

Physically consistent predictive reduced-order modeling by enhancing operator inference with state constraints

Hyeonghun Kim *, Boris Kramer

Department of Mechanical and Aerospace Engineering, University of California, San Diego, CA, USA

ARTICLE INFO

Keywords:

Model reduction
Data-driven modeling
Nonlinear dynamical system
Multiscale
Multiphysics problem
Operator inference
State constraints

ABSTRACT

Numerical simulations of complex multiphysics systems, such as char combustion considered herein, yield numerous state variables that inherently exhibit physical constraints. This paper presents a new approach to augment Operator Inference—a methodology within scientific machine learning that enables learning from data a low-dimensional representation of a high-dimensional system governed by nonlinear partial differential equations—by embedding such state constraints in the reduced-order model predictions. In the model learning process, we propose a new way to choose regularization hyperparameters based on a key performance indicator. Since embedding state constraints improves the stability of the Operator Inference reduced-order model, we compare the proposed state constraints-embedded Operator Inference with the standard Operator Inference and other stability-enhancing approaches. For an application to char combustion, we demonstrate that the proposed approach yields state predictions superior to the other methods regarding stability and accuracy. It extrapolates over 200% past the training regime while being computationally efficient and physically consistent.

1. Introduction

Computational fluid dynamics (CFD) enables simulation of complex multiscale and multiphysics gas-solid flow dynamics in fluidized bed reactors that involve heat and mass transfer, particle collision, as well as homogeneous and heterogeneous chemical reactions [1]. However, CFD simulations are computationally expensive, particularly for problems governed by highly nonlinear partial differential equations (PDEs) where fine spatial and temporal discretization is required, which leads to high-dimensional nonlinear systems of ordinary differential equations (ODEs). This poses a significant obstacle for many-query tasks such as optimization, design, and uncertainty quantification.

Model reduction provides one avenue to address this challenge by constructing low-dimensional surrogate models for high-dimensional systems. In particular, projection-based model reduction techniques construct a reduced-order model (ROM) by projecting the high-dimensional system onto a low-dimensional subspace, or a nonlinear manifold, and solving the governing equations in the reduced space. This approach effectively incorporates the physics of the high-dimensional problem into the surrogate model; see the textbooks and survey [2–4]. Such an approach is intrusive because it requires access to codes that implement the semi-discretization of the PDEs of the high-fidelity system, often called the full-order model (FOM). However, in many real-world settings, the solution of highly complex problems often relies on complicated legacy code or commercial software, in which cases access to the FOM operators is not possible, rendering intrusive model reduction not applicable [5].

* Corresponding author.

E-mail address: hyk049@ucsd.edu (H. Kim).

Nonintrusive ROM methods have emerged as an efficient solution to many engineering challenges, leveraging the abundance of data and obviating a need for explicit numerical operators of the FOM. This paper focuses on Operator Inference (OpInf) [6], a nonintrusive ROM technique that has been extended in many directions and applied to a wide range of problems in science and engineering over the last decade. OpInf learns a low-dimensional representation of the high-dimensional system by exploiting the physical/mathematical structure and insights from high-fidelity numerical simulation data; see also the recent survey [7]. OpInf fits the ROM operators to the data by allowing for polynomial terms on the right-hand side of the ROM, effectively capturing the nonlinear dynamics of a system. To date, OpInf has demonstrated effectiveness in a variety of applications: advection-dominated magnetohydrodynamic systems [8], nonlinear aerodynamics and coupled aeroelastic flutter [9], incompressible flow [10], Rayleigh and Bénard convection [11], and soft robotics [12], among others. In addition, OpInf for parametric PDEs has been investigated for the shallow water equation [13], the heat equation and the FitzHugh-Nagumo neuron model [14], and a large-scale rotating detonation rocket engine combustion application [15]. However, stability and physical consistency of OpInf ROMs are still open challenges in certain applications, as we discuss below.

Numerous studies have been conducted on the stability of ROMs, a necessary condition for maintaining physical consistency. In the intrusive setting, the authors of [16] proposed an approach to stabilize projection-based ROMs through full-state feedback to modify unstable eigenvalues of a linear operator and through a constrained nonlinear least-squares optimization problem. In the nonintrusive setting, the authors in [17] adapted the trapping theorem [18] to promote global stability in data-driven machine learning models with quadratic nonlinearities. Another avenue to drive systems to stability is adding closure models; see the survey [19]. The authors in [20] enhance ROM stability and accuracy by proposing new closure models for POD ROMs of turbulent flows. Furthermore, the authors in [21,22] provided a stabilizing closure model for the 2D and 3D Boussinesq equations based on data-driven multi-parametric extremum seeking. Finally, the authors in [23] introduced a stabilization scheme that leverages the information encapsulated within the POD modes to predict the long-term dynamics of the full-order system accurately.

In the OpInf framework, various approaches have been proposed to develop physically consistent ROMs by promoting stability and preserving the physical/mathematical structure present in the FOM. To ensure the stability of OpInf ROMs, [24] introduced analytical and optimization-based estimates of stability domains for quadratic-bilinear OpInf ROMs using Lyapunov functions. The authors in [25] extended this work to OpInf models with nonlinear cubic terms, where the stability conditions are incorporated as constraints in the OpInf learning problem. The authors in [26] used regularization to obtain a stable OpInf ROM for rocket combustion, and the authors in [27] provided the generalized scalable algorithmic implementation of regularization to improve the robustness of OpInf. Maintaining physical consistency of ROMs can also be achieved through structure-preserving methods. In particular, embedding the underlying physical and mechanical structure into the OpInf framework has been extensively studied via structure-preserving OpInf for canonical Hamiltonian systems [28], for noncanonical Hamiltonian systems [29], for Lagrangian systems [30,31], for general systems that are conservative or dissipative in energy, and which exhibit a gradient structure [32], and for linear mechanical systems with second-order structure [33]. Lastly, the authors in [34] enforced the energy-preserving structure in the quadratic operator of OpInf through constrained optimization.

Physical systems often exhibit inherent bounds on their state variables, imposing crucial constraints that must be respected for accuracy in the FOM or ROM modeling. For example, species concentrations in chemical systems must remain non-negative and often have maximum possible concentrations due to constraints posed by reaction equations and boundary conditions. Fluid velocities in specific flow regimes cannot exceed a particular value (e.g., the speed of sound in the subsonic regime), and temperatures in thermodynamic systems may have lower or upper bounds depending on boundary conditions, etc. Violating these constraints, posed by fundamental physical laws of the system of interest, can lead to unphysical behavior of states in ROMs, poor prediction accuracy, and often model instability. Although the aforementioned OpInf ROM approaches conserve mathematical stability (e.g., Lyapunov), system energy, and mechanistic principles, they do not necessarily guarantee that the ROM predictions are consistent with the inherent physical bounds. In intrusive projection-based ROMs of reacting flows, the authors of [35] embedded state constraints on species concentrations to address unphysical oscillations observed in certain quantities and spatial regions and obtained stable ROMs within the training time interval. These oscillations were attributed to the ROM's sensitivity to the coupling between fluctuations in species mass fractions and temperature. To mitigate this issue, they used conditional probability density functions, which improved the prediction robustness. However, when reliable conditional PDFs between certain states are not available, this approach does not apply. Moreover, while previous OpInf formulations in [26,27,36,37] have successfully avoided stability issues related to unphysical predictions in a single-injector rocket combustion application, a general framework to systematically preserve inherent state bounds within the state prediction process is still lacking.

In this paper, we propose a new method to incorporate state constraints into a nonintrusive ROM, enabling it to produce physically consistent predictions for complex nonlinear systems. Our approach builds on the concept of species limiters introduced in [35] for intrusive projection-based ROMs of reacting flows, but generalizes it to the nonintrusive and predictive setting. The proposed methodology departs from [35] in three key aspects: (i) state constraints are explicitly enforced during the ROM prediction step via an online correction mechanism that partially reintroduces FOM information, thereby establishing a hybrid offline-online framework, (ii) the framework applies beyond the training regime, and (iii) it is general, i.e., it can be applied to any ROM that evolves low-dimensional states with known physical bounds of high-dimensional states—not solely to species concentrations. Beyond these three advances, we further propose a new way of selecting regularization hyperparameters in ROMs via application-specific key performance indicators (KPIs). While illustrated in this work using thermal energy in char combustion, this approach can be applied to other reacting flow problems or physical systems, allowing regularization to target the most relevant system metrics for accuracy and stability.

The rest of the paper is organized as follows. [Section 2](#) presents the theoretical and simulation background of char combustion. [Section 3](#) introduces the OpInf ROM with state constraints and also compares stabilizing effects provided by two different types of bases. [Section 4](#) provides numerical results, comparing the proposed method with the standard OpInf and other stability-enhancing approaches, and [Section 5](#) gives conclusions and an outlook to future work.

2. Char combustion application

A fluidized bed combustor is a combustion technology characterized by its efficient burning of fuels, such as char and biomass, with low pollutant emissions. It combines intricate processes characterized by a multiscale flow structure, the multiphysics associated with dense two-phase reactive flows, a nonlinear coupling of gas and solids, as well as complex homogeneous and heterogeneous chemical reactions involved [38,39]. We simulate char combustion with a numerical model via the particle-in-cell (PIC) method, using the open-source software MFiX-PIC (version 23.1.1) [40]. [Section 2.1](#) presents some theoretical background for the MFiX-PIC, briefly describing the governing equations for the gas and solid phase models alongside the associated chemical equations of the char combustion model. [Section 2.2](#) defines the computational domain, parameters, and initial and boundary conditions of the char combustion problem. [Section 2.3](#) presents details of the state variables in the MFiX-PIC data, and [Section 2.4](#) presents a convergence study of the char combustion simulation.

2.1. Governing equations of MFiX-PIC

In the PIC method, the term *particle* refers to an individual unit of material characterized by specific physical properties such as density, diameter, and chemical composition. The PIC method groups particles with the same physical properties as a computational parcel, a statistical assembly of particles. The PIC method solves a two-phase fluid problem by considering a Lagrangian solids phase model that tracks the trajectory of each computational parcel, not each particle, within an Eulerian fluid. Thus, it achieves considerable efficiency in computational costs compared to other methods that track each particle. Here, we briefly present the governing equations of the Eulerian gas phase model and the Lagrangian solids phase model that are solved in MFiX-PIC, as in [40,41].

2.1.1. Eulerian gas phase model

The Eulerian gas phase model is governed by conservation equations of the gas phase mass, i th species mass in the gas mixture, momentum, and internal energy, constituted mathematically as:

$$\frac{\partial}{\partial t}(\epsilon_g \rho_g) + \nabla \cdot (\epsilon_g \rho_g \mathbf{v}_g) = \sum_{i=1}^{N_g} R_{g,i} + S_g, \quad (1)$$

$$\frac{\partial}{\partial t}(\epsilon_g \rho_g Y_i) + \nabla \cdot (\epsilon_g \rho_g \mathbf{v}_g Y_i) = \nabla \cdot (\epsilon_g \mathbf{j}_g) + R_{g,i} + S_{g,i}, \quad (2)$$

$$\frac{\partial}{\partial t}(\epsilon_g \rho_g \mathbf{v}_g) + \nabla \cdot (\epsilon_g \rho_g \mathbf{v}_g \mathbf{v}_g) = -\nabla p_g + \nabla \cdot \boldsymbol{\tau}_g + \epsilon_g \rho_g \mathbf{g} + \mathbf{S}_g, \quad (3)$$

$$\epsilon_g \rho_g c_{p,g} \left[\frac{\partial T_g}{\partial t} + \nabla \cdot (\mathbf{v}_g T_g) \right] = \nabla \cdot (\epsilon_g \kappa \nabla T_g) + S_g. \quad (4)$$

In [Eq. \(2\)](#), the diffusive species mass flux vector \mathbf{j}_g is determined by Fick's law $\mathbf{j}_g = -\rho_g D_{g,i} \nabla Y_i$, where $D_{g,i}$ is the diffusion coefficient. The scalar S_g is a user-defined total gas-phase mