.5. Main result 1: Lyapunov-based closure model

Here we introduce the first main result of this paper, namely a Lyapunov-based closure model that is robust to parametric uncertainties. We first rewrite the right-hand side of the ROM model (10) to isolate the linear viscous term as follows,

$$F(q(t), \mu) = \tilde{F}(q(t)) + \mu Dq(t), \tag{19}$$

where  $D \in \mathbb{R}^{r \times r}$  represents a constant, symmetric negative definite matrix, and the function  $\tilde{F}(\cdot)$  represents the remainder of the ROM model, i.e., the part without damping.

We now consider the case where  $\tilde{F}(\cdot)$  might be unknown, but bounded by a known function. This includes the case of parametric uncertainties in (9) that produce structured uncertainties in (19). To treat this case, we use Lyapunov theory and propose a nonlinear closure model that robustly stabilizes the ROM in the sense of Lagrange. Assume that  $\tilde{F}(\cdot)$  satisfies

**Assumption 2 (Boundedness of  $\tilde{F}$ )** The norm of the vector field  $\tilde{F}(\cdot)$  is bounded by a known  $C^1$  function of  $q$ , i.e.,  $\|\tilde{F}(q)\| \leq \tilde{f}(q)$ .

**Remark 2.** Assumption 2 allows us to consider a general class of PDEs and their associated ROMs. Indeed, all we require is that the right-hand side of (10) can be decomposed as (19), where a linear damping term can be extracted and the remaining nonlinear term  $\tilde{F}$  is bounded. This could allow for more general parametric dependencies and includes many structured uncertainties of the ROM, e.g., a bounded parametric uncertainty can be formulated in this manner.

We now present our first main result.

**Theorem 1.** Consider the PDE (9) under Assumption 1, together with its stabilized ROM model

$$\dot{q}(t) = \tilde{F}(q(t)) + \mu_{cl} Dq(t) + H(q(t)), \tag{20}$$

where  $\tilde{F}(\cdot)$  satisfies Assumption 2,  $D \in \mathbb{R}^{r \times r}$  is symmetric negative definite, and  $\mu_{cl}$  is given by

$$\mu_{cl} = \mu + \mu_e. \tag{21}$$

Here  $\mu$  is the nominal value of the viscosity coefficient in (9), and  $\mu_e$  is the additional constant term. Then, the nonlinear closure model

$$H(q) = \mu_{nl} \tilde{f}(q) \text{diag}(d_{11}, \dots, d_{rr}) q, \quad \mu_{nl} > 0 \tag{22}$$

stabilizes the solutions of the ROM to the invariant set

$$S = \{q \in \mathbb{R}^r \text{ s.t. } \mu_{cl} \frac{\lambda_{\max}(D)\|q\|}{\tilde{f}(q)} + \mu_{nl}\|q\| \max\{d_{11}, \dots, d_{rr}\} + 1 \geq 0\}.$$

**Proof.** First, we prove that the nonlinear closure model (22) stabilizes the ROM (20) to an invariant set. To do so, we use the following energy-like Lyapunov function

$$V(q) = \frac{1}{2} q^* q. \tag{23}$$

We then evaluate the derivative of  $V$  along the solutions of (20), and use (22) and Assumption 2 to write

$$\begin{aligned} \dot{V} &= q^* (\tilde{F}(q) + \mu_{cl} Dq + \mu_{nl} \tilde{f}(q) \text{diag}(d_{11}, \dots, d_{rr}) q) \\ &\leq \|q\| \tilde{f}(q) + \mu_{cl} \|q\|^2 \lambda_{\max}(D) + \mu_{nl} \tilde{f}(q) \|q\|^2 \max\{d_{11}, \dots, d_{rr}\} \\ &\leq \|q\| \tilde{f}(q) (1 + \mu_{cl} \frac{\lambda_{\max}(D) \|q\|}{\tilde{f}(q)} + \mu_{nl} \max\{d_{11}, \dots, d_{rr}\} \|q\|). \end{aligned}$$

This shows convergence to the invariant set  $S$ . □

**Remark 3.** Note that  $\lambda_{\max}(D)$  and  $\max\{d_{11}, \dots, d_{rr}\}$  are negative, thus the sizes of  $\mu_{cl}$  and  $\mu_{nl}$  directly influence the size of  $S$ . It is also apparent how the use of the term  $H$  offers robustness when the uncertainty in  $F(\cdot, \cdot)$  is not precisely known.

**Remark 4.** One can see that the closure model in (22) is written in terms of the diagonal values of the damping matrix  $D$ , the reason for that is twofold: first, we need a matrix term for dimensionality coherence with the vector  $q$ . Second, we need a negative term to be able to upper-bound the norm  $q$  by a positive term, and define the invariant set. Furthermore, using the natural diagonal damping terms implies that the free parameter  $\mu_{nl}$  will be naturally scaled as function of the system’s damping, this allows us to use one tuning parameter instead of  $r$  parameters, to optimally tune the closure model performance, as shown in the next section.

### 3.6. Main result 2: MES-based closure model auto-tuning

As discussed in the introduction as well as in [9], tuning the closure model amplitudes is important to achieve an optimal stabilization of the ROM. In this study, we use model-free MES optimization algorithms to tune the coefficients  $\mu_e$  and  $\mu_{nl}$  of the closure model presented in Section 3.5. An advantage of using MES over other optimization approaches is the auto-tuning capability that such algorithms allow for, as well as their ability to continually tune the closure model, *even during on-line operation of the system*. Indeed, MES algorithms are data-driven, therefore they can be used on-line, based on realtime measurements, to update the closure model coefficients.

Note that MES-based closure model auto-tuning has many advantages. First of all, the closure models can be valid for longer time intervals when compared to standard closure models with constant coefficients that are identified off-line over a (fixed) finite time interval. Secondly, the optimality of the closure model ensures that the ROM obtains the most accuracy for a given low-dimensional basis, leading to the smallest possible ROM for a given application.

We begin by defining a suitable learning cost function for the MES algorithm. The goals of the learning (or tuning) are (i) to enforce Lagrange stability of the ROM model (10), ensured by the proper design of the closure model (as introduced in Theorem 1), and (ii) to ensure that the solutions of the ROM (10) are close to those of the approximation  $z_n(t, \cdot)$  to the original PDE (9). The latter learning goal is important for the accuracy of the solution.

We define the learning cost as a positive definite function of the norm of the error between the approximate solutions of (9) and the ROM (11),

$$\begin{aligned} Q(\hat{\mu}) &= \tilde{H}(e_z(t, \hat{\mu})), \\ e_z(t, \hat{\mu}) &= z_n^{pod}(t, x; \hat{\mu}) - z_n(t, x; \mu), \end{aligned} \tag{24}$$

where  $\hat{\mu} = [\hat{\mu}_e, \hat{\mu}_{nl}]^* \in \mathbb{R}^2$  denotes the learned parameters, and  $\tilde{H}(\cdot)$  is a positive definite function of  $e_z$ . Note that the error  $e_z$  could be computed off-line using solutions of the ROM (8), (11) and exact (e.g., FEM-based) solutions of the PDE (9). The error could be also computed on-line where the  $z_n^{pod}(t, x; \hat{\mu})$  is obtained from solving the ROM model (8), (11) on-line, and the  $z_n(t, x; \mu)$  are real measurements of the system at selected spatial locations  $\{x_i\}$ . The latter approach would circumvent the FEM model, and directly operate on the system, making the reduced-order model more consistent with respect to the operating plant.

A practical way to implement the MES-based tuning of  $\hat{\mu}$ , is to begin with an off-line tuning of the closure model. One then uses the obtained ROM (with the optimal values of  $\hat{\mu}$ , namely  $\mu^{opt}$ ) in the on-line operation of the system, e.g., control and estimation. We can then fine-tune the ROM on-line by continuously learning the best value of  $\hat{\mu}$  at any given time during the operation of the system.

To derive formal convergence results, we introduce some classical assumptions on the learning cost function.

**Assumption 3.** The cost function  $Q(\cdot)$  in (24) has a local minimum at  $\hat{\mu} = \mu^{opt}$ .

**Assumption 4.** The cost function  $Q(\cdot)$  in (24) is analytic and its variation with respect to  $\mu$  is bounded in the neighborhood of  $\mu^{opt}$ , i.e.,  $\|\nabla_{\mu} Q(\tilde{\mu})\| \leq \xi_2$ ,  $\xi_2 > 0$ , for all  $\tilde{\mu} \in \mathcal{N}(\mu^{opt})$ , where  $\mathcal{N}(\mu^{opt})$  denotes a compact neighborhood of  $\mu^{opt}$ .

Under these assumptions the following lemma holds.

**Lemma 1.** Consider the PDE (9) under Assumption 1, together with its ROM model (20), (21), and (22). Furthermore, suppose the closure model amplitudes  $\hat{\boldsymbol{\mu}} = [\mu_e, \mu_{nl}]^*$  are tuned using the MES algorithm

$$\begin{aligned} \dot{y}_1(t) &= a_1 \sin\left(\omega_1 t + \frac{\pi}{2}\right) Q(\hat{\boldsymbol{\mu}}), \\ \hat{\mu}_e(t) &= y_1 + a_1 \sin\left(\omega_1 t - \frac{\pi}{2}\right), \\ \dot{y}_2(t) &= a_2 \sin\left(\omega_2 t + \frac{\pi}{2}\right) Q(\hat{\boldsymbol{\mu}}), \\ \hat{\mu}_{nl}(t) &= y_2 + a_2 \sin\left(\omega_2 t - \frac{\pi}{2}\right), \end{aligned} \tag{25}$$

where  $\omega_{\max} = \max(\omega_1, \omega_2) > \omega^{\text{opt}}$ ,  $\omega^{\text{opt}}$  large enough, and  $Q(\cdot)$  is given by (24). Let  $e_{\boldsymbol{\mu}}(t) := [\mu_e^{\text{opt}} - \hat{\mu}_e(t), \mu_{nl}^{\text{opt}} - \hat{\mu}_{nl}(t)]^*$  be the error between the current tuned values, and the optimal values  $\mu_e^{\text{opt}}, \mu_{nl}^{\text{opt}}$ . Then, under Assumptions 3 and 4, the norm of the distance to the optimal values admits the following bound

$$\|e_{\boldsymbol{\mu}}(t)\| \leq \frac{\xi_1}{\omega_{\max}} + \sqrt{a_1^2 + a_2^2}, \quad t \rightarrow \infty, \tag{26}$$

where  $a_1, a_2 > 0, \xi_1 > 0$ , and the learning cost function approaches its optimal value within the following upper-bound

$$\|Q(\hat{\boldsymbol{\mu}}) - Q(\boldsymbol{\mu}^{\text{opt}})\| \leq \xi_2 \left(\frac{\xi_1}{\omega} + \sqrt{a_1^2 + a_2^2}\right), \tag{27}$$

as  $t \rightarrow \infty$ , where  $\xi_2 = \max_{\boldsymbol{\mu} \in \mathcal{N}(\boldsymbol{\mu}^{\text{opt}})} \|\nabla_{\boldsymbol{\mu}} Q(\boldsymbol{\mu})\|$ .

