Performance comparison of data-driven reduced models for a single-injector combustion process

Parikshit Jain* and Boris Kramer†
University of California San Diego, CA, 92093

Shane A. McQuarrie‡
University of Texas at Austin, TX, 78712

This paper derives a new data-driven reduced model for a challenging large-scale combustion simulation and offers a detailed performance comparison with two state-of-the-art reduced models that were previously developed for this application. In particular, we learn a physics-based cubic reduced-order model (ROM) via the operator inference framework (OPINF). The key to the efficiency and physics-based nature of the model lies in the use of variable transformations of the original highly nonlinear system that make the governing equations polynomial in the new variables. We compare this approach with a quadratic operator inference ROM and with dynamic mode decomposition with control (DMDc), a modal decomposition method that has been shown to work well in many fluid dynamical applications. An extensive comparison of these approaches yields interesting insights: in particular, we find that the cubic and quadratic models based on operator inference capture more high-frequency content of the system compared to DMDc; we also find that the presence of the cubic term in the operator inference ROM improves stability properties of the system. Each of the reduced models we examine are accurate and provide a significant computational speedup compared to the high-fidelity model.

I. Introduction

Low-dimensional, computationally efficient dynamical systems models are a key enabler of uncertainty quantification, design, and control in combustion applications. The past several years have seen a spike in interest in data-driven approaches to low-dimensional modeling, including physics-based methods, machine learning techniques, and hybrid procedures combining elements of each. These advances are made possible by the increased availability of data—both physically observed and numerically simulated—in all areas of aerospace engineering, as well as greater processing power for extracting relevant information from such vast amounts of data. Combustion applications are no exception to this trend, but low-dimensional modeling of combustion processes remains challenging due to their nonlinear, multiscale, and highly advective nature. Nevertheless, several recent innovations in data-driven approaches have produced encouraging results in combustion applications [1–4]. The subject of this work is to develop additional computational tools for these combustion applications and to provide new analysis from a spectral decomposition perspective.

There are a variety of modeling techniques for learning low-dimensional systems from data, each with their own strengths and weaknesses. For systems where the structure is assumed to be linear, the Loewner framework [5–8], the eigensystem realization algorithm [9–12], and vector fitting [13, 14] have all been shown to produce accurate and fast simulation results. These methods are largely well established and applicable to different system domains, data generation schemes, and so forth. For nonlinear systems, there are many low-dimensional model learning approaches based on machine learning concepts for learning general nonlinear functions (see, e.g., [15] in the context of combustion) or for representing the dynamical state in a nonlinear manner [16]. Moreover, nonlinear system identification methods such as [17, 18] can learn governing equations from data, provided a proper library of basis functions that express the data is found. Many learning approaches assume a specific form of the nonlinear terms, such as the Loewner approach for quadratic-bilinear systems [19]. Additional consideration might be given to the nonlinear terms to enforce efficient models, such as by using the discrete empirical interpolation method (DEIM) [20, 21]; see also [1, 22] for a DEIM-based method applied to the combustion application considered herein.

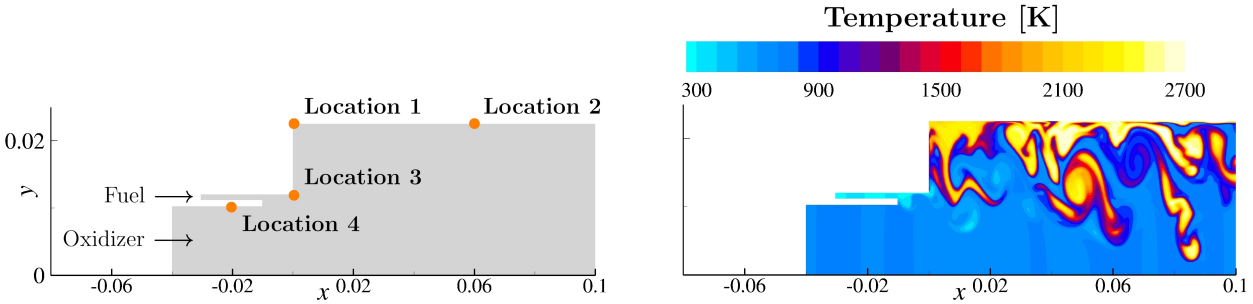
*Graduate Student, Department of Mechanical and Aerospace Engineering
†Assistant Professor, Department of Mechanical and Aerospace Engineering, Senior Member AIAA
‡Graduate Student, Oden Institute for Computational Engineering and Sciences

In this paper, we focus on two highly transparent data-driven methods for constructing low-dimensional models of nonlinear systems. First, dynamic mode decomposition (DMD) is a modal decomposition technique related to Koopman operator theory [23] that learns linear reduced models for nonlinear dynamical systems. The performance of DMD can be greatly improved by including knowledge of suitable observables that make the input-output map low-dimensional [24–26]. In this study, we make use of dynamic mode decomposition with control (DMDc) [27], an extension of DMD for forced systems [28, 29]. Second, the operator inference (OPINF) approach introduced in [30] learns low-dimensional nonlinear operators that are polynomial in the state and can have external forcing. The framework has been extended to general nonlinear systems by utilizing variable transformations [2, 3, 31, 32] and can also work in hybrid situations where part of the model is known in analytic form but other parts have to be learned [33]. Building on the successful application of quadratic operator inference for combustion, we present a data-driven cubic reduced-order model (ROM) for a high-dimensional discretization of a single-injector combustion application and compare it to the quadratic operator inference ROM presented in [2, 3] to better understand the effect of including higher-order polynomial terms. In addition, we construct a ROM based on DMDc where we use the original DMD model with an added control term in the equation. Our numerical experiments compare the ROMs and their performance both in the time domain and in the spectral domain, offering insight into the replication of flow frequencies and phase changes in the ROM. To our knowledge, this is the first such comparison of these techniques for such a complex nonlinear transient problem. The results provide valuable insights for predictive low-dimensional modeling in reactive flow applications.

The paper is structured as follows: Section II describes the governing equations and discretization of the considered combustion application; Section III presents details of the dynamic mode decomposition and operator inference methods for low-dimensional modeling; numerical results are presented in Section IV; and conclusions are stated in Section V.

II. Combustion application

In this section, we give details on the combustion model, which is solved with the General Equation and Mesh Solver (GEMS) CFD code [34]. Specifically, we consider the single-injector combustor computational model setup from [2] with the computational domain shown in Figure 1a, which also shows the four locations where we monitor the state variables. We briefly describe the model here and refer to [2, 34] for more details.



(a) The computational domain and the locations where we measure the state variables.

(b) Example temperature field of a solution snapshot over the computational domain.

Fig. 1 Setup and geometry of the single-injector combustor.

A. Governing equations

The dynamics of the two-dimensional combustor are governed by the conservation equations for mass, momentum, energy and species mass fractions,

$$\frac{\partial \vec{q}^{\,c}}{\partial t} + \nabla \cdot (\vec{K} - \vec{K}_v) = \vec{S}, \tag{1}$$

describing the evolution of the conservative variables $\vec{q}^{\,c}(x, y, t) = [\rho \;\; \rho v \;\; \rho w \;\; \rho e \;\; \rho Y_1 \;\; \ldots \;\; \rho Y_4]^\top$. Here, ρ is the density ($\frac{\text{kg}}{\text{m}^3}$), v and w are the x and y velocity ($\frac{\text{m}}{\text{s}}$), e is the total energy ($\frac{\text{J}}{\text{m}^3}$), and Y_l is the lth species mass fraction with $l = 1, 2, 3, 4$. The inviscid flux is $\vec{K}$, viscous flux is denoted $\vec{K}_v$, and the vector $\vec{S}$ contains the source terms from the chemical reactions. In this implementation, following [35], four chemical species are modeled in a 1-step combustion

reaction governed by

$$CH_4 + 2O_2 \rightarrow CO_2 + 2H_2O,$$

and their molar concentrations are $c_1 = [CH_4]$, $c_2 = [O_2]$, $c_3 = [CO_2]$, and $c_4 = [H_2O]$. The general relationship between a species molar concentration, c_l, and a species mass fraction, Y_l, is $Y_l = \frac{c_l M_l}{\rho}$, where the molar mass of CH_4 is $M_1 = 16.04\frac{\text{g}}{\text{mol}}$, the molar mass of O_2 is $M_2 = 32.0\frac{\text{g}}{\text{mol}}$, the molar mass of CO_2 is $M_3 = 44.01\frac{\text{g}}{\text{mol}}$, and the molar mass of H_2O is $M_4 = 18.0\frac{\text{g}}{\text{mol}}$. In this work, we relate density and pressure to temperature through the ideal gas state equation $\rho = \frac{p}{RT}$, where $R(Y_1, \ldots, Y_4) = \frac{R_u}{M}$ and $M = \left(\sum_{l=1}^{4} \left(\frac{Y_l}{M_l}\right)\right)^{-1}$ is the average molar mass of the mixtures. Thus, we can obtain temperature via $T = \frac{p}{\rho\, R(Y_1,\ldots,Y_4)}$ from the states $\rho, p, Y_1, \ldots, Y_4$.

At the downstream end of the combustor, a non-reflecting boundary condition is imposed to maintain the chamber pressure. The corresponding input is

$$u(t) = 10^6(1 + 0.1\sin(2\pi f t)) \quad [\text{Pa}], \tag{2}$$

where $f = 5000$Hz. The top and bottom wall boundary conditions are no-slip conditions, and for the upstream boundary we impose constant mass flow at the inlets.

B. Spatially discretized high-fidelity model

GEMS uses the finite volume method to spatially discretize the conservation equations Eq. (1). The primitive variables $\vec{q}^{\,\text{p}} = [p\ v\ w\ T\ Y_1 \ldots Y_4]^\top$ are chosen as solution variables in GEMS. For a spatial discretization with n_x cells, this results in a dn_x-dimensional system of nonlinear ordinary differential equations (ODEs)

$$\frac{\mathrm{d}\mathbf{q}^{\text{p}}}{\mathrm{d}t} = \mathbf{G}(\mathbf{q}^{\text{p}}, u(t)), \qquad \mathbf{q}^{\text{p}}(0) = \mathbf{q}_0^{\text{p}}, \tag{3}$$

for $0 < t \leq T$, where we have $d = 8$ physical unknowns (four flow variables and four species concentrations) in the PDE governing equations. In Eq. (3), $\mathbf{q}^{\text{p}}(t) \in \mathbb{R}^{dn_x}$ is the discretized state vector at time t, $\mathbf{q}_0^{\text{p}}$ are the specified initial conditions, and $\frac{\mathrm{d}\mathbf{q}^{\text{p}}}{\mathrm{d}t}$ is the time derivative of the state vector at time t. The one-dimensional input $u(t) \in \mathbb{R}$ arises from the time-dependent boundary condition, Eq. (2). The nonlinear function $\mathbf{G} : \mathbb{R}^{dn_x} \times \mathbb{R} \rightarrow \mathbb{R}^{dn_x}$ maps the discretized states $\mathbf{q}^{\text{p}}$ and the input u to the state time derivatives, representing the spatial discretization of the governing equations described in Section II.A.

III. Data-driven reduced-order modeling for nonlinear systems

Our goal is to learn low-dimensional dynamical systems models from data of the combustion simulations described in the previous section. We describe in Section III.A our strategy for learning from transformed variables, which has been shown in [2, 3, 31, 32] to significantly simplify the learning problem. Next, we discuss two different data-driven reduced-order modeling strategies, which we compare numerically in Section IV. In Section III.B, we briefly introduce dynamic mode decomposition (DMDc) [27–29], and in Section III.C, we review operator inference (OPINF) [30].

A. Learning reduced models from transformed data

In [2], it was shown that the highly nonlinear governing equations Eq. (3) admit a largely quadratic form when expressed in the variables $\{p,\ v,\ w,\ \rho^{-1},\ c_1, \ldots c_4\}$. The authors then showed that quadratic data-driven reduced models of Eq. (3) accurately reproduce many of the quantities of interest, such as pressure oscillations, species concentration, and even full state fields. Subsequently, [3] showed that including the temperature as an additional variable leads to an even better set of learning variables for reduced models. Drawing from these results, we focus herein on the vector of discretized variables

$$\mathbf{q} = \begin{bmatrix} \mathbf{p}^\top & \mathbf{v}^\top & \mathbf{w}^\top & \mathbf{T}^\top & (1/\boldsymbol{\rho})^\top & \mathbf{Y}_1^\top & \cdots & \mathbf{Y}_4^\top \end{bmatrix}^\top \in \mathbb{R}^n \tag{4}$$

as the data that we use for model learning ($1/\boldsymbol{\rho}$ denotes the componentwise inverse). Here we use $n = 9n_x$ to simplify the notation of the state dimension. Our goal is to learn a reduced-order dynamical system in these variables. The next two sections discuss two alternative methods and strategies to achieve that task.

B. Dynamic mode decomposition with control (DMDc)

Dynamic mode decomposition (DMD) originated in the fluid dynamics community as a method to decompose complex flows into a representation based on spatiotemporal coherent structures and their dynamic evolution [28, 29]. DMDc is an extension of DMD that capitalizes on all of the advantages of DMD and provides the additional innovation of being able to disambiguate between the underlying dynamics and the effects of actuation, resulting in accurate input-output models [27]. The chosen variables $\mathbf{q}$ evolve in the form of a general finite-dimensional dynamical system, written in discrete-time form as

$$\mathbf{q}_{k+1} = \mathbf{G}_{\mathrm{d}}(\mathbf{q}_k, \mathbf{u}_k), \qquad k = 0, 1, 2, \ldots, \tag{5}$$

where for times $0 = t_0 < t_1 < \cdots < t_K = T$, $\mathbf{u}_k \equiv \mathbf{u}(t_k) \in \mathbb{R}^m$ is the input, $\mathbf{q}_k \equiv \mathbf{q}(t_k) \in \mathcal{M} \subseteq \mathbb{R}^n$ is the discretized state, $\mathcal{M}$ is the solution manifold, and $\mathbf{G}_{\mathrm{d}}$ denotes the discretized forward operator. Let g be a function defined on $\mathcal{M}$ whose outputs are *observables* of the system, that is, we have access to $g(\mathbf{q}_0), g(\mathbf{q}_1), \ldots, g(\mathbf{q}_K)$. The Koopman operator $\mathcal{K}$ for the dynamical system Eq. (5) is an infinite-dimensional linear operator that acts on such observable functions as

$$[\mathcal{K}g](\mathbf{q}_k) = g(\mathbf{G}_{\mathrm{d}}(\mathbf{q}_k, \mathbf{u}_k)) = g(\mathbf{q}_{k+1}).$$

Thus, the Koopman operator is the propagator of observables. The DMD algorithm yields a reduced-order approximation of the Koopman operator in the space of observables. In many cases, the observable is chosen as the full state, i.e., $g(\mathbf{q}) = \mathbf{q}$, yet other linear or nonlinear observables might yield better results. Moreover, extending the observables by nonlinear functions of the state variables has also been shown to improve the approximation properties of DMD [24]. Here, we focus on the learning variables from Eq. (4) as observables. We then collect the data of $K + 1$ observables and inputs in the snapshot matrices

$$\begin{aligned} \mathbf{Q} &= \begin{bmatrix} \mathbf{q}_0 & \mathbf{q}_1 & \cdots & \mathbf{q}_{K-1} \end{bmatrix} \in \mathbb{R}^{n\times K}, \\ \mathbf{Q}' &= \begin{bmatrix} \mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_{K} \end{bmatrix} \in \mathbb{R}^{n\times K}, \qquad \mathbf{U} = \begin{bmatrix} \mathbf{u}_0 & \mathbf{u}_1 & \cdots & \mathbf{u}_{K-1} \end{bmatrix} \in \mathbb{R}^{m\times K}. \end{aligned} \tag{6}$$

Since the (infinite-dimensional) Koopman operator $\mathcal{K}$ is linear, we approximate the dynamics of observables in finite dimensions as linear in the state and linear in the inputs, i.e.,

$$\mathbf{Q}' = \mathbf{A}^{\mathrm{DMDc}}\mathbf{Q} + \mathbf{B}\mathbf{U} \tag{7}$$

with a control operator $\mathbf{B} \in \mathbb{R}^{n\times m}$ that, in our application, encodes the location of the the external pressure force acting on the downstream end of the combustion chamber, see Eq. (2).* Defining $\mathbf{Q}'_u = \mathbf{Q}' - \mathbf{B}\mathbf{U} \in \mathbb{R}^{n\times K}$, Eq. (7) can be written as $\mathbf{Q}'_u = \mathbf{A}^{\mathrm{DMDc}}\mathbf{Q}$, hence $\mathbf{A}^{\mathrm{DMDc}}$ can be learned by minimizing $\|\mathbf{Q}'_u - \mathbf{A}^{\mathrm{DMDc}}\mathbf{Q}\|_F$. This yields the solution

$$\mathbf{A}^{\mathrm{DMDc}} = \mathbf{Q}'_u\mathbf{Q}^{\dagger},$$

where $\mathbf{Q}^{\dagger}$ is Moore–Penrose pseudoinverse of $\mathbf{Q}$. However, this strategy is numerically problematic due to ill-conditioning of the matrix $\mathbf{Q}$, and the resulting DMDc matrix $\mathbf{A}^{\mathrm{DMDc}} \in \mathbb{R}^{n\times n}$ is large and therefore expensive to analyze directly. For these reasons, and to achieve a computational speedup, DMDc seeks a low-dimensional approximation of the Koopman operator by approximating the state snapshot matrix $\mathbf{Q}$ with its rank-r singular value decomposition,

$$\mathbf{Q} \approx \mathbf{V}\mathbf{\Sigma}\mathbf{W}^{\top}, \tag{8}$$

where $\mathbf{V} \in \mathbb{R}^{n\times r}$, $\mathbf{\Sigma} \in \mathbb{R}^{r\times r}$, and $\mathbf{W} \in \mathbb{R}^{K\times r}$ contain the left singular vectors, singular values, and right singular vectors of $\mathbf{Q}$, respectively. The reduced rank r can be chosen based on an energy criterion, e.g., $\sum_{i=1}^{r}\sigma_i^2 \,/\, \sum_{i=1}^{\mathrm{rank}(\mathbf{Q})}\sigma_i^2 > \varepsilon_{\mathrm{tol}}$, where $\varepsilon_{\mathrm{tol}}$ is a chosen threshold; one commonly uses $\varepsilon = 0.95$ or $\varepsilon = 0.99$, which is interpreted as the basis covering 95% or 99% of the variation/energy in the data. We then obtain a low-dimensional approximation of the DMDc operator,

$$\widehat{\mathbf{A}}^{\mathrm{DMDc}} = \mathbf{V}^{\top}\mathbf{A}^{\mathrm{DMDc}}\mathbf{V} = \mathbf{V}^{\top}\mathbf{Q}'_u\mathbf{W}\mathbf{\Sigma}^{-1} \in \mathbb{R}^{r\times r}.$$

The corresponding DMDc ROM is defined by

$$\widehat{\mathbf{q}}_{k+1} = \widehat{\mathbf{A}}^{\mathrm{DMDc}}\mathbf{q}_k + \widehat{\mathbf{B}}\mathbf{u}_k,$$

*One can also treat the control operator $\mathbf{B}$ as an unknown in the learning problem, as demonstrated in [27].

where $\widehat{\mathbf{B}} = \mathbf{V}^\top \mathbf{B} \in \mathbb{R}^{r\times m}$. States in the original high-dimensional space are approximated as $\mathbf{q} \approx \mathbf{V}\widehat{\mathbf{q}}$.

To see that DMDc is a modal decomposition technique, consider the eigenvalue decomposition of the low-dimensional operator,

$$\widehat{\mathbf{A}}^{\text{DMDc}}\mathbf{\Omega} = \mathbf{\Omega}\mathbf{M}.$$

The eigenvalues $\mu_1, \ldots, \mu_r \in \mathbb{C}$ on the diagonal of $\mathbf{M}$ are called DMDc eigenvalues, and the DMDc modes (in the original high dimension) are the columns of the matrix

$$\mathbf{\Phi} = \mathbf{Q}_u' \mathbf{W}\mathbf{\Sigma}^{-1}\mathbf{\Omega} \in \mathbb{C}^{n\times r}.$$

One can easily verify that the DMDc eigenvalues and modes are a subset of the eigenpairs of $\mathbf{A}^{\text{DMDc}}$, i.e., $\mathbf{A}^{\text{DMDc}}\mathbf{\Phi} = \mathbf{\Phi}\mathbf{M}$. This makes DMDc a popular method for understanding the dynamics in complex systems from the perspective of individual DMDc modes evolving according to their associated eigenvalues and adjusted by the control at each step.

C. Polynomial operator inference

Polynomial operator inference [30] is a model learning procedure that mimics the projection-based ROM setting but obtains the operators defining the ROM in a fully data-driven way. Results regarding the convergence and error of the learned model operators compared to their intrusive projection-based counterparts can be found in [30, 36]. The starting point of any projection-based ROM method is to approximate the high-dimensional state $\mathbf{q}$ in a low-dimensional basis $\mathbf{V} \in \mathbb{R}^{n\times r}$, with $r \ll n$, as $\mathbf{q} \approx \mathbf{V}\widehat{\mathbf{q}}$. Using a Galerkin projection, it is straightforward to see that the polynomial structure of a full-order model is retained in the ROM after projection. Thus, our goal is to learn a ROM of the form

$$\frac{\mathrm{d}\widehat{\mathbf{q}}}{\mathrm{d}t} = \widehat{\mathbf{A}}\widehat{\mathbf{q}} + \sum_{i=2}^{\ell} \widehat{\mathbf{N}}_i \widehat{\mathbf{q}}^{(i)} + \widehat{\mathbf{B}}\mathbf{u} + \widehat{\mathbf{c}}, \tag{9}$$

where $\widehat{\mathbf{A}} \in \mathbb{R}^{r\times r}$ is the linear operator, the $\widehat{\mathbf{N}}_i \in \mathbb{R}^{r\times r^i}$ are matricized higher-order tensors, and $\widehat{\mathbf{q}}^{(i)}$ is a i-times Kronecker product of $\mathbf{q}$ (e.g., $\widehat{\mathbf{q}}^{(3)} = \widehat{\mathbf{q}} \otimes \widehat{\mathbf{q}} \otimes \widehat{\mathbf{q}}$). Moreover, $\widehat{\mathbf{B}} \in \mathbb{R}^{r\times m}$ and $\widehat{\mathbf{c}} \in \mathbb{R}^r$ is a constant vector.

As a first step, we collect K snapshots of the state by solving the high-fidelity model. We store the snapshots and the inputs used to generate them in the matrices

$$\mathbf{Q} = \begin{bmatrix} \mathbf{q}_0 & \mathbf{q}_1 & \cdots & \mathbf{q}_{K-1} \end{bmatrix} \in \mathbb{R}^{n\times K}, \qquad \mathbf{U} = \begin{bmatrix} \mathbf{u}_0 & \mathbf{u}_1 & \cdots & \mathbf{u}_{K-1} \end{bmatrix} \in \mathbb{R}^{m\times K},$$

which are similar to the data matrices used in DMDc, but which include data from the initial and final times. In general, $n \gg K$, so the matrix $\mathbf{Q}$ is tall and skinny. Second, since we want to learn a low-dimensional model, we need to identify a low-dimensional subspace of the high-fidelity solutions in which to learn the ROM. We elect to use the proper orthogonal decomposition [37] (POD), also known as principal component analysis, to compute the low-dimensional subspace. The POD method computes the rank-r approximation of the snapshot matrix via singular value decomposition

$$\mathbf{Q} \approx \mathbf{V}\mathbf{\Sigma}\mathbf{W}^\top,$$

so that $\mathbf{V} \in \mathbb{R}^{n\times r}$, which in this context is called the POD basis. Third, we project the state snapshot data onto the POD subspace spanned by the columns of $\mathbf{V}$ and obtain the reduced snapshot matrices

$$\widehat{\mathbf{Q}} = \mathbf{V}^\top\mathbf{Q} = \begin{bmatrix} \widehat{\mathbf{q}}_0 & \widehat{\mathbf{q}}_1 & \cdots & \widehat{\mathbf{q}}_{K-1} \end{bmatrix} \in \mathbb{R}^{r\times K}, \qquad \dot{\widehat{\mathbf{Q}}} = \begin{bmatrix} \dot{\widehat{\mathbf{q}}}_0 & \dot{\widehat{\mathbf{q}}}_1 & \cdots & \dot{\widehat{\mathbf{q}}}_{K-1} \end{bmatrix} \in \mathbb{R}^{r\times K},$$

where the columns of $\dot{\widehat{\mathbf{Q}}}$ are computed from the columns of $\widehat{\mathbf{Q}}$ using any time derivative approximation (see, e.g., [38–40]), or which can be obtained—if available—by collecting and projecting snapshots of $\mathbf{G}$, i.e., $\dot{\widehat{\mathbf{q}}}_k = \mathbf{V}^\top\mathbf{G}(\mathbf{q}_k, \mathbf{u}_k)$.

Operator inference solves a least-squares problem to find the reduced operators that best match the projected snapshot data in a minimum residual sense. In order to learn a polynomial ROM as in Eq. (9), operator inference solves the least-squares problem

$$\min_{\widehat{\mathbf{A}}\in\mathbb{R}^{r\times r}, \widehat{\mathbf{N}}_i\in\mathbb{R}^{r\times r^i}, \widehat{\mathbf{B}}\in\mathbb{R}^{r\times m}, \widehat{\mathbf{c}}\in\mathbb{R}^r} \left\| \widehat{\mathbf{A}}\widehat{\mathbf{Q}} + \sum_{i=2}^{\ell} \widehat{\mathbf{N}}_i \widehat{\mathbf{Q}}^{(i)} + \widehat{\mathbf{B}}\mathbf{U} + \widehat{\mathbf{c}}\mathbf{1}_K^\top - \dot{\widehat{\mathbf{Q}}} \right\|_F^2, \tag{10}$$

where $\mathbf{1}_K = [1, 1, \ldots, 1]^\top \in \mathbb{R}^K$ and $\widehat{\mathbf{Q}}^{\textcircled{i}} = \left[\widehat{\mathbf{q}}_0^{\textcircled{i}} \quad \widehat{\mathbf{q}}_1^{\textcircled{i}} \quad \cdots \quad \widehat{\mathbf{q}}_{K-1}^{\textcircled{i}}\right] \in \mathbb{R}^{r^i \times K}$. In sum, operator inference enables us to compute the ROM operators $\widehat{\mathbf{A}}$, $\widehat{\mathbf{N}}_i$, $\widehat{\mathbf{B}}$, and $\widehat{\mathbf{c}}$ directly from data of the high-fidelity solver, but without access to any of its internal codes or routines.

The unknown operators of Eq. (9) can be concatenated in a matrix

$$\mathbf{O} = \left[\begin{array}{ccccccc} \widehat{\mathbf{A}} & \widehat{\mathbf{N}}_2 & \cdots & \widehat{\mathbf{N}}_\ell & \widehat{\mathbf{B}} & \widehat{\mathbf{c}} \end{array}\right] \in \mathbb{R}^{r \times (r + r^2 + \cdots + r^\ell + m + 1)},$$

and the known low-dimensional data in the data matrix

$$\mathbf{D} = \left[\begin{array}{ccccccc} \widehat{\mathbf{Q}}^\top & (\widehat{\mathbf{Q}}^{\textcircled{2}})^\top & \cdots & (\widehat{\mathbf{Q}}^{\textcircled{\ell}})^\top & \mathbf{U}^\top & \mathbf{1}_K \end{array}\right] \in \mathbb{R}^{K \times (r + r^2 + \cdots + r^\ell + m + 1)}. \tag{11}$$

It was proven in [30] that Eq. (10) can be rewritten as r independent least-squares problems, which makes the method efficient and scalable. Additionally, to avoid overfitting and combat model form error, [2, 3] noted that regularization becomes necessary. More details on the regularization choices are given in Section IV. With these observations, Eq. (10) becomes

$$\min_{\mathbf{O}} \left\| \mathbf{D}\mathbf{O}^\top - \dot{\widehat{\mathbf{Q}}}^\top \right\|_F^2 + \left\| \mathbf{\Gamma}\mathbf{O}^\top \right\|_F^2 \tag{12}$$

where $\mathbf{\Gamma} = \operatorname{diag}(\lambda_1 \mathbf{I}_{(r)}, \lambda_2 \mathbf{I}_{(r^2)}, \ldots, \lambda_\ell \mathbf{I}_{(r^\ell)}, \lambda_1 \mathbf{I}_{(m)}, \lambda_1) \in \mathbb{R}^{(r + r^2 + \cdots + r^\ell + m + 1) \times (r + r^2 + \cdots + r^\ell + m + 1)}$ is a diagonal regularization matrix with $\mathbf{I}_{(p)}$ the $p \times p$ identity matrix.

Similar to the case of regularization for quadratic operator inference ($\ell = 2$) in [3], we define the regularized learning problem for cubic operator inference ($\ell = 3$) as

$$\min_{\widehat{\mathbf{A}}, \widehat{\mathbf{N}}_2, \widehat{\mathbf{N}}_3, \widehat{\mathbf{B}}, \widehat{\mathbf{c}}} \left[\left\| \widehat{\mathbf{A}}\widehat{\mathbf{Q}} + \widehat{\mathbf{N}}_2\widehat{\mathbf{Q}}^{\textcircled{2}} + \widehat{\mathbf{N}}_3\widehat{\mathbf{Q}}^{\textcircled{3}} + \widehat{\mathbf{B}}\mathbf{U} + \widehat{\mathbf{c}}\mathbf{1}_K^\top - \dot{\widehat{\mathbf{Q}}} \right\|_F^2 + \lambda_1 (\|\widehat{\mathbf{A}}\|_F^2 + \|\widehat{\mathbf{B}}\|_F^2 + \|\widehat{\mathbf{c}}\|_2^2) + \lambda_2 \|\widehat{\mathbf{N}}_2\|_F + \lambda_3 \|\widehat{\mathbf{N}}_3\|_F \right].$$

Here, λ_1 regularizes the linear, input, and constant operators, λ_2 regularizes the quadratic operator, and λ_3 regularizes the cubic operator. In this case, the optimization takes the form of Eq. (12) with the matrices

$$\begin{aligned} \mathbf{O} &= \left[\begin{array}{ccccc} \widehat{\mathbf{A}} & \widehat{\mathbf{N}}_2 & \widehat{\mathbf{N}}_3 & \widehat{\mathbf{B}} & \widehat{\mathbf{c}} \end{array}\right] \in \mathbb{R}^{r \times (r + r^2 + r^3 + m + 1)}, \\ \mathbf{D} &= \left[\begin{array}{ccccc} \widehat{\mathbf{Q}}^\top & (\widehat{\mathbf{Q}}^{\textcircled{2}})^\top & (\widehat{\mathbf{Q}}^{\textcircled{3}})^\top & \mathbf{U}^\top & \mathbf{1}_K \end{array}\right] \in \mathbb{R}^{K \times (r + r^2 + r^3 + m + 1)}, \\ \mathbf{\Gamma} &= \operatorname{diag}(\lambda_1 \mathbf{I}_{(r)}, \lambda_2 \mathbf{I}_{(r^2)}, \lambda_3 \mathbf{I}_{(r^3)}, \lambda_1 \mathbf{I}_{(m)}, \lambda_1) \in \mathbb{R}^{(r + r^2 + r^3 + m + 1) \times (r + r^2 + r^3 + m + 1)}. \end{aligned}$$

The solution to Eq. (12) is then obtained by solving the normal equations

$$\left(\mathbf{D}^\top \mathbf{D} + \mathbf{\Gamma}^\top \mathbf{\Gamma}\right) \mathbf{O}^\top = \mathbf{D}^\top \dot{\widehat{\mathbf{Q}}}^\top.$$

IV. Numerical results

This section presents our numerical findings for the single-injector combustion process. Section IV.A presents details of the numerical tests and their implementation, and Sections IV.B and IV.C present the final results. The public repository `https://github.com/Willcox-Research-Group/ROM-OpInf-Combustion-2D` contains the code and details for these numerical experiments.

A. Data and implementation details

1. Discretization and data size

The computational domain of the combustion chamber displayed in Fig. 1a is discretized into finite volumes with a total number of $n_x = 38,523$ cells. We use the physical variables in Eq. (4) for model learning, for a total of $d = 9$ physical variables. Consequently, the dimension n of each snapshot $\mathbf{q}_k$ is $n = 9n_x = 346,707$. The full-order model is simulated for a time duration of 6ms with a time step size of 1×10^{-7}s, resulting in 60,000 snapshots of the high-fidelity GEMS model. The data set is publicly available at [41]. We use $K = 20,000$ snapshots for the model training and the remaining 40,000 snapshots in time for the model validation. The training data matrix $\mathbf{Q}$ is thus arranged as follows:

$$\mathbf{Q} = \overbrace{\begin{bmatrix} \mathbf{p}_0 & \mathbf{p}_1 & \cdots & \mathbf{p}_{19,999} \\ \mathbf{v}_0 & \mathbf{v}_1 & \cdots & \mathbf{v}_{19,999} \\ \mathbf{w}_0 & \mathbf{w}_1 & \cdots & \mathbf{w}_{19,999} \\ (\boldsymbol{\rho}^{-1})_0 & (\boldsymbol{\rho}^{-1})_1 & \cdots & (\boldsymbol{\rho}^{-1})_{19,999} \\ \mathbf{T}_0 & \mathbf{T}_1 & \cdots & \mathbf{T}_{19,999} \\ \mathbf{Y}_0^{\mathrm{CH_4}} & \mathbf{Y}_1^{\mathrm{CH_4}} & \cdots & \mathbf{Y}_{19,999}^{\mathrm{CH_4}} \\ \mathbf{Y}_0^{\mathrm{O_2}} & \mathbf{Y}_1^{\mathrm{O_2}} & \cdots & \mathbf{Y}_{19,999}^{\mathrm{O_2}} \\ \mathbf{Y}_0^{\mathrm{H_2O}} & \mathbf{Y}_1^{\mathrm{H_2O}} & \cdots & \mathbf{Y}_{19,999}^{\mathrm{H_2O}} \\ \mathbf{Y}_0^{\mathrm{CO_2}} & \mathbf{Y}_1^{\mathrm{CO_2}} & \cdots & \mathbf{Y}_{19,999}^{\mathrm{CO_2}} \end{bmatrix}}^{\text{20,000 columns = snapshots in time}} \Bigg\} \text{346,707 rows = spatial values of 9 variables} \tag{13}$$

2. Scaling of data

The range of the data, i.e., the minimum, maximum, and mean values are listed in Table 1. This illustrates that there is a significant difference in magnitude, for instance, the mean H_2O concentration and the mean pressure differ by nine orders of magnitude. Therefore, scaling is necessary for this problem, as is common in multiphysics and multiscale applications. For our experiments, we scale each variable entrywise to the interval $[-1, 1]$ via a maximum absolute scaling: pressure, temperature, and specific volume are first centered about 0, then each variable is normalized by its maximum absolute value. To be precise, let $p_{k,j}$ be the jth entry of the kth pressure snapshot $\mathbf{p}_k \in \mathbb{R}^{n_x}$; the pressure is scaled as

$$p_{k,i}^{\text{scaled}} = \frac{p_{k,i} - \bar{p}}{\max\limits_{\ell,j} \left\{|p_{\ell,j} - \bar{p}|\right\}}, \qquad \bar{p} = \frac{1}{Kn_x} \sum_{k=1}^{K} \sum_{i=1}^{n_x} p_{k,i}.$$

Temperature and specific volume are transformed in the same way. Velocity and species concentration data are normalized without additional centering, e.g., $v_{k,i}^{\text{scaled}} = v_{k,i} \big/ \max_{\ell,j} \left\{|v_{\ell,j}|\right\}$. In addition to significantly improving the conditioning of the learning problem, this scaling strategy preserves the null velocities, null molar species concentrations, and mean pressure, temperature, and specific volume in each ROM.

After scaling the raw data, the SVD of the scaled data is computed (details are in the next section), the ROM is built and simulated, and the states generated by the ROM simulation are then unscaled before errors are computed. In other words, learning and reduced-order computing are done in a scaled space, but input and output states are compared in the original state space.

Table 1 The minimum, mean, and maximum of each of the nine physical variables in the state vector q over the entire data set of 60,000 **snapshots.**

Variable	Minimum	Mean	Maximum
Pressure ($\mathbf{p}$)	8.5797×10^5	1.1478×10^6	1.5359×10^6
x-velocity ($\mathbf{v}$)	−427.0257	72.5567	332.6312
y-velocity ($\mathbf{w}$)	−324.4188	1.3172	212.9931
Temperature ($\mathbf{T}$)	262.2406	1041.6196	2804.4548
Specific Volume ($\boldsymbol{\rho}^{-1}$)	0.0995	0.3574	1.0583
CH_4 Molar Concentration ($\mathbf{c}^{\mathrm{CH_4}}$)	0.0000	0.0334	0.6269
O_2 Molar Concentration ($\mathbf{c}^{\mathrm{O_2}}$)	0.0000	0.0369	0.0715
H_2O Molar Concentration ($\mathbf{c}^{\mathrm{H_2O}}$)	0.0000	0.0019	0.0081
CO_2 Molar Concentration ($\mathbf{c}^{\mathrm{CO_2}}$)	0.0000	0.1022	0.1671

3. Randomized singular value decomposition

The computation of the full singular value decomposition (SVD) of the snapshot matrix $\mathbf{Q} \in \mathbb{R}^{346,707\times 20,000}$ in Eq. (13) has complexity $O(Kn^2 + K^3)$ and is therefore prohibitively expensive. Randomized methods for computing the SVD, abbreviated RSVD, are often faster and quite robust [42]. The basic idea of RSVD is to sample the range space of $\mathbf{Q}$ by computing the action of $\mathbf{Q}$ on random vectors (that are orthonormal with high probability). Once the range space is found, orthonormalization is performed and a smaller deterministic SVD is computed to obtain the right singular vectors. This prototype RSVD algorithm, presented also in [42], is summarized here in Algorithm 1 and implemented in the python package `Scikit-learn` [43] under the function `sklearn.utils.extmath.randomized_svd()`.

Algorithm 1 Randomized SVD

Input: Data $\mathbf{Q} \in \mathbb{R}^{n\times K}$, rank $r \in \mathbb{Z}^+$, oversampling parameter $\ell \in \mathbb{Z}^+$ ($\ell + r \leq \min\{n, K\}$)
Output: Approximate SVD factors $\mathbf{V} \in \mathbb{R}^{n\times r}$, $\mathbf{\Sigma} \in \mathbb{R}^{r\times r}$, $\mathbf{W}^\top \in \mathbb{R}^{r\times K}$

1: Pick an integer exponent $\kappa \in \mathbb{Z}^+$ (say $\kappa = 1$ or $\kappa = 2$).
2: Generate a Gaussian test matrix $\mathbf{\Gamma} \in \mathbb{R}^{n\times(\ell+r)}$.
3: Form $\mathbf{Y} = (\mathbf{Q}\mathbf{Q}^\top)^\kappa \mathbf{Q}\mathbf{\Gamma} \in \mathbb{R}^{n\times(\ell+r)}$ by multiplying alternately with $\mathbf{Q}$ and $\mathbf{Q}^\top$.
4: Construct a matrix $\mathbf{P} \in \mathbb{R}^{n\times(\ell+r)}$ whose columns form an orthonormal basis for the range of $\mathbf{Y}$.
5: Let $\mathbf{B} = \mathbf{P}^\top\mathbf{Q} \in \mathbb{R}^{(\ell+r)\times K}$.
6: Compute the SVD factors $\check{\mathbf{V}} \in \mathbb{R}^{(\ell+r)\times(\ell+r)}$, $\tilde{\mathbf{\Sigma}} \in \mathbb{R}^{(\ell+r)\times(\ell+r)}$ and, $\tilde{\mathbf{W}}^\top \in \mathbb{R}^{(\ell+r)\times K}$ of the (small) matrix $\mathbf{B}$, i.e., $\mathbf{B} = \check{\mathbf{V}}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{W}}^\top$.
7: Set $\tilde{\mathbf{V}} = \mathbf{P}\check{\mathbf{V}} \in \mathbb{R}^{n\times(\ell+r)}$.
8: Extract the rank-r SVD factors $\mathbf{V} = \tilde{\mathbf{V}}_{:,1:r} \in \mathbb{R}^{n\times r}$, $\mathbf{\Sigma} = \tilde{\mathbf{\Sigma}}_{1:r,1:r} \in \mathbb{R}^{r\times r}$, and $\mathbf{W}^\top = \tilde{\mathbf{W}}^\top_{1:r,:} \in \mathbb{R}^{r\times K}$.

4. Regularization of the operator inference optimization problem

As reported in [2, 3], regularization of the operator inference problem Eq. (12) is necessary to produce stable ROMs. For the quadratic case ($\ell = 2$), we find that the regularization hyperparameters $\lambda_1 = 10^2$ for the linear terms and $\lambda_2 = 10^6$ for the quadratic terms yield good OPINF ROMs. In the cubic case ($\ell = 3$), we use $\lambda_1 = 10^4$, $\lambda_2 = 10^5$ and $\lambda_3 = 10^6$. In both the cases we have chosen the model basis size as $r = 44$. These hyperparameter choices are not necessarily optimal, but they suffice for our numerical comparison; see [3] for a discussion on algorithmically selecting hyperparameters that are optimal in an empirical sense.

5. Signal analysis in frequency domain

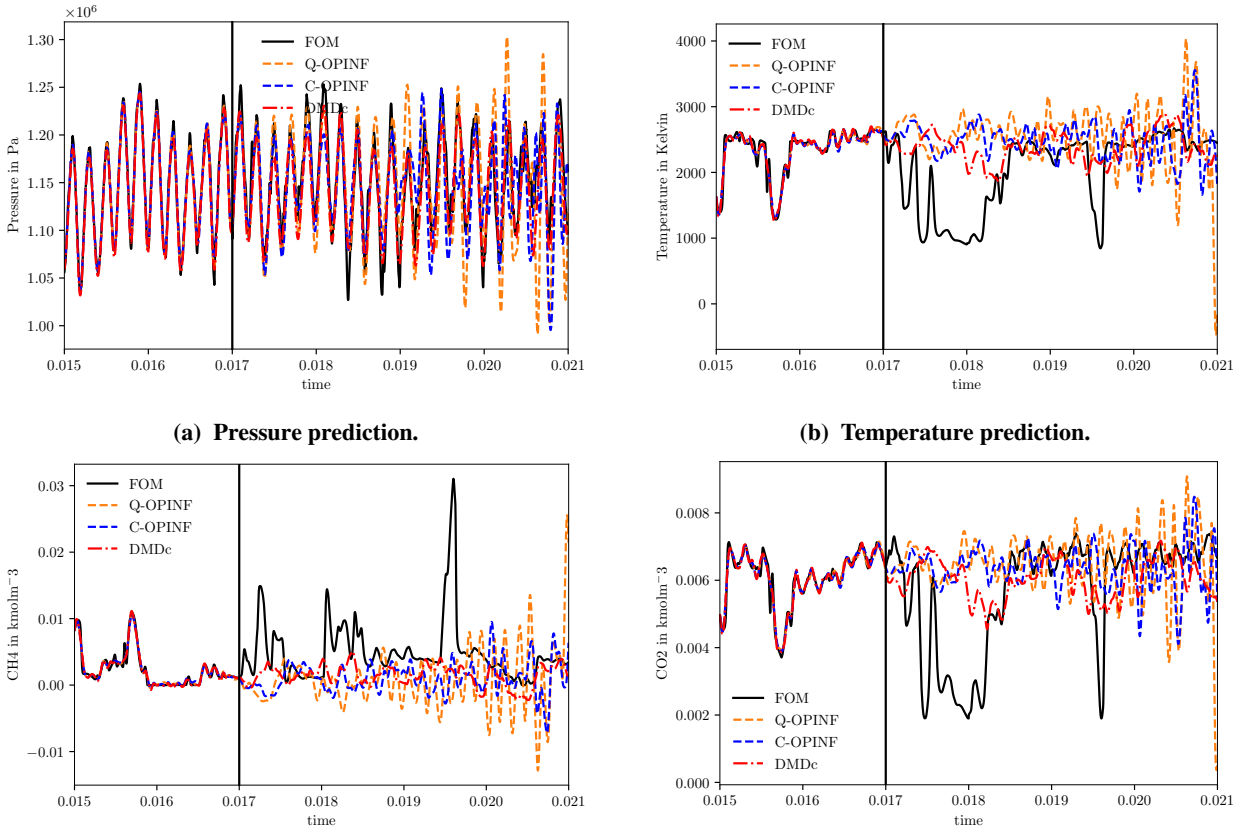
We compare the time series for several physical variables at specific locations in the combustor for our learned ROMs with the full-order GEMS data. A more complete picture of the approximation quality, however, arises when we compute the spectral information inherent in the signals, such as amplitude and phase. The frequency resolution for the spectral analysis depends on the number of time snapshots available. We have $K = 60{,}000$ snapshots in time with time step $\Delta T = 1 \times 10^{-7}$s, so the sampling frequency is $f_s = 1/(\Delta T) = 10$MHz. We analyze the training and testing data separately. Therefore, the lowest frequency resolution for the training data is $f_s/20{,}000 = 500$Hz, and for the testing data it is $f_s/40{,}000 = 250$Hz. To analyze the data, we employ the standard Fast Fourier Transform (FFT) from `SciPy-fft` [44]. The FFT of the length-K time-domain sequence $\{x_0, x_1, \ldots, x_{K-1}\}$ is the length-K sequence $\{X_0, X_1, \ldots, X_{K-1}\}$ where

$$X_k = \sum_{j=0}^{K-1} \exp\left(-2\pi i \frac{kj}{K}\right) x_j \qquad k = 0, 1, \ldots, K-1.$$

In our context, $x_k \in \mathbb{R}$ represents the value a single physical variable at a particular location in the domain at time t_k, while $X_k \in \mathbb{C}$ is the amplitude of the kth frequency of that signal.

B. Comparison of learned DMDc and OPINF ROM performance

In this section we present a numerical comparison, in both the temporal and frequency domains, of quadratic and cubic OPINF reduced models and DMDc reduced models, all of which are compared to the full-order GEMS data (we also refer to the GEMS data as the full-order model, or FOM). We use $r = 44$ basis vectors for both the quadratic operator inference (Q-OPINF) and the cubic operator inference (C-OPINF). As stated earlier, the training data contains



(a) Pressure prediction.

(b) Temperature prediction.

(c) CH_4 concentration.

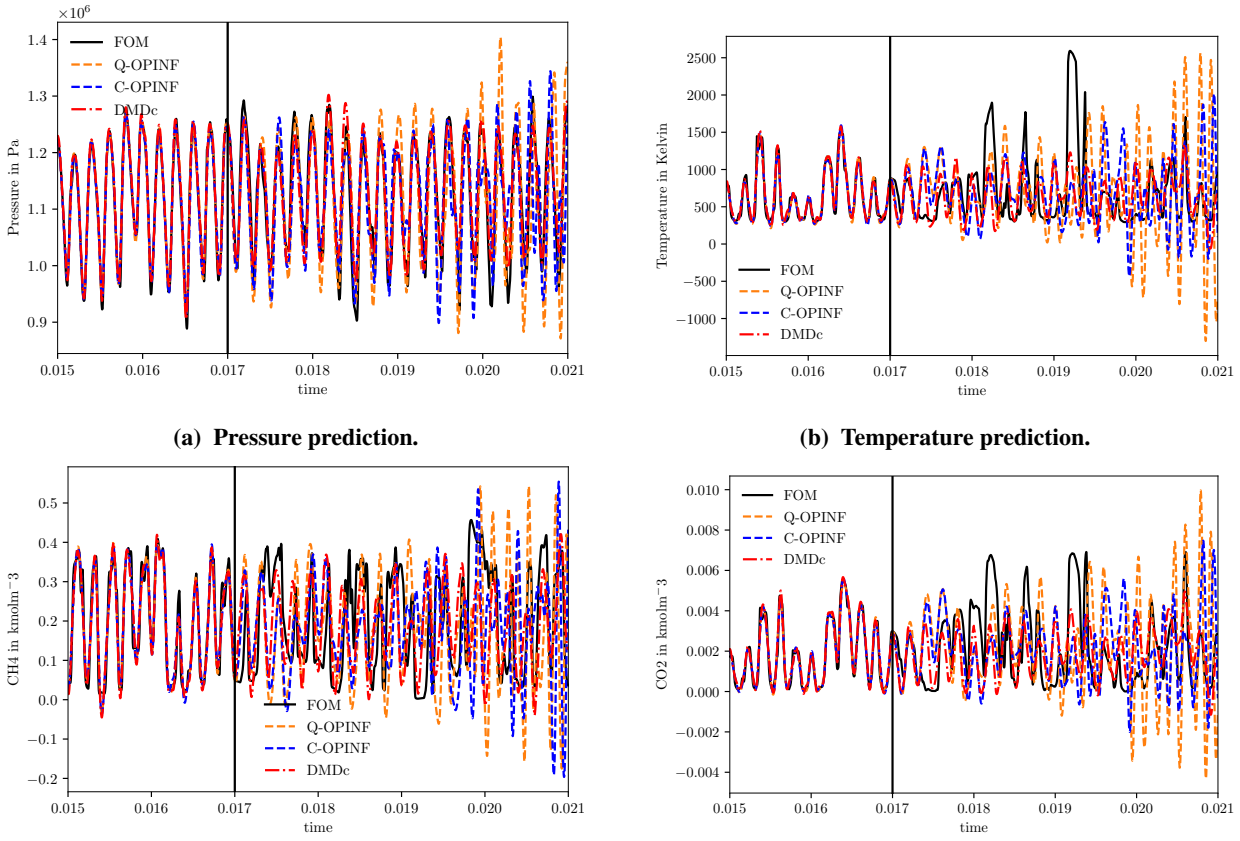
(d) CO_2 concentration.

Fig. 2 Training and prediction at monitor location 2. The black vertical line denotes the end of the training data and the beginning of the test data. All the three models, Q-OPINF, C-OPINF, and DMDc have the same basis size of $r = 44$.

$K = 20{,}000$ snapshots and the testing data contains another 40,000 snapshots. Therefore, the results show a true prediction of ROM simulations for 200% past the training data.

Figures 2 and 3 show the pressure, temperature, and two chemical species—CO_2 and CH_4—at monitor locations 2 and 3 in the combustor, respectively (marked in Figure 1a). The end of the training data is marked by the vertical black line. These two locations are physically representative of different phenomena that are both challenging to capture with a low-dimensional representation: at monitor location 3, significant mixing occurs due to the merging of the fuel and oxidizer inlets; at monitor location 2, the combustion is most active with high rates of heat release. From Figure 2, we observe that the Q-OPINF model for this particular setting has large oscillations, and seems to have an onset of an instability toward the end of the testing interval. Adding the cubic nonlinear terms to the OPINF model dampens the oscillations, increasing the stability of the ROM. The same pattern is observed in Figure 3. The DMDc model also predicts the dynamics quite well, with generally smaller oscillations than the Q-OPINF model.

Figure 4 and Figure 5 show the spectral content (amplitude vs frequency) of the signals at the monitoring locations. The left column of these figures shows that information for the training interval, and the right column analyzes the time traces purely on the testing interval. It is evident that each ROM captures the low-frequency content well and that pressure is best approximated in all models. The peak in the pressure plot at 5,000Hz is the dominant frequency in the flow, a result of the external forcing term $u(t)$ from Eq. (2). Both DMDc and OPINF produce accurate ROMs on the training interval. However, Q-OPINF and C-OPINF include more dynamics in the higher frequency range. Furthermore, at the monitor location 3—where there is not much mixing and turbulence in the flow—each ROM method predicts the pressure, temperature and the species accurately.



(a) Pressure prediction.

(b) Temperature prediction.

(c) CH_4 concentration.

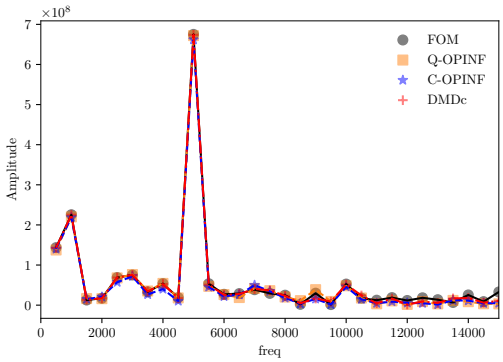
(d) CO_2 concentration.

Fig. 3 Training and prediction at monitor location 3. The black vertical line denotes the end of the training data and the beginning of the test data. All the three models, Q-OPINF, C-OPINF, and DMDc have the same basis size of $r = 44$.

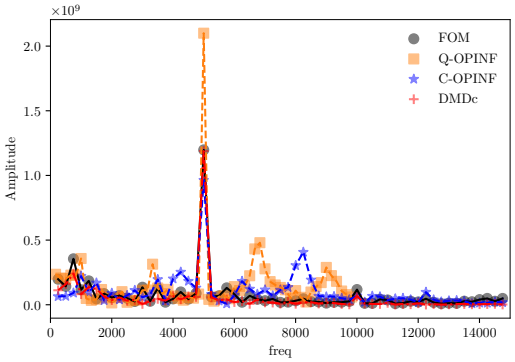
The high quality of the three low-dimensional modeling techniques on a problem of such complexity is significant. We attribute much of the good predictive performance of the ROMs to the fact that each strategy is based on the underlying structure of the physical equations, resulting in stable predictions well beyond the training regime.

C. Further assessment of cubic OPINF ROM

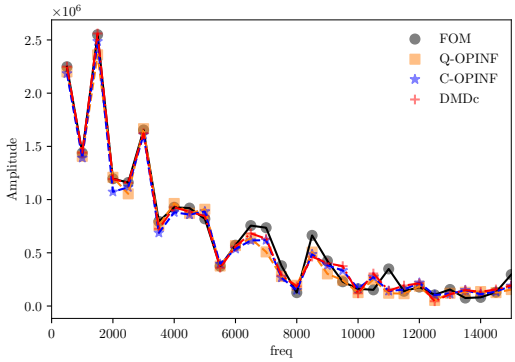
Since this is the first time that a cubic OPINF ROM has been developed for this combustion application, we also assess the performance of the cubic operator inference at different values of r. For the quadratic OPINF models, detailed analyses are available in [2, 3]. Figure 6 shows several different errors between the C-OPINF and the GEMS data at different values of r. Missing dots indicate that the resulting model is numerically unstable for that value of r. As previously stated, we use $\lambda_1 = 10^4$, $\lambda_2 = 10^5$ and $\lambda_3 = 10^6$ as the regularization parameters. The errors are computed after the ROM solutions are reconstructed in the full-order dimension and rescaled to the original variable ranges. Following [2], the pressure and temperature error are calculated using $\|\xi_k^{\text{FOM}} - \xi_k^{\text{ROM}}\| / \|\xi_k^{\text{FOM}}\|$, where ξ_k denotes the value of one of the physical variables at the kth snapshot in time. Because of the small values of species concentrations, dividing by the true value can skew a small error. Similarly, velocities range from positive to negative, including zero. Thus, for the four species concentrations and horizontal and vertical velocities, we use a normalized error defined as $\|\xi_k^{\text{FOM}} - \xi_k^{\text{ROM}}\| / \max_j(\|\xi_j^{\text{FOM}}\|)$.



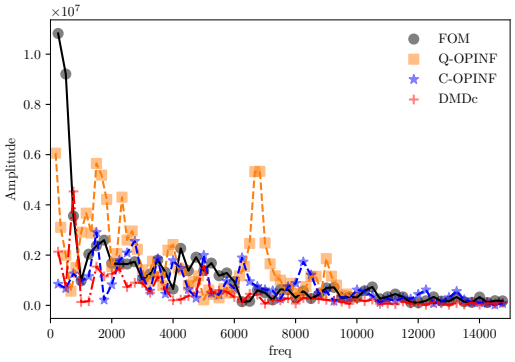
(a) Pressure data training results.



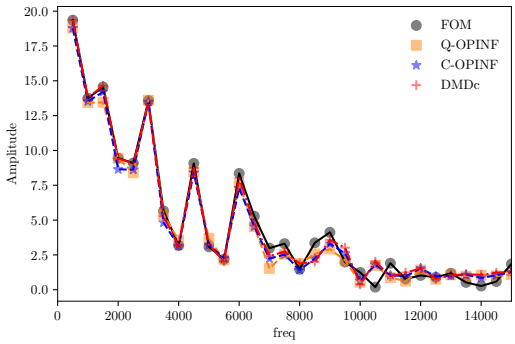
(b) Pressure data testing results.



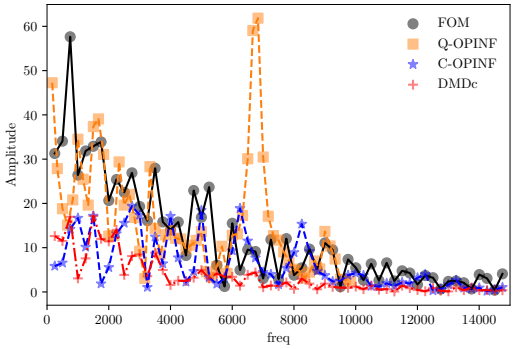
(c) Temperature data training results.



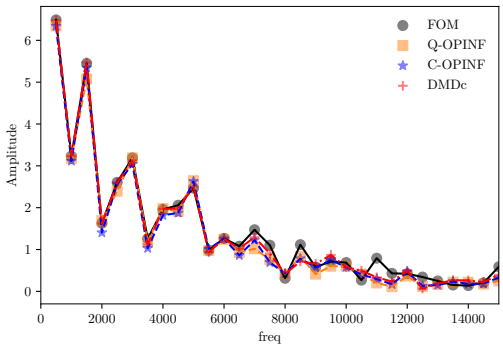
(d) Temperature data testing results.



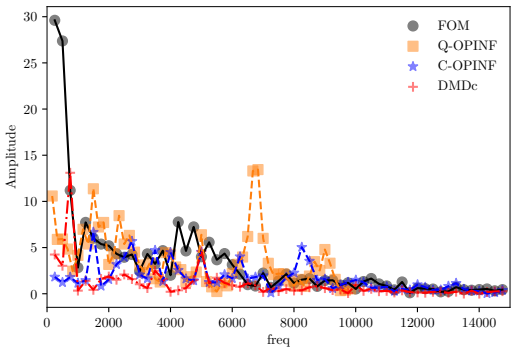
(e) CH_4 concentration data training results.



(f) CH_4 concentration data testing results.

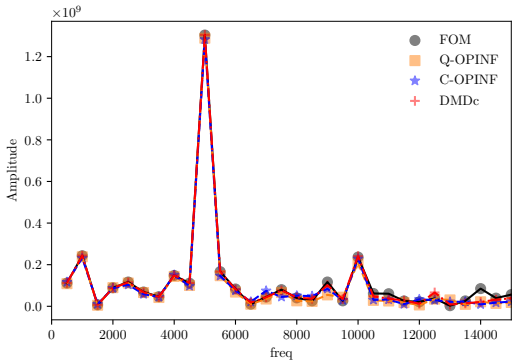


(g) CO_2 concentration data training results.

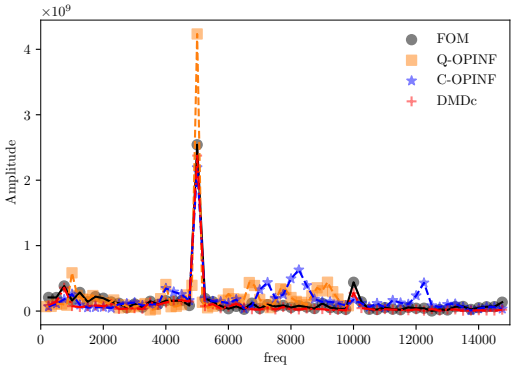


(h) CO_2 concentration data testing results.

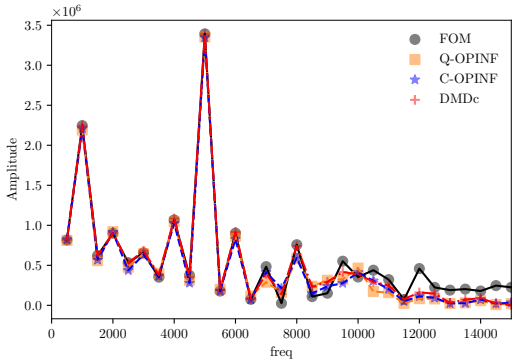
Fig. 4 Training (left column) and prediction (right columns) at monitor location 2. All the three models, Q-OPINF, C-OPINF, and DMDc have the same basis size of $r = 44$.



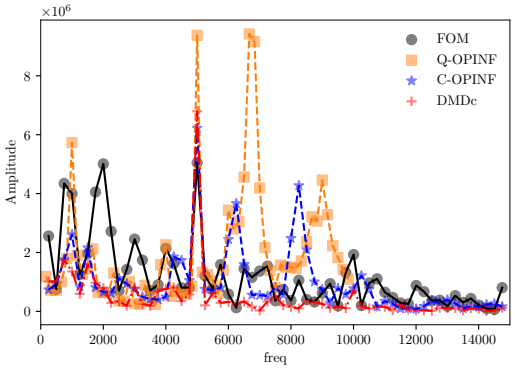
(a) Pressure data training results.



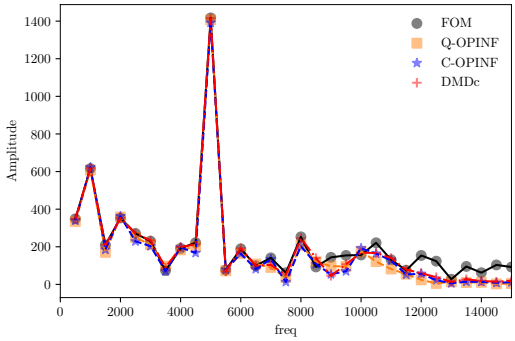
(b) Pressure data testing results.



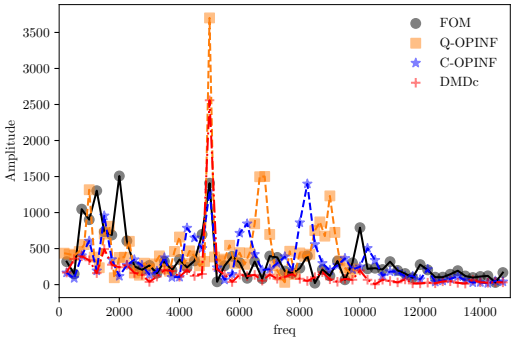
(c) Temperature data training results.



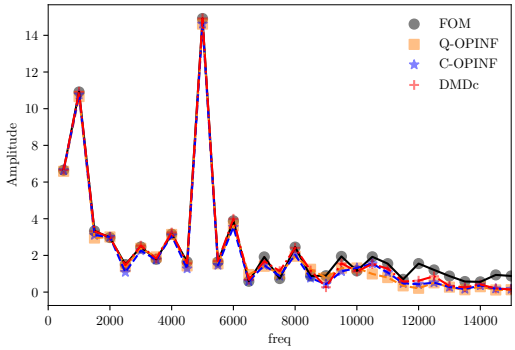
(d) Temperature data testing results.



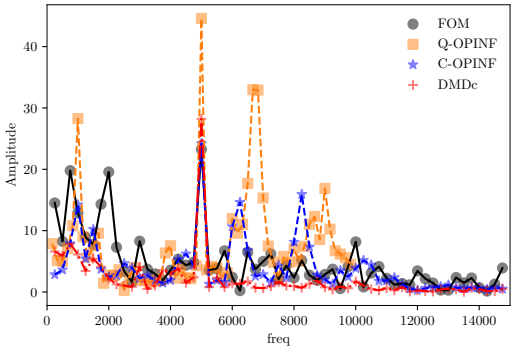
(e) CH_4 concentration data training results.



(f) CH_4 concentration data testing results.



(g) CO_2 concentration data training results.



(h) CO_2 concentration data testing results.

Fig. 5 Training (left column) and prediction (right columns) at monitor location 3. All the three models, Q-OPINF, C-OPINF, and DMDc have the same basis size of $r = 44$.

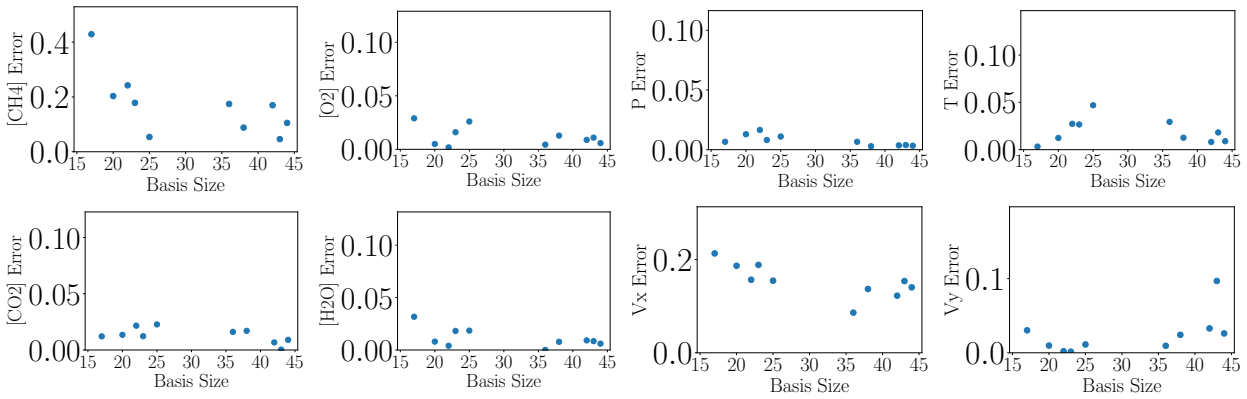


Fig. 6 Relative error quantities versus basis size r for cubic operator inference (C-OPINF) at the last time step of training data, averaged over the four monitor locations. Missing dots reflect unstable models.

We observe that for $26 \leq r \leq 35$, the C-OPINF models were unstable, yet lower-order (as low as $r = 16$) and higher-order models (up to $r = 44$) were again stable. However, for each r the same regularization hyperparameters were utilized, and choosing different hyperparameters may improve stability. While higher-order C-OPINF ROMs might be slightly more accurate, there is a trade-off regarding computational cost, as the size of the learning problem scales with r^3. We also observe again that pressure is the easiest physical variable to predict, and that species are rather challenging to predict correctly pointwise. This is unsurprising due to the highly advective nature of the chemical concentrations.

V. Conclusion

We presented a detailed numerical study comparing the performance of learned reduced models based on dynamic mode decomposition with control (DMDc) and operator inference with quadratic (Q-OPINF) and cubic (C-OPINF) terms on the challenging problem of a single-injector rocket combustor. This is the first time that a cubic OPINF model has been built for such a complex application, from which we derive new insights. The frequency domain analysis showed, as expected, that the OPINF models can capture higher frequency content, whereas DMDc does very well in capturing low-frequency content. After all, DMD is founded upon the notion of separating individual frequencies in the flow, and OPINF has POD-like characteristics in that it targets energetic modes that mix frequency content. In the time domain, we found that while each method performed well, the cubic OPINF model oscillates less wildly than its quadratic counterpart, indicating that the addition of the cubic term to the dynamical system equations has a stabilizing effect. The DMDc model also performed well on this test model; a difference of DMDc and OPINF is that DMD requires both the right and left singular vectors of the data matrix, $\mathbf{V}, \mathbf{W}$, whereas OPINF requires only $\mathbf{V}$. This could also hint at additional information that DMD uses versus OPINF and remains a topic of further study.

Acknowledgments

S.M. has been supported in part by the Air Force Center of Excellence on Multi-Fidelity Modeling of Rocket Combustor Dynamics (Air Force Office of Scientific Research) under award FA9550-17-1-0195, the U.S. Department of Energy AEOLUS MMICC center under award DE-SC0019303, and the U.S. Department of Energy National Nuclear Security Administration under award DE-NA0003969. B.K. has been supported in part by the National Science Foundation under award NSF-PHY-2028125, and the U.S. Department of Defense Newton award HQ-00342010022.

References

[1] Huang, C., Xu, J., Duraisamy, K., and Merkle, C., "Exploration of reduced-order models for rocket combustion applications," *2018 AIAA Aerospace Sciences Meeting*, 2018, p. 1183.

[2] Swischuk, R., Kramer, B., Huang, C., and Willcox, K., "Learning physics-based reduced-order models for a single-injector combustion process," *AIAA Journal*, Vol. 58:6, 2020, pp. 2658–2672.

[3] McQuarrie, S. A., Huang, C., and Willcox, K. E., "Data-driven reduced-order models via regularised Operator Inference for a single-injector combustion process," *Journal of the Royal Society of New Zealand*, Vol. 51, No. 2, 2021, pp. 194–211.

[4] Wentland, C. R., Huang, C., and Duraisamy, K., "Investigation of sampling strategies for reduced-order models of rocket combustors," *AIAA Scitech 2021 Forum*, 2021, p. 1371.

[5] Mayo, A. J., and Antoulas, A. C., "A framework for the solution of the generalized realization problem," *Linear Algebra and its Applications*, Vol. 425, No. 2-3, 2007, pp. 634–662.

[6] Peherstorfer, B., Gugercin, S., and Willcox, K., "Data-driven reduced model construction with time-domain Loewner models," *SIAM Journal on Scientific Computing*, Vol. 39, No. 5, 2017, pp. A2152–A2178.

[7] Schulze, P., Unger, B., Beattie, C., and Gugercin, S., "Data-driven structured realization," *Linear Algebra and its Applications*, Vol. 537, 2018, pp. 250–286.

[8] Antoulas, A. C., Gosea, I. V., and Heinkenschloss, M., "On the Loewner framework for model reduction of Burgers' equation," *Active Flow and Combustion Control 2018*, Springer, 2019, pp. 255–270.

[9] Kung, S.-Y., "A new identification and model reduction algorithm via singular value decomposition," *Proc. 12th Asilomar Conf. Circuits, Syst. Comput., Pacific Grove, CA*, 1978, pp. 705–714.

[10] Ma, Z., Ahuja, S., and Rowley, C. W., "Reduced-order models for control of fluids using the eigensystem realization algorithm," *Theoretical Comput. Fluid Dyn.*, Vol. 25, No. 1-4, 2011, pp. 233–247.

[11] Kramer, B., and Gugercin, S., "Tangential interpolation-based eigensystem realization algorithm for MIMO systems," *Mathematical and Computer Modeling of Dynamical Systems*, Vol. 22, No. 4, 2016, pp. 282–306.

[12] Kramer, B., and Gorodetsky, A., "System identification via CUR-factored Hankel approximation," *SIAM Journal on Scientific Computing*, Vol. 40, No. 2, 2018, pp. A848–A866.

[13] Gustavsen, B., and Semlyen, A., "Rational approximation of frequency domain responses by vector fitting," *IEEE Transactions on Power Systems*, Vol. 14, No. 3, 1999, pp. 1052–1061.

[14] Drmac, Z., Gugercin, S., and Beattie, C., "Quadrature-based vector fitting for discretized $\mathcal{H}_2$ approximation," *SIAM Journal on Scientific Computing*, Vol. 37, No. 2, 2015, pp. A625–A652.

[15] Wang, Q., Hesthaven, J. S., and Ray, D., "Non-intrusive reduced order modeling of unsteady flows using artificial neural networks with application to a combustion problem," *Journal of computational physics*, Vol. 384, 2019, pp. 289–307.

[16] Lee, K., and Carlberg, K. T., "Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders," *Journal of Computational Physics*, Vol. 404, 2020, p. 108973.

[17] Schmidt, M., and Lipson, H., "Distilling free-form natural laws from experimental data," *Science*, Vol. 324, No. 5923, 2009, pp. 81–85.

[18] Brunton, S. L., Proctor, J. L., and Kutz, J. N., "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proceedings of the National Academy of Sciences*, Vol. 113, No. 15, 2016, pp. 3932–3937.

[19] Gosea, I. V., and Antoulas, A. C., "Data-driven model order reduction of quadratic-bilinear systems," *Numerical Linear Algebra with Applications*, Vol. 25, No. 6, 2018, p. e2200.

[20] Chaturantabut, S., and Sorensen, D. C., "Nonlinear model reduction via discrete empirical interpolation," *SIAM Journal on Scientific Computing*, Vol. 32, No. 5, 2010, pp. 2737–2764.

[21] Peherstorfer, B., "Model reduction for transport-dominated problems via online adaptive bases and adaptive sampling," *SIAM Journal on Scientific Computing*, Vol. 42, No. 5, 2020, pp. A2803–A2836.

[22] Huang, C., Duraisamy, K., and Merkle, C., "Challenges in reduced order modeling of reacting flows," *2018 Joint Propulsion Conference*, Cincinnati, OH, 2018, p. 4675. Paper AIAA-2018-4675.

[23] Mezić, I., "Analysis of fluid flows via spectral properties of the Koopman operator," *Annual Review of Fluid Mechanics*, Vol. 45, 2013, pp. 357–378.

[24] Williams, M. O., Kevrekidis, I. G., and Rowley, C. W., "A data-driven approximation of the Koopman operator: Extending dynamic mode decomposition," *Journal of Nonlinear Science*, Vol. 25, No. 6, 2015, pp. 1307–1346.

[25] Otto, S. E., and Rowley, C. W., "Linearly recurrent autoencoder networks for learning dynamics," *SIAM Journal on Applied Dynamical Systems*, Vol. 18, No. 1, 2019, pp. 558–593.

[26] Netto, M., Susuki, Y., Krishnan, V., and Zhang, Y., "On Analytical Construction of Observable Functions in Extended Dynamic Mode Decomposition for Nonlinear Estimation and Prediction," *IEEE Control Systems Letters*, Vol. 5, No. 6, 2021, pp. 1868–1873. https://doi.org/10.1109/LCSYS.2020.3047586.

[27] Proctor, J. L., Brunton, S. L., and Kutz, J. N., "Dynamic mode decomposition with control," *SIAM Journal on Applied Dynamical Systems*, 2016, pp. 142–161.

[28] Rowley, C. W., Mezić, I., Bagheri, S., Schlatter, P., and Henningson, D. S., "Spectral analysis of nonlinear flows," *Journal of Fluid Mechanics*, Vol. 641, 2009, pp. 115–127.

[29] Schmid, P. J., "Dynamic mode decomposition of numerical and experimental data," *Journal of Fluid Mechanics*, Vol. 656, 2010, pp. 5–28.

[30] Peherstorfer, B., and Willcox, K., "Data-driven operator inference for nonintrusive projection-based model reduction," *Computer Methods in Applied Mechanics and Engineering*, Vol. 306, 2016, pp. 196–215.

[31] Qian, E., Kramer, B., Marques, A., and Willcox, K., "Transform & Learn: A data-driven approach to nonlinear model reduction," *AIAA Aviation 2019 Forum*, 2019, pp. 2019–3707.

[32] Qian, E., Kramer, B., Peherstorfer, B., and Willcox, K., "Lift & Learn: Physics-informed machine learning for large-scale nonlinear dynamical systems." *Physica D: Nonlinear Phenomena*, Vol. 406, 2020, p. 132401.

[33] Benner, P., Goyal, P., Kramer, B., Peherstorfer, P., and Willcox, K., "Operator inference for non-intrusive model reduction of systems with non-polynomial nonlinear terms," *Computer Methods in Applied Mechanics and Engineering*, Vol. 372, 2020, p. 113433.

[34] Harvazinski, M. E., Huang, C., Sankaran, V., Feldman, T. W., Anderson, W. E., Merkle, C. L., and Talley, D. G., "Coupling between hydrodynamics, acoustics, and heat release in a self-excited unstable combustor," *Physics of Fluids*, Vol. 27, No. 4, 2015, p. 045102.

[35] Westbrook, C. K., and Dryer, F. L., "Simplified reaction mechanisms for the oxidation of hydrocarbon fuels in flames," *Combustion Science and Technology*, Vol. 27, No. 1-2, 1981, pp. 31–43.

[36] Peherstorfer, B., "Sampling Low-Dimensional Markovian Dynamics for Preasymptotically Recovering Reduced Models from Data with Operator Inference," *SIAM Journal on Scientific Computing*, Vol. 42, No. 5, 2020, pp. A3489–A3515.

[37] Holmes, P., Lumley, J. L., Berkooz, G., and Rowley, C. W., *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*, Cambridge University Press, 2012.

[38] Martins, J. R., and Hwang, J. T., "Review and unification of methods for computing derivatives of multidisciplinary computational models," *AIAA journal*, Vol. 51, No. 11, 2013, pp. 2582–2599.

[39] Knowles, I., and Renka, R. J., "Methods for numerical differentiation of noisy data," *Electronic Journal of Differential Equations*, Vol. 21, 2014, pp. 235–246.

[40] Chartrand, R., "Numerical differentiation of noisy, nonsmooth, multidimensional data," *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, IEEE, 2017, pp. 244–248.

[41] Huang, C., "[Updated] 2D Benchmark Reacting Flow Dataset for Reduced Order Modeling Exploration [Data set]," *University of Michigan - Deep Blue Data*, 2020. https://doi.org/10.7302/nj7w-j319.

[42] Halko, N., Martinsson, P.-G., and Tropp, J. A., "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM Review*, Vol. 53, No. 2, 2011, pp. 217–288.

[43] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E., "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, Vol. 12, 2011, pp. 2825–2830.

[44] Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental algorithms for scientific computing in Python," *Nature Methods*, Vol. 17, 2020, pp. 261–272.