**Proof.** Based on Assumptions 3 and 4, the extremum seeking nonlinear dynamics (25), can be approximated by a linearly averaged dynamic model (using an averaging approximation over time, [27, p. 435, Definition 1]). Furthermore,  $\exists \xi_1, \omega^{\text{opt}}$ , such that for all  $\omega > \omega^{\text{opt}}$ , the solution of the averaged model  $\hat{\boldsymbol{\mu}}_{\text{aver}}(t)$  is locally close to the solution of the original MES dynamics, and satisfies [27, p. 436]

$$\|\hat{\boldsymbol{\mu}}(t) - \mathbf{d}(t) - \hat{\boldsymbol{\mu}}_{\text{aver}}(t)\| \leq \frac{\xi_1}{\omega}, \quad \xi_1 > 0, \quad \forall t \geq 0,$$

with  $\mathbf{d}(t) = [a_1 \sin(\omega_1 t - \frac{\pi}{2}), a_2 \sin(\omega_2 t - \frac{\pi}{2})]^*$ . Moreover, since  $Q(\cdot)$  is analytic it can be approximated locally in  $\mathcal{N}(\boldsymbol{\mu}^{\text{opt}})$  with a quadratic function, e.g., Taylor series up to second-order, which leads to [27, p. 437],

$$\lim_{t \rightarrow \infty} \hat{\boldsymbol{\mu}}_{\text{aver}}(t) = \boldsymbol{\mu}^{\text{opt}}.$$

Based on the above, we can write

$$\|\hat{\boldsymbol{\mu}}(t) - \boldsymbol{\mu}^{\text{opt}}\| - \|\mathbf{d}(t)\| \leq \|\hat{\boldsymbol{\mu}}(t) - \boldsymbol{\mu}^{\text{opt}} - \mathbf{d}(t)\| \leq \frac{\xi_1}{\omega},$$

so that

$$\|\hat{\boldsymbol{\mu}}(t) - \boldsymbol{\mu}^{\text{opt}}\| \leq \frac{\xi_1}{\omega} + \|\mathbf{d}(t)\|, \quad t \rightarrow \infty,$$

which implies

$$\|\hat{\boldsymbol{\mu}}(t) - \boldsymbol{\mu}^{\text{opt}}\| \leq \frac{\xi_1}{\omega} + \sqrt{a_1^2 + a_2^2}, \quad \xi_1 > 0, \quad t \rightarrow \infty.$$

Next, the cost function upper-bound is easily obtained from the previous bound, using the fact that  $Q(\cdot)$  is locally Lipschitz, with Lipschitz constant  $\xi_2 = \max_{\boldsymbol{\mu} \in \mathcal{N}(\boldsymbol{\mu}^{\text{opt}})} \|\nabla_{\boldsymbol{\mu}} Q(\boldsymbol{\mu})\|$ .  $\square$

**Remark 5.** At this point, the reader might be thinking that by merging together the result of Theorem 1 and Lemma 1, the stability result of Theorem 1 might be jeopardized. Indeed, it is well known that for the stability in the sense of Lyapunov, switching between a sequence of stable models might lead to instability, e.g., [28], however, in this work we are using the concept of stability in the sense of boundedness, i.e., Lagrange stability [29], in which case the switching between (Lagrange) stable models, will still maintain the boundedness, with changing bounds. Indeed, the only thing that will change due to the change of the parameters  $\mu_{cl}, \mu_{nl}$  is the upper-bound defining the invariant set  $S$ . This concept is often used in modular learning adaptive control, where modularity is achieved by imposing boundedness, e.g., in the sense of input-to-state stability (ISS), and then complementing the ISS dynamics with a learning algorithm, e.g., refer to [29] and references therein.

#### 4. The Boussinesq equations

As an example to demonstrate our approach, we consider the incompressible Boussinesq equations that describe the evolution of velocity  $\mathbf{v}$ , pressure  $p$ , and temperature  $T$  of a fluid. This system serves as a model for the flow of air in a room. The coupled equations reflect the conservation of momentum, mass, and energy, respectively,

$$\rho \left( \frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} \right) = -\nabla p + \nabla \cdot \boldsymbol{\tau}(\mathbf{v}) + \rho \mathbf{g}, \tag{28}$$

$$\nabla \cdot \mathbf{v} = 0, \tag{29}$$

$$\rho c_p \left( \frac{\partial T}{\partial t} + \mathbf{v} \cdot \nabla T \right) = \nabla \cdot (\kappa \nabla T), \tag{30}$$

where the buoyancy force is driven by changes in density  $\rho = \rho_0 + \Delta\rho$ , and is modeled as perturbations from the nominal temperature  $T_0$  using the perfect gas law  $\Delta\rho\mathbf{g} \approx -\rho_0\beta(T - T_0)\mathbf{g}$ ,  $\beta = 1/T_0$ , and the term  $\rho_0\mathbf{g}$  is absorbed into the pressure. The viscous stress is  $\tau(\mathbf{v}) = \rho\nu(\nabla\mathbf{v} + \nabla\mathbf{v}^T)$  with kinematic viscosity  $\nu$  and thermal conductivity  $\kappa$ , and the gravitational acceleration is  $\mathbf{g} = -g\hat{\mathbf{e}}_3$ . One typically non-dimensionalizes these equations in different ways depending on the application. For this study, we consider two different non-dimensionalizations, one typically used in engineering applications and one specific to free-convection studies that does not involve a characteristic velocity. We first present the standard engineering non-dimensionalization and introduce the other with our numerical example. By introducing a characteristic length  $L$ , characteristic velocity  $\mathbf{v}_0$ , wall temperature  $T_w$ , and defining  $\tilde{x} = \frac{x}{L}$ ,  $\tilde{t} = \frac{t\nu_0}{L}$ ,  $\tilde{\mathbf{v}} = \frac{\mathbf{v}}{\mathbf{v}_0}$ ,  $\tilde{p} = \frac{p}{\rho\nu_0^2}$ , and  $\tilde{T} = \frac{T-T_0}{T_w-T_0}$  we can reduce the number of free parameters to three. These are the Reynolds number  $\text{Re} = \frac{\mathbf{v}_0 L}{\nu}$ , the Grashof number  $\text{Gr} = \frac{g\beta(T_w-T_0)L^3}{\nu^2}$ , and the Prandtl number  $\text{Pr} = \frac{\nu}{\kappa/\rho c_p}$ . Thus,

$$\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} = -\nabla p + \nabla \cdot \tau(\mathbf{v}) + \frac{\text{Gr}}{\text{Re}^2} T \hat{\mathbf{e}}_3, \tag{31}$$

$$\nabla \cdot \mathbf{v} = 0, \tag{32}$$

$$\frac{\partial T}{\partial t} + \mathbf{v} \cdot \nabla T = \nabla \cdot \left( \frac{1}{\text{RePr}} \nabla T \right), \tag{33}$$

where  $\tau(\mathbf{v}) = \frac{1}{\text{Re}}(\nabla\mathbf{v} + \nabla\mathbf{v}^T)$  and we have dropped the tilde notation.

Following a Galerkin projection onto the subspace spanned by the POD basis functions, the Boussinesq equation is reduced to a POD-ROM with the following structure (details included in [Appendix A](#)),

$$\dot{q}(t) = \mu D q(t) + [Cq(t)]q(t), \tag{34}$$

$$\mathbf{v}^{pod}(x, t) = \mathbf{v}_0(x) + \sum_{i=1}^{r_v} q_i(t)\phi_i^v(x), \tag{35}$$

$$T^{pod}(x, t) = T_0(x) + \sum_{i=r_v+1}^{r_v+r_T} q_i(t)\phi_i^T(x), \tag{36}$$

where  $\mu > 0$  is the viscosity, i.e., the inverse of the Reynolds number in (31) or the Prandtl number in our second non-dimensionalization,  $D$  is a negative definite diffusion matrix with diagonal blocks corresponding to the viscous stress and thermal diffusion (scaled by  $\text{Pr}$  to extract the parameter  $\mu$ ), and  $C$  is a three-dimensional tensor corresponding to the convection terms in (31) and (33).

**Remark 6.** For this Boussinesq example, we could maintain one set of POD coefficients for both velocity and temperature [30]. This would be reasonable for the class of free-convection problems considered here. However, to accommodate forced- and mixed-convection problems, we apply the POD procedure below for velocity and temperature data separately. We continue to use the framework in (3) and consider separate basis functions for velocity,  $\phi_i = [(\phi_i^v)^*; 0^*]^*$  for  $i = 1, \dots, r_v$  and temperature,  $\phi_{r_v+i} = [0^*; (\phi_i^T)^*]^*$  for  $i = 1, \dots, r_T$ . The different groups of coefficient functions  $\{q_i\}_{i=1}^{r_v}$  and  $\{q_i\}_{i=r_v+1}^{r_v+r_T}$  (with  $r = r_v + r_T$ ) are associated with the independent variables  $\mathbf{v}$  and  $T$ , respectively.

We notice that this POD-ROM has mainly a linear term and two quadratic terms, so that it can be written in the form (19), with

$$\tilde{F}(q) = [Cq]q.$$

If we consider bounded parametric uncertainties for the entries of  $C$ , we can write

$$\tilde{F}(q) = [(C + \Delta C)q]q,$$

where  $\|C + \Delta C\|_F \leq c_{\max}$ , we have the upper-bound

$$\|\tilde{F}(q)\| \leq \tilde{f}(q) \equiv c_{\max}\|q\|^2.$$

In this case the nonlinear closure model (22) is

$$H(q) = \mu_{nl}c_{\max}\|q\|^2 \text{diag}(d_{11}, \dots, d_{rr})q, \tag{37}$$

for  $\mu_{nl} > 0$  with  $d_{ii}$ ,  $i = 1, \dots, r$  being the diagonal elements of  $D$ .

#### 4.1. Numerical solvers for the Boussinesq equations

##### 2D Boussinesq solver

The numerical solution for the 2D Rayleigh–Benard convection problem follows the scheme outlined in [31]. Compact fourth-order Padé finite difference schemes are used for the spatial derivatives as in [32]. The first derivative is approximated using the stencil

$$\alpha f'_{i-1} + f'_i + \alpha f'_{i+1} = \frac{a}{2h} (f_{i+1} - f_{i-1})$$

with  $\alpha = 1/4$  and  $a = \frac{2}{3}(\alpha + 2)$ . The second derivative is approximated using the stencil

$$\alpha f''_{i-1} + f''_i + \alpha f''_{i+1} = \frac{a}{h^2} (f_{i+1} - 2f_i + f_{i-1})$$

with  $\alpha = 1/10$  and  $a = \frac{4}{3}(1 - \alpha)$ . Time integration is performed using the optimal third-order total variation diminishing Runge–Kutta (TVDRK3) scheme given in [35]. The stream function  $\psi$  can be obtained from the vorticity using the Poisson equation

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = -\omega.$$

This is solved using the Mehrstellen scheme [36].

##### 3D Boussinesq solver

For the 3D Rayleigh–Benard simulation data, we used a 3D finite element solver using Taylor–Hood (quadratic-linear) elements with quadratic elements for the temperature. This is solved using a penalty method, with penalty parameter  $\epsilon = 1 \times 10^{-4}/\mu$ . A second-order Crank–Nicolson time integration scheme is used for time discretization.

#### 4.2. Boussinesq equation MES-based POD ROM stabilization

*Case one:* we consider the Rayleigh–Bénard differential-heated cavity problem, modeled with the 3D Boussinesq Eqs. (31)–(33) with the following parameters and boundary conditions. The unit cube was discretized with 495k quadratic tetrahedral elements with 611k nodes leading to 1.83M velocity degrees of freedom and 611k temperature degrees of freedom. Thus,  $n \approx 2.4 \times 10^6$ . The velocity was taken as zero on the boundary and the temperature was specified at  $\pm 0.5$  on the x-faces and taken as homogeneous Neumann on the remaining faces. The non-dimensional parameters were taken as  $Re = 4.964 \times 10^4$ ,  $Pr = 712 \times 10^{-3}$ , and  $Gr = 7.369 \times 10^7$ , reasonable values in a quiet room. The simulation was run from zero velocity and temperature and snapshots were collected to  $t_f = 78$  [s] seconds.

We apply the results of Theorem 1 and Lemma 1 to this problem. In this case we use 8 POD basis functions for each variable, for the POD model (POD-ROM-G). The upper bounds on the uncertainties in the matrix and tensor entries are assumed to be  $c_{\max} = 10$ . The two closure model amplitudes  $\hat{\boldsymbol{\mu}} = [\mu_e, \mu_{nl}]^*$  are tuned using the discrete version of the MES algorithm (25), given by

$$\begin{aligned} y_1(k+1) &= y_1(k) + a_1 \Delta t \sin\left(\omega_1 k \Delta t + \frac{\pi}{2}\right) Q(\hat{\boldsymbol{\mu}}), \\ \hat{\mu}_e(k+1) &= y_1(k+1) + a_1 \sin\left(\omega_1 k \Delta t - \frac{\pi}{2}\right), \\ y_2(k+1) &= y_2(k) + a_2 \Delta t \sin\left(\omega_2 k \Delta t + \frac{\pi}{2}\right) Q(\hat{\boldsymbol{\mu}}), \\ \hat{\mu}_{nl}(k+1) &= y_2(k+1) + a_2 \sin\left(\omega_2 k \Delta t - \frac{\pi}{2}\right), \end{aligned} \quad (38)$$

where  $y_1(0) = y_2(0) = 0$ ,  $k = 0, 1, 2, \dots$  is the number of learning iterations, and  $\Delta t$  is the time increment. We use MES parameter values:  $a_1 = 0.08$  [–],  $\omega_1 = 10$  [ $\frac{\text{rad}}{\text{s}}$ ],  $a_2 = 10^{-7}$  [–],  $\omega_2 = 50$  [ $\frac{\text{rad}}{\text{s}}$ ]. The learning cost function is chosen as

$$Q(\boldsymbol{\mu}) = \int_0^{t_f} \langle e_T, e_T \rangle_{\mathcal{H}} dt + \int_0^{t_f} \langle e_v, e_v \rangle_{(\mathcal{H})^3} dt. \quad (39)$$

Moreover,  $e_T = P_T T_n - T_n^{pod}$ ,  $e_v = P_T \mathbf{v}_n - \mathbf{v}_n^{pod}$  define the errors between the projection of the true model solution onto the POD space  $\mathcal{Z}^T$  and the POD-ROM solution for temperature and velocity, respectively.

We first report in Fig. 1 the true velocity and temperature solutions. Fig. 2 show the solutions obtained at  $t = 50$  [s] with the nominal Galerkin ROM, with no closure model. If we examine the two figures closely, we can for instance notice that the nominal ROM-G velocity solutions, shown in Fig. 2 (left), exhibits a red area at the bottom-right of the volume-clip which is at a magnitude level of 0.07, this red area does not appear as pronounced on the clip of the true solution shown in Fig. 1 (left), where this area is at a magnitude level of 0.06. Since it is hard to compare the two solutions visually, we also report the errors between the true solutions and the POD-ROM-G solutions in Figs. 3 (left) and 4 (left). These errors will be compared later to the errors obtained with the learning-based ROM-G model.

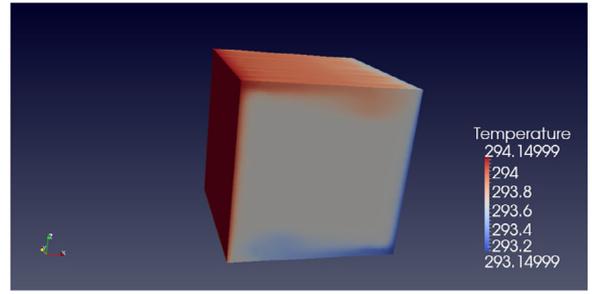
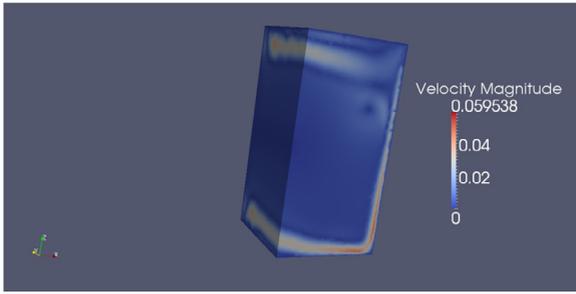


Fig. 1. True solution profile (left-side: velocity, right-side: temperature) – Case one.

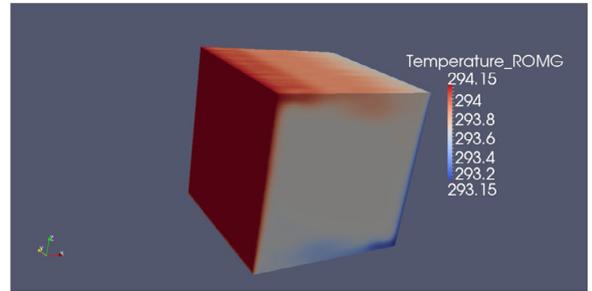
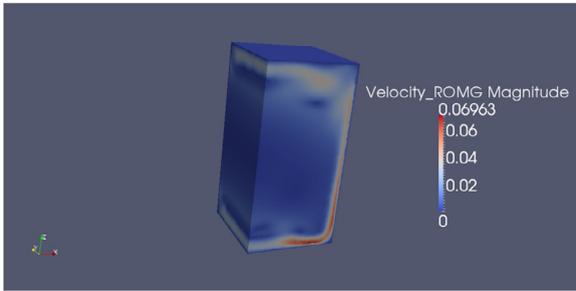


Fig. 2. ROM-G solution profile (left-side: velocity, right-side: temperature) – Case one.

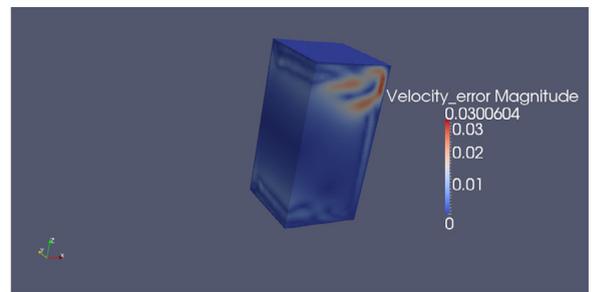
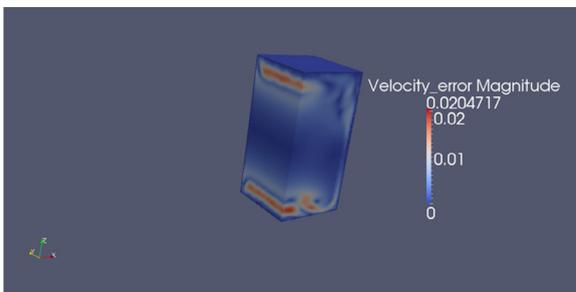


Fig. 3. ROM-G velocity errors profile (left-side: ROM-G, right-side: ROM-G-Learning) – Case one.

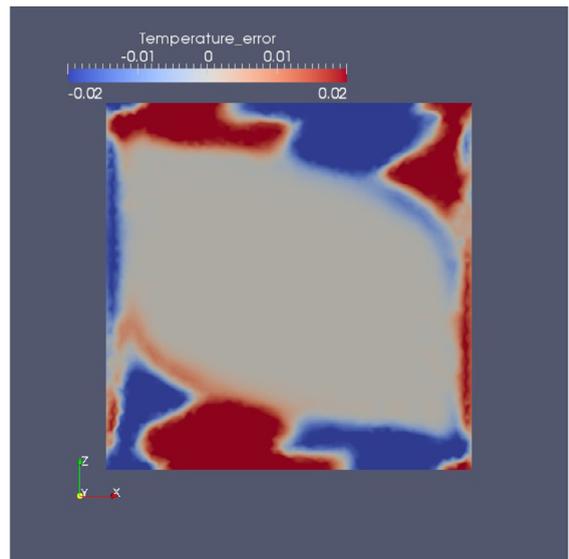
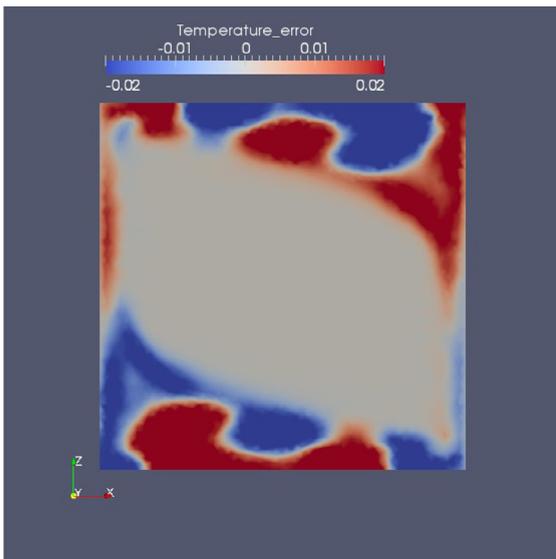


Fig. 4. ROM-G temperature errors profile (left-side: ROM-G, right-side: ROM-G-Learning) – Case one.



Fig. 5. ROM-G-Learning solution profile (left-side: velocity, right-side: temperature) – Case one.

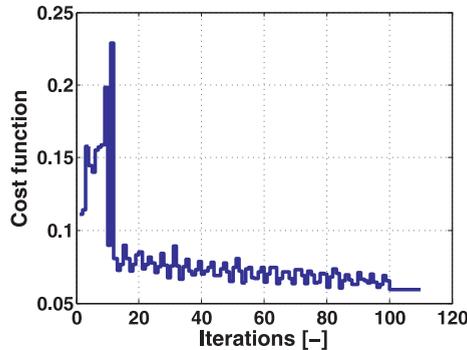


Fig. 6. Learning cost function vs. number of learning iterations – Case one.

For better evaluation of the learning performance w.r.t. the model correction, we show the profile of the learning cost function over the learning iterations in Fig. 6. We can see a quick decrease of the cost function within the first 20 iterations. This means that the MES manages to improve the overall solutions of the POD-ROM very quickly. We decide to stop the learning after 100 iterations, since the cost function has clearly decreased by then. Furthermore, we wanted to fairly evaluate the effect of the learning on all the examples, by stopping the learning when the iterations reach the same threshold. The parameters reach the minimizing values of  $\hat{\mu}_e \simeq 8.5 \times 10^{-1}$ , and  $\hat{\mu}_{nl} \simeq 1.25 \times 10^{-6}$ . The solutions due to the ROM-G with learning are reported in Fig. 5, where we can see that the velocity solution is closer to the true velocity, shown in Fig. 1 (left), where the red area is at a level of 0.06.

We also show the effect of the learning on the POD-ROM solutions by plotting the errors  $e_v$  and  $e_T$  in Figs. 3 (right), and 4 (right), which by comparison with Figs. 3 (left), and 4 (left), show an improvement of the POD-ROM solutions with the MES tuning of the closure model’s amplitudes.

Case two: in the second test, we consider a more challenging 3D free convection flow in a cube. Since there is no nominal velocity reference, the non-dimensionalization is performed using thermal parameters as in other free convection POD studies (cf. [30,33,34]):

$$x^* = \frac{x}{L}, \quad t^* = \frac{t\alpha}{L^2}, \quad \mathbf{v}^* = \frac{\mathbf{v}L}{\alpha}, \quad p^* = \frac{pL^2}{\rho\alpha^2}, \quad \text{and} \quad T^* = \frac{T - T_0}{T_w - T_0}.$$

The non-dimensionalized equations are

$$\begin{aligned} \frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} &= -\nabla p + \nabla \cdot \tau(\mathbf{v}) + \text{GrPr}^2 T \\ \nabla \cdot \mathbf{v} &= 0 \\ \frac{\partial T}{\partial t} + \mathbf{v} \cdot \nabla T &= \Delta T, \end{aligned}$$

where

$$\tau(\mathbf{v}) = \text{Pr}(\nabla \mathbf{v} + \nabla \mathbf{v}^T).$$

Note that the Rayleigh number is  $\text{Ra} = \text{GrPr}$ . For this test, we simulate the full-order model with Rayleigh number  $\text{Ra} = 5 \times 10^6$  and  $\text{Pr} = 1$ . The non-dimensionalized equations are solved on the unit cube with  $\mathbf{v} = \mathbf{0}$  on all boundaries,  $T = 1$  on the  $x = 0$  boundary,  $T = 0$  on the  $x = 1$  boundary and insulated ( $\nabla T \cdot \hat{n} = 0$ ) boundary conditions on the remaining surfaces. The simulation was performed with time steps of  $\Delta t = 1 \times 10^{-5}$  collecting snapshots every 10 time steps to  $t_f = 72 \times 10^{-4}$  [s].



Fig. 7. True solution profile (left-side: velocity, right-side: temperature) – Case two.

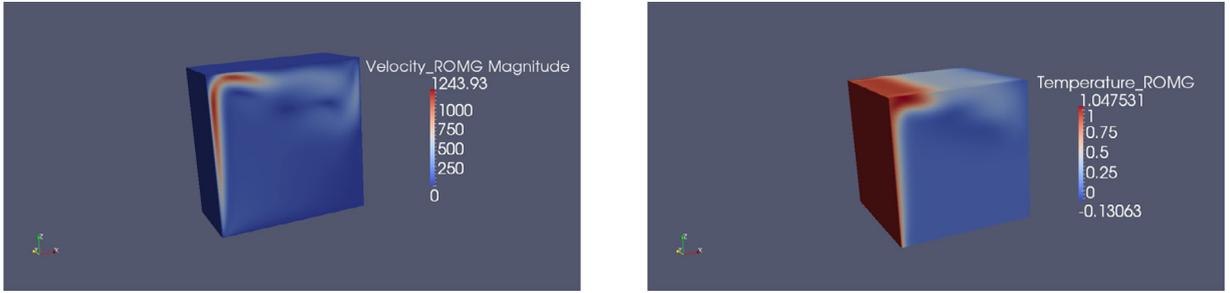


Fig. 8. ROM-G solution profile (left-side: velocity, right-side: temperature) – Case two.

We again apply the results of Theorem 1 and Lemma 1 to this problem. One difference with Case one, is that we use here two closure model coefficients  $\mu_{nl-T}$ , and  $\mu_{nl-V}$ , one for the temperature dynamics and one for the velocity dynamics. This gives us more degrees of freedom to improve the nominal ROM-G in this challenging case, where the flows are exhibit more nonlinear behavior compared to the first set of tests. We also make the model reduction task more challenging by using only 5 POD basis functions for each variable in the POD model (POD-ROM-G). The upper bounds on the uncertainties in the matrix and tensor entries are assumed to be  $c_{\max} = 10$ . The parameters  $\hat{\mu} = [\mu_e, \mu_{nl-T}, \mu_{nl-V}]^*$  are tuned using the discrete MES algorithm

$$\begin{aligned}
 y_1(k+1) &= y_1(k) + a_1 \Delta t \sin\left(\omega_1 k \Delta t + \frac{\pi}{2}\right) Q(\hat{\mu}), \\
 \hat{\mu}_e(k+1) &= y_1(k+1) + a_1 \sin\left(\omega_1 k \Delta t - \frac{\pi}{2}\right), \\
 y_2(k+1) &= y_2(k) + a_2 \Delta t \sin\left(\omega_2 k \Delta t + \frac{\pi}{2}\right) Q(\hat{\mu}), \\
 \hat{\mu}_{nl-T}(k+1) &= y_2(k+1) + a_2 \sin\left(\omega_2 k \Delta t - \frac{\pi}{2}\right), \\
 y_3(k+1) &= y_3(k) + a_3 \Delta t \sin\left(\omega_3 k \Delta t + \frac{\pi}{2}\right) Q(\hat{\mu}), \\
 \hat{\mu}_{nl-V}(k+1) &= y_3(k+1) + a_3 \sin\left(\omega_3 k \Delta t - \frac{\pi}{2}\right),
 \end{aligned} \tag{40}$$

where  $y_1(0) = y_2(0) = 0$ ,  $k = 0, 1, 2, \dots$  is the number of learning iterations, and  $\Delta t$  is the time increment. We use the following coefficients:  $a_1 = 0.08 [-]$ ,  $\omega_1 = 10 [\frac{\text{rad}}{\text{s}}]$ ,  $a_2 = 10^{-7} [-]$ ,  $\omega_2 = 50 [\frac{\text{rad}}{\text{s}}]$ ,  $a_3 = 8 \times 10^{-7} [-]$ ,  $\omega_3 = 12 [\frac{\text{rad}}{\text{s}}]$ . We use the same learning cost function (39). As a reference, we first report in Fig. 7 a snapshot of the exact solutions at  $t = 3.9 \times 10^{-3}$  [s]. One can observe in Fig. 7 (left), that the velocity amplitudes are much higher than in previous case. Next, we show the solution profiles at the same snapshot time, obtained by using the nominal POD ROM-G (without closure model), in Fig. 8. We can see a difference between the true velocity which has a maximum amplitude of 1020, and the ROM-G velocity with a maximum amplitude of 1244. The temperature solution also shows a slight difference in amplitude's range and color's intensity. To better see the difference between the true solutions and the ROM-G solutions, we report the corresponding error plots in Fig. 9, which will be later compared to the error plots of the learning-based ROM-G.

We then add the closure model to the nominal ROM-G and tune its parameters using the MES algorithm. The learning-based ROM-G solutions are reported in Fig. 10, and the corresponding errors are shown in Fig. 11. Where we can see that the range of the velocity and temperature ROM-G-learning solutions are closer to the true values than the ROM-G solutions. This is even clearer on the error plots, where the maximum magnitude of velocity error is about 235.5 comparatively to 271.36 for the ROM-G, and an excursion of the temperature error magnitude of 0.35, comparatively to 0.6 for the ROM-G.

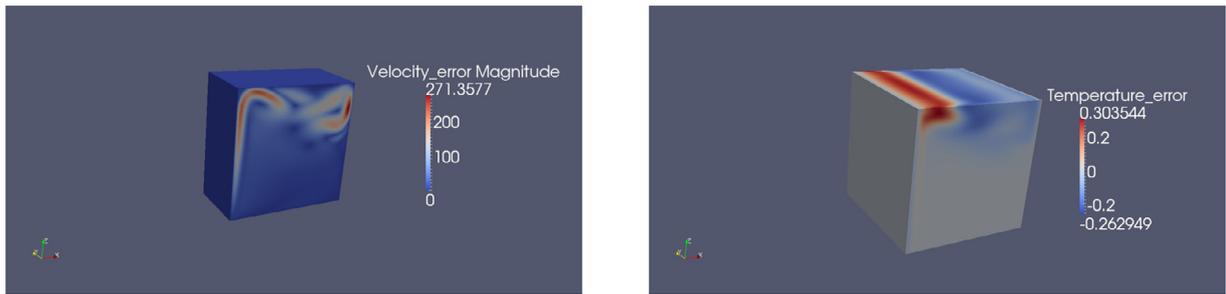


Fig. 9. ROM-G errors profile (left-side: velocity error, right-side: temperature error) – Case two.



Fig. 10. ROM-G-Learning solution profile (left-side: velocity, right-side: temperature) – Case two.

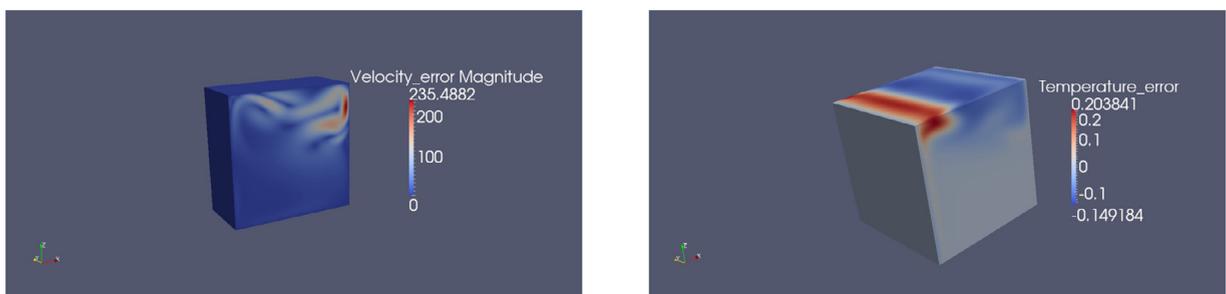


Fig. 11. ROM-G-Learning errors profile (left-side: velocity error, right-side: temperature error) – Case two.

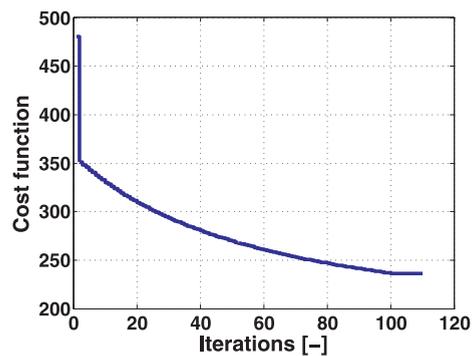


Fig. 12. Learning cost function vs. number of learning iterations – Case two.

Finally, the learning cost function is reported in Fig. 12, where we can see the convergence trend on the cost function decrease, after which, the parameters reach the minimizing values of  $\hat{\mu}_e \simeq 3 \times 10^{-1}$ ,  $\hat{\mu}_{nl-T} \simeq 3.6 \times 10^{-7}$ , and  $\hat{\mu}_{nl-V} \simeq 3 \times 10^{-6}$ . One can underline here that the observed convergence trend can be accelerated by improving the MES algorithm, for instance by choosing an algorithm with time-varying dither amplitudes  $a_i$ 's, however, it is not the goal of this paper to focus on the extremum seeking algorithm, we invite the interested reader to see e.g., [29], and references therein for more details on the MES tuning for performance improvement.

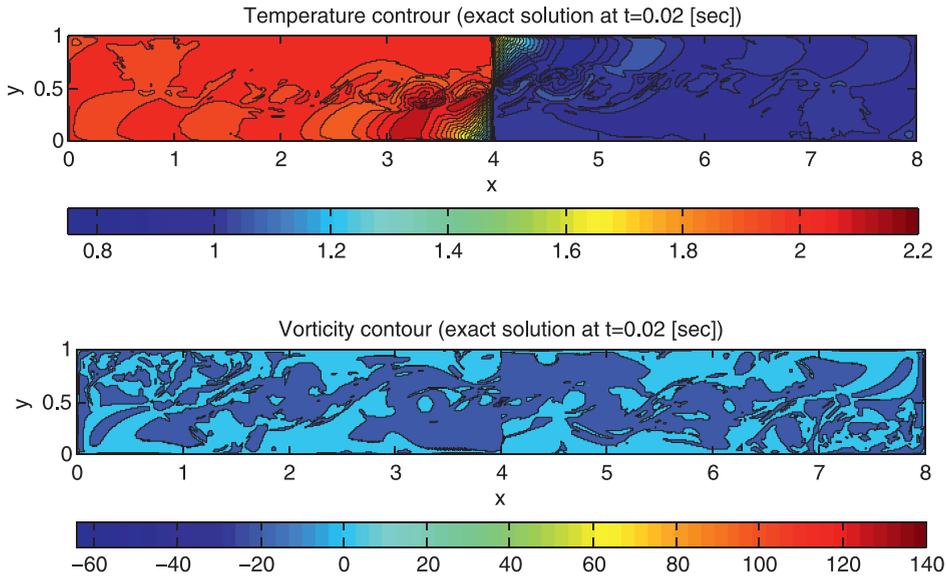


Fig. 13. True solution profile snapshot at 0.02 [s] (top: temperature, bottom: vorticity) – Case three.

Case three: we finally consider the case of 2D incompressible Boussinesq equation modeling the unsteady lock-exchange flow problem. This case exhibits a strong shear flow induced by temperature jump which results in the Kelvin–Helmholtz instability [31]. Instead of the original formulation in the velocity and temperature variables, here we follow [31] and rewrite the Boussinesq equation in terms of vorticity  $\omega = (\nabla \times \nu) \cdot \mathbf{k}$ , stream function  $\psi$ , and temperature  $T$ . In this case, the equations can be written as the two coupled scalar transport equations

$$\frac{\partial \omega}{\partial t} + \frac{\partial \psi}{\partial y} \frac{\partial \omega}{\partial x} - \frac{\partial \psi}{\partial x} \frac{\partial \omega}{\partial y} = \frac{1}{\text{Re}} \Delta \omega + \frac{\text{Gr}}{\text{Re}^2} \frac{\partial T}{\partial x}, \tag{41}$$

$$\frac{\partial T}{\partial t} + \frac{\partial \psi}{\partial y} \frac{\partial T}{\partial x} - \frac{\partial \psi}{\partial x} \frac{\partial T}{\partial y} = \frac{1}{\text{RePr}} \Delta T, \tag{42}$$

along with the kinematic relationship between vorticity and stream function given by

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = -\omega. \tag{43}$$

The desired flow velocity components can be recovered from

$$u = \frac{\partial \psi}{\partial y}, \quad v = -\frac{\partial \psi}{\partial x}. \tag{44}$$

In this numerical tests, we use the following parameters:  $\text{Re} = 10^4$ ,  $\text{Gr} = 4 \times 10^8$ ,  $\text{Pr} = 1$ . The PDE is solved over the time interval  $[0, 8]$ , with a time discretization step  $\Delta t = 0.0005$ . The  $(x, y)$  space defined over  $[0, 8] \times [0, 1]$  has been discretized using a compact finite difference scheme as described in [31] grid of  $4096 \times 512$ . For the nominal (without closure model) POD model (POD–ROM–G), we use 10 POD basis functions for the vorticity and 10 POD basis functions for the temperature variables computed from snapshots taken every  $\Delta t = 2 \times 10^{-2}$  [s]. In Figs. 13 and 14, we present the direct numerical simulation (DNS) of (41) and (42). We cannot show all the snapshots of the solution, instead we choose to show the initial and final snapshots. The first snapshot, shown in Fig. 13, is taken at the initial instant  $t = 2 \times 10^{-2}$  [s] (to avoid the trivial snapshot corresponding to  $t = 0$  s showing the static boundary conditions), the second snapshot, shown in Fig. 14, corresponds to  $t = 8$  [s]. We see in Fig. 13 the setup of this experiment, where two fluids of different temperatures (or equivalently, different densities) are separated by a vertical barrier at  $x = 4$ . On the right side of the barrier we have low temperature (equivalently, high fluid density), whereas, on the left side of the barrier we have high temperature (equivalently, low fluid density). Based on this setup, when we remove the barrier between the two fluids, we expect the low density, warmer fluid to rise, while the high density, cooler fluid sinks, as seen in Fig. 14. Vortices are generated, then diffuse as these fluids slide past one another. Next, we show in Fig. 15, the solutions obtained by POD–ROM–G (without a closure model) at  $t = 8$  s. One observes that they are significantly different from the exact snapshot solution displayed in Fig. 14. The error is due to the truncation of the higher POD modes, since we used only the first 10 POD modes to build the ROM model. To see these error more quantitatively, the errors between the exact and the POD–ROM–G solutions are plotted in Fig. 16. By examining the legend in Fig. 16, one can see that the error in temperature appears to be mainly between  $-2.6$  and  $1.5$ . The error in vorticity shown in Fig. 16, varies between  $-100$  and  $20$ .

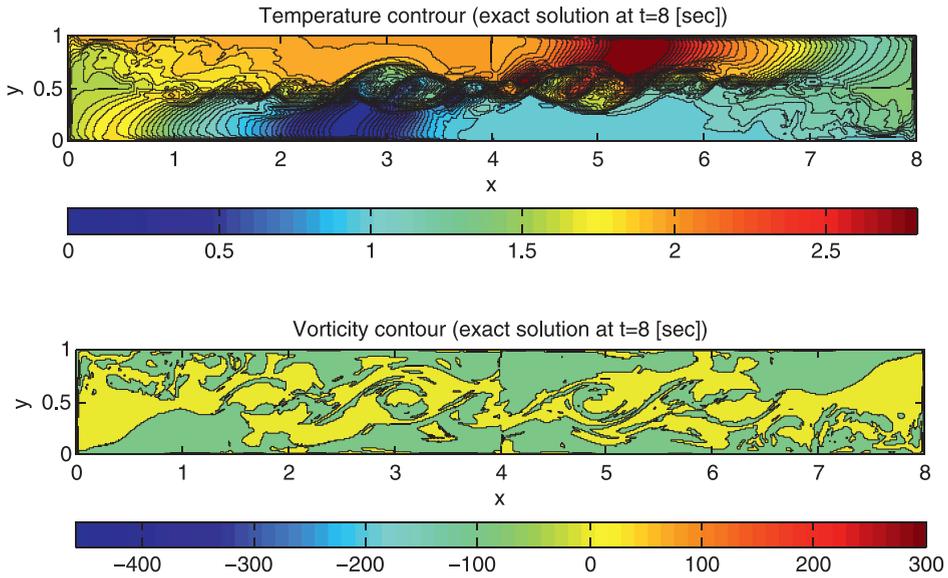


Fig. 14. True solution profile snapshot at 8 [s] (top: temperature, bottom: vorticity) – Case three.

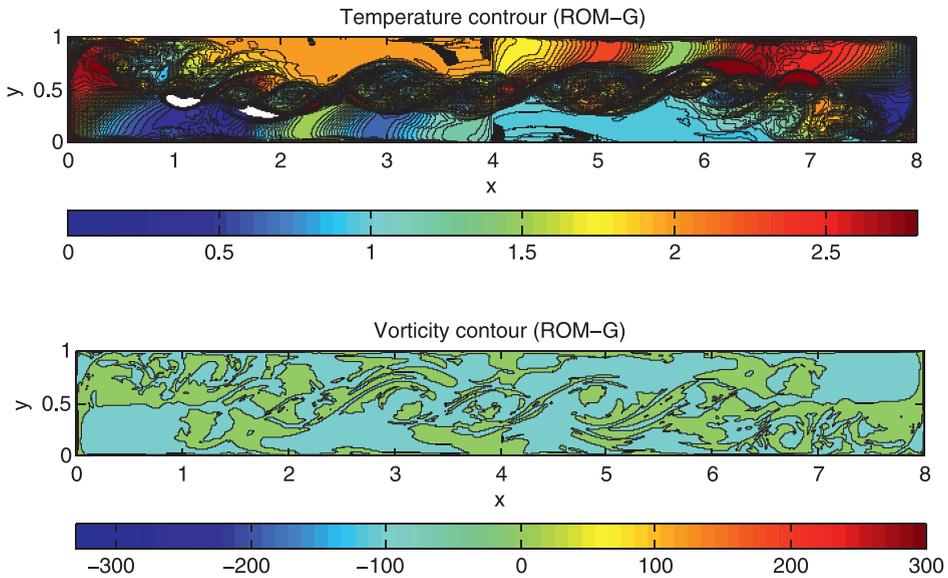


Fig. 15. POD-ROM-G (no closure) solution profile snapshot at 8 [s] (top: temperature, bottom: vorticity) – Case three.

To reduce these errors, we use the closure model proposed in Theorem 1, and complement it with the learning-based tuning of Lemma 1. Note, that here we have two closure model parameters to tune, the linear gain  $\mu_e$ , and the nonlinear gain  $\mu_{nl}$ . To tune these parameters we use similar learning cost function as in the two previous cases, and the extremum seeking algorithm, where we select:  $a_1 = 10^{-5}$  [-],  $\omega_1 = 15$  [ $\frac{\text{rad}}{\text{s}}$ ],  $a_2 = 10^{-12}$  [-],  $\omega_2 = 4$  [ $\frac{\text{rad}}{\text{s}}$ ]. We run the extremum seeking algorithm for 100 iterations. The corresponding cost function is shown in Fig. 17. We see that the cost function drops below 1000 (tenth of the initial cost) after only 14 iterations, however, we decided to continue the learning until 100 to be consistent with the other tests. The associated closure model optimal coefficients obtained are  $\hat{\mu}_e \simeq 5.4 \times 10^{-5}$ , and  $\hat{\mu}_{nl} \simeq 4 \times 10^{-8}$ . Notice that the obtained optimal values for the closure model coefficients are small, which indicates that in this challenging example, fine-tuning the closure model is important, and the use of extremum seekers is even more justified, especially considering the fact that they can be implemented in realtime to fine-tune the ROM, by comparing the ROM solutions to real measurements.

Although, it is clear from the plot of the cost function that the errors between the solutions of ROM-G-Learning (with tuned closure model) and the exact solutions decrease, we show, for the purpose of comparison, the solutions of the ROM-G-Learning at  $t = 8$  [s] in Fig. 18, as well as the associated errors compared with the full-order simulation, in Fig. 19. One

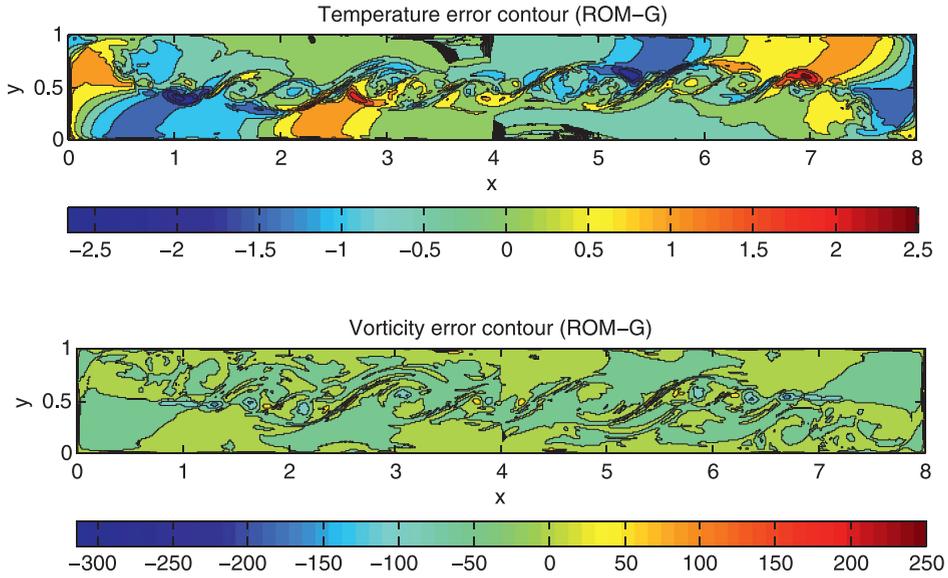


Fig. 16. POD-ROM-G (no closure) error profile snapshot at 8 [s] (top: temperature, bottom: vorticity) – Case three.

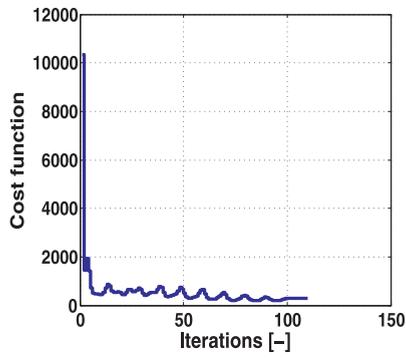


Fig. 17. Learning cost function vs. number of learning iterations – Case three.

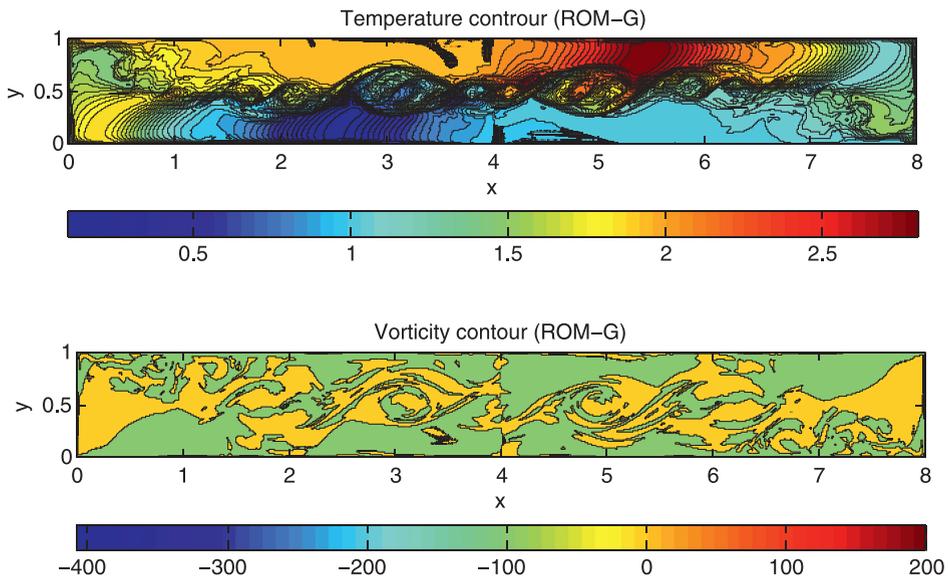


Fig. 18. ROM-G-Learning solution profile snapshot at 8 [s] (top: temperature, bottom: vorticity) – Case three.

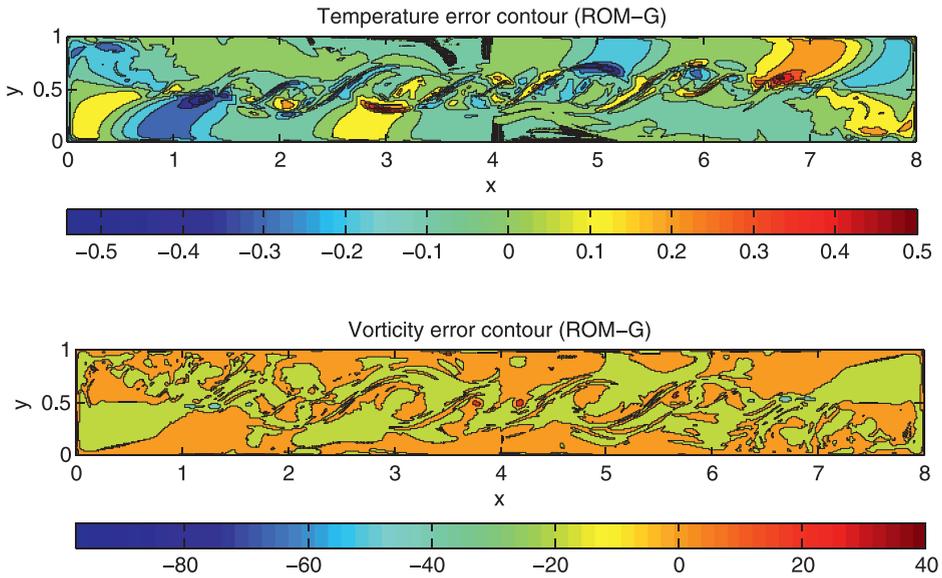


Fig. 19. ROM-G-Learning error profile snapshot at 8 [s] (top: temperature, bottom: vorticity) – Case three.

can see that the error in temperature mainly falls between  $-0.4$  and  $0.2$ , whereas the error in vorticity, varies between  $-30$  and  $0$ , which shows a clear improvement of the closure model quality.

**5. Conclusion**

In this work we have proposed a new closure model for ROMs that provide robust stabilization when applied to PDEs with parametric uncertainties. We have also proposed the use of a data-driven multi-parametric extremum seeking (MES) algorithm to auto-tune the closure model coefficients that optimize the POD-ROM solution predictions. One main feature of the proposed approach w.r.t. available stabilizing closure models is the fact that the closure model proposed here takes into account model uncertainties in its formulation. Moreover, the fact that we are using a data-driven optimization algorithm, motivated by the control theory of extremum seeking, allows us to continuously update our closure model, on-line, based on real-data measurements. We have validated the proposed method on a challenging 2D, and 3D Boussinesq test-case by considering the unsteady lock-exchange flow problem, and the Rayleigh–Bénard differentially heated cavity problem. The proposed closure model has shown encouraging performance in terms of improving solution precision in the laminar flow cases considered here.

Although the proposed method performs well in the case of simulated laminar flow, we still need to test it on more challenging flows and real-world experiments, therefore, future investigations will be conducted on more challenging turbulent flows, and on-line experimental tests using a water-tank test-bed. Another point that will be investigated in our future work is the comparison between the robust closure model with its on-line auto-tuning, and classical closure models (nonrobust–nonadaptive), to see how the robust closure model performs comparatively to classical closure models in real-world test-beds, where uncertainties and systems’ drifts over time, are unequivocal, and where robustness and adaptability can help improve the model performance.

**Appendix A**

Here we provide details on the derivation of Eq. (34). Recall the Boussinesq equations

$$\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} = -\nabla p + \nabla \cdot \tau(\mathbf{v}) + \frac{Gr}{Re^2} T \hat{\mathbf{e}}_3, \tag{31}$$

$$\nabla \cdot \mathbf{v} = 0, \tag{32}$$

$$\frac{\partial T}{\partial t} + \mathbf{v} \cdot \nabla T = \nabla \cdot \left( \frac{1}{RePr} \nabla T \right) \tag{33}$$

that are solved in flow domain  $\Omega$ . Let  $\mathbf{v}_0$  and  $T_0$  be steady-state solutions to (31)–(33), and let  $\{\phi_i^{\mathbf{v}}\}_{i=1}^{r_v}$  and  $\{\phi_i^T\}_{i=r_v+1}^{r_v+r_T}$  be POD basis functions that well represent differences between representative simulation data  $(\mathbf{v}_{sim}, T_{sim})$  and  $(\mathbf{v}_0, T_0)$ . Then

we introduce the POD approximations to  $\mathbf{v}$  and  $T$  as

$$\mathbf{v}^{pod}(x, t) = \mathbf{v}_0(x) + \sum_{j=1}^{r_v} q_j(t) \phi_j^{\mathbf{v}}(x), \tag{35}$$

$$T^{pod}(x, t) = T_0(x) + \sum_{j=r_v+1}^{r_v+r_T} q_j(t) \phi_j^T(x). \tag{36}$$

These two equations serve as the reduced-order model for solutions to (31)–(33) once we derive a dynamical system for the coefficients  $\{q_i(t)\}_{i=1}^{r_v+r_T}$  below. First of all, note that the POD procedure constructs orthonormal basis functions for the velocity and basis functions can be written as

$$\phi_i^{\mathbf{v}}(x) = \frac{1}{t_f} \int_0^{t_f} (\mathbf{v}_{sim}(x, t) - \mathbf{v}_0(x)) w_i(t) dt,$$

where the weight functions  $w_i(t)$  are orthogonal, i.e.  $\int_0^{t_f} w_i(t) w_j(t) dt = 0$ , if  $i \neq j$ . To summarize, POD basis functions accurately represent the discrepancy between our solution and a given steady-state solution and satisfy the orthogonality conditions

$$\int_{\Omega} \phi_i^{\mathbf{v}}(x) \phi_j^{\mathbf{v}}(x) dx = \delta_{ij} \quad \text{and} \quad \int_{\Omega} \phi_i^T(x) \phi_j^T(x) dx = \delta_{ij}, \tag{45}$$

where  $\delta_{ij}$  is 1 if  $i = j$  and 0 otherwise. Since our simulation data  $\mathbf{v}_{sim}$  and steady-state solution  $\mathbf{v}_0$  each satisfy (32), we know that  $\nabla \cdot \phi_i^{\mathbf{v}}(x) = 0$  for each  $i = 1, \dots, r_v$ . Therefore, (32) is trivially satisfied with our approximation (35).

To perform Galerkin projection, we take the dot product of (31) with  $\phi_i^{\mathbf{v}}$  and multiply (33) by  $\phi_i^T$ , integrate these equations over the fluid domain  $\Omega$ , and perform integration by parts on the terms involving  $-\nabla p$ ,  $\nabla \cdot \tau(\mathbf{v})$  and  $\nabla \cdot (\frac{1}{\text{RePr}} \nabla T)$ . Beginning with (31), we have

$$\int_{\Omega} \frac{\partial \mathbf{v}}{\partial t} \cdot \phi_i^{\mathbf{v}} dx = - \int_{\Omega} (\mathbf{v} \cdot \nabla) \mathbf{v} \cdot \phi_i^{\mathbf{v}} + \tau(\mathbf{v}) : \epsilon(\phi_i^{\mathbf{v}}) - \frac{\text{Gr}}{\text{Re}^2} T \hat{\mathbf{e}}_3 \cdot \phi_i^{\mathbf{v}} dx, \tag{46}$$

for  $i = 1, \dots, r_v$ , where we have introduced  $\epsilon_{ij}(\mathbf{v}) = \frac{1}{2} \left( \frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right)$ ,  $\tau(\mathbf{v}) = \frac{2}{\text{Re}} \epsilon(\mathbf{v})$ , and used the fact that  $\phi_i^{\mathbf{v}}$  is divergence-free to eliminate the pressure term. There may be boundary integral terms that arise in the integration by parts. In our examples, we consider problems where the  $\phi_i^{\mathbf{v}}$  functions are either zero on the boundary (an important motivation for subtracting  $\mathbf{v}_0$ ) or have zero stress (a linear property that the basis inherits from the zero stress boundary conditions on  $\mathbf{v}$  and  $\mathbf{v}_0$ ). In fact, our numerical examples have zero velocity boundary conditions and we take  $\mathbf{v}_0 = \mathbf{0}$ . However, to maintain generality, we consider non-trivial steady-state solutions but ignore the boundary integral terms.

Performing the calculation with (33), we find

$$\int_{\Omega} \frac{\partial T}{\partial t} \phi_i^T dx = - \int_{\Omega} (\mathbf{v} \cdot \nabla) T \phi_i^T + \left( \frac{1}{\text{RePr}} \nabla T \right) \cdot \nabla \phi_i^T dx, \tag{47}$$

for  $i = r_v + 1, \dots, r_v + r_T$ . Now we replace the  $\mathbf{v}$  and  $T$  in (46) and (47) with our expressions for  $\mathbf{v}^{pod}$  and  $T^{pod}$  in (35) and (36), respectively. Using the orthogonality properties (45), Eq. (46) becomes

$$\dot{q}_i(t) = b_i + \frac{1}{\text{Re}} \sum_{j=1}^{r_v} D_{ij} q_j(t) + \sum_{j=1}^{r_v} \sum_{k=1}^{r_v} C_{ijk} q_k(t) q_j(t) \tag{48}$$

where

$$b_i = - \int_{\Omega} (\mathbf{v}_0(x) \cdot \nabla \mathbf{v}_0(x)) \cdot \phi_i^{\mathbf{v}}(x) + \tau(\mathbf{v}_0(x)) : \epsilon(\phi_i^{\mathbf{v}}(x)) - \frac{\text{Gr}}{\text{Re}^2} T_0(x) \hat{\mathbf{e}}_3 \cdot \phi_i^{\mathbf{v}}(x) dx,$$

$$D_{ij} = - \int_{\Omega} \text{Re}(\mathbf{v}_0(x) \cdot \nabla \phi_j^{\mathbf{v}}(x)) \cdot \phi_i^{\mathbf{v}}(x) + \nabla \mathbf{v}_0(x) \phi_j^{\mathbf{v}}(x) \cdot \phi_i^{\mathbf{v}}(x) + 2\epsilon(\phi_j^{\mathbf{v}}(x)) : \epsilon(\phi_i^{\mathbf{v}}(x))$$

for  $j = 1, \dots, r_v$ , and

$$D_{ij} = - \frac{\text{Gr}}{\text{Re}} \int_{\Omega} \phi_j^T(x) \hat{\mathbf{e}}_3 \cdot \phi_i^{\mathbf{v}}(x) dx$$

for  $j = r_v + 1, \dots, r_v + r_T$ , and

$$C_{ijk} = - \int_{\Omega} (\phi_j^{\mathbf{v}}(x) \cdot \nabla \phi_k^{\mathbf{v}}(x)) \cdot \phi_i^{\mathbf{v}}(x) dx,$$

for  $i = 1, \dots, r_v$ .

Performing the substitution for (47) and again using (45) leads to

$$\dot{q}_i(t) = b_i + \frac{1}{\text{Re}} \sum_{j=1}^{r_T} D_{ij} q_j(t) + \sum_{j=r_V+1}^{r_V+r_T} \sum_{k=1}^{r_V} C_{ijk} q_k(t) q_j(t) \quad (49)$$

where

$$b_i = - \int_{\Omega} (\mathbf{v}_0(x) \cdot \nabla T_0(x)) \phi_i^T(x) dx + \frac{1}{\text{RePr}} \nabla T_0(x) \cdot \nabla \phi_i^T(x) dx,$$

$$D_{ij} = - \int_{\Omega} \text{Re}(\phi_j^V \cdot \nabla T_0(x)) \phi_i^T(x) dx$$

for  $j = 1, \dots, r_V$ , and

$$D_{ij} = - \int_{\Omega} \text{Re}(\mathbf{v}_0(x) \cdot \nabla \phi_j^T(x)) \phi_i^T(x) + \frac{1}{\text{Pr}} \nabla \phi_j^T(x) \cdot \nabla \phi_i^T(x) dx$$

for  $j = r_V + 1, \dots, r_V + r_T$ , and

$$C_{ijk} = - \int_{\Omega} (\phi_k^V(x) \cdot \nabla \phi_j^T(x)) \phi_i^T(x) dx$$

for  $j = r_V + 1, \dots, r_V + r_T$ ,  $k = 1, \dots, r_V$ , and  $i = r_V + 1, \dots, r_V + r_T$ .

Combining (48) and (49) leads to a dynamical system for the POD coefficients of the form

$$\dot{q}(t) = b + \frac{1}{\text{Re}} Dq(t) + [Cq(t)]q(t).$$

Initial conditions for the model can be obtained by projecting the difference between the initial conditions for (31)–(33) and the steady-state solutions  $\mathbf{v}_0$  and  $T_0$  onto the POD basis

$$q_i(0) = \int_{\Omega} (\mathbf{v}(x, 0) - \mathbf{v}_0(x)) \cdot \phi_i^V(x) dx, \quad i = 1, \dots, r_V \quad (50)$$

$$q_i(0) = \int_{\Omega} (T(x, 0) - T_0(x)) \phi_i^T(x) dx, \quad i = r_V + 1, \dots, r_V + r_T. \quad (51)$$

Finally, for cases where we can choose trivial values of  $\mathbf{v}_0$  and  $T_0$ , the right hand side simplifies considerably. The  $b$  term vanishes and  $D$  simply contains diagonal blocks of  $-2 \int_{\Omega} \epsilon(\phi_j^V) : \epsilon(\phi_i^V) dx$  and  $-\int_{\Omega} \nabla \phi_j^T \cdot \nabla \phi_i^T / \text{Pr} dx$  and the off-diagonal buoyancy term.

## References

- [1] P. Holmes, J.L. Lumley, G. Berkooz, *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*, Cambridge University Press, 1998.
- [2] M. Couplet, C. Basdevant, P. Sagaut, Calibrated reduced-order POD-Galerkin system for fluid flow modelling, *J. Comput. Phys.* 207 (1) (2005) 192–220.
- [3] V.L. Kalb, A.E. Deane, An intrinsic stabilization scheme for proper orthogonal decomposition based low-dimensional models, *Phys. Fluids* 19 (5) (2007) 054106.
- [4] T. Bui-Thanh, K. Willcox, O. Ghattas, B. van Bloemen Waanders, Goal-oriented, model-constrained optimization for reduction of large-scale systems, *J. Comput. Phys.* 224 (2) (2007) 880–896.
- [5] M. Ilak, S. Bagheri, L. Brandt, C.W. Rowley, D.S. Henningson, Model reduction of the nonlinear complex Ginzburg–Landau equation, *SIAM J. Appl. Dyn. Syst.* 9 (4) (2010) 1284–1302.
- [6] I. Kalashnikova, B. van Bloemen Waanders, S. Arunajatesan, M. Barone, Stabilization of projection-based reduced order models for linear time-invariant systems via optimization-based eigenvalue reassignment, *Comput. Methods Appl. Mech. Eng.* 272 (2014) 251–270.
- [7] Z. Wang, I. Akhtar, J. Borggaard, T. Iliescu, Proper orthogonal decomposition closure models for turbulent flows: a numerical comparison, *Comput. Methods Appl. Mech. Eng.* 237–240 (2012) 10–26.
- [8] O. San, T. Iliescu, Proper orthogonal decomposition closure models for fluid flows: Burgers equation, *Int. J. Numer. Anal. Model.* 1 (1) (2013) 1–18.
- [9] O. San, J. Borggaard, Basis selection and closure for POD models of convection dominated Boussinesq flows, in: *Proceeding of the Twenty-first International Symposium on Mathematical Theory of Networks and Systems*, Groningen, The Netherlands, 2014, pp. 132–139.
- [10] M. Benosman, B. Kramer, P.T. Boufounos, P. Grover, Learning-based reduced order model stabilization for partial differential equations: application to the coupled Burgers' equation, in: *Proceeding of the 2016 American Control Conference*, 2016, pp. 1673–1678.
- [11] D. Kim, J. Braun, E.M. Cliff, J. Borggaard, Development, validation and application of a coupled reduced-order CFD model for building control applications, *Build. Environ.* 93 (2015) 97–111.
- [12] L. Cordier, B. Noack, G. Tissot, G. Lehnasch, J. Delville, M. Balajewicz, G. Daviller, R.K. Niven, Identification strategies for model-based control, *Exp. Fluids* 54 (1580) (2013) 1–21.
- [13] M. Balajewicz, E. Dowell, B. Noack, Low-dimensional modelling of high-Reynolds-number shear flows incorporating constraints from the Navier–Stokes equation, *J. Fluid Mech.* 729 (1) (2013) 285–308.
- [14] M. Balajewicz, Lyapunov stable Galerkin models of post-transient incompressible flows, 2013. arXiv: 1312.0284
- [15] M. Benosman, J. Borggaard, B. Kramer, in: Robust reduced-order model stabilization for partial differential equations based on Lyapunov theory and extremum seeking with application to the 3D Boussinesq equations, 2016. arXiv: 1604.04586
- [16] W. Haddad, V.S. Chellaboina, *Nonlinear Dynamical Systems and Control: A Lyapunov-based Approach*, Princeton University Press, 2008.
- [17] M. Gunzburger, *Finite Element Methods for Viscous Incompressible Flows*, Academic Press, 1989.
- [18] K. Kunisch, S. Volkwein, Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics, *SIAM J. Numer. Anal.* 40 (2) (2007) 492–515.
- [19] B. Kramer, P. Grover, P. Boufounos, M. Benosman, S. Nabi, in: Sparse sensing and DMD based identification of flow regimes and bifurcations in complex flows, 2015. arXiv: 1510.02831

- [20] K. Veroy, A. Patera, Certified real-time solution of the parametrized steady incompressible Navier–Stokes equations: rigorous reduced-basis a posteriori error bounds, *Int. J. Numer. Methods Fluids* 47 (8) (2005) 773–788.
- [21] D. Rempfer, Kohärente Strukturen und Chaos beim Laminar-turbulenten Grenzschichtumschlag, University of Stuttgart, 1991. Master's thesis, Ph.D.
- [22] E. Tadmor, Convergence of spectral methods for nonlinear conservation laws, *SIAM J. Numer. Anal.* 26 (1) (1989) 30–44.
- [23] S. Sirisup, G.E. Karniadakis, A spectral viscosity method for correcting the long-term behavior of pod models, *J. Comput. Phys.* 194 (1) (2004) 92–116.
- [24] J.P. Chollet, Two-point closure used for a sub-grid scale model in large eddy simulations, in: *Turbulent Shear Flows 4*, Springer, 1984, pp. 62–72.
- [25] M. Lesieur, O. Metais, New trends in large-eddy simulations of turbulence, *Ann. Rev. Fluid Mech.* 28 (1) (1996) 45–82.
- [26] G.S. Karamanos, G.E. Karniadakis, A spectral vanishing viscosity method for large-eddy simulations, *J. Comput. Phys.* 163 (1) (2000) 22–50.
- [27] M.A. Rotea, Analysis of multivariable extremum seeking algorithms, in: *Proceeding of the 2000 American Control Conference*, 2000, pp. 433–437.
- [28] D. Liberzon, Switching in systems and control, in: *Systems and Control: Foundations and Applications*, Birkhauser, 2006. Volume in Series
- [29] M. Benosman, *Learning-Based Adaptive Control: An Extremum Seeking Approach – Theory and Applications*, Butterworth-Heinemann, 2016. Elsevier Ed.
- [30] B. Podvin, A. Sergent, Proper orthogonal decomposition investigation of turbulent Rayleigh–Bénard convection in a rectangular cavity, *Phys. Fluids* 24 (105106) (2012).
- [31] O. San, J. Borggaard, Principal interval decomposition framework for POD reduced-order modeling of convection Boussinesq flows, *Int. J. Numer. Methods Fluids* 78 (2015) 37–62.
- [32] S. Lele, Compact finite difference schemes with spectral-like resolution, *J. Comput. Phys.* 103 (1) (1992) 16–42.
- [33] J. Bailon-Cuba, J. Schumacher, Low-dimensional model of turbulent Rayleigh–Bénard convection in a Cartesian cell with square domain, *Phys. Fluids* 23 (077101) (2011).
- [34] A.E. Deane, L. Sirovich, A computational study of Rayleigh–Bénard convection. Part 1. Rayleigh-number scaling, *J. Fluid Mech.* 222 (1991) 231–250.
- [35] S. Gottlieb, C.-W. Shu, Total variation diminishing RungeKutta schemes, *Mathematics of Computation* 67 (221) (1998) 73–85.
- [36] Y. Wang, J. Zhang, Sixth order compact scheme combined with multigrid method and extrapolation technique for 2D Poisson equation, *Journal of Computational Physics* 228 (1) (2009) 137–146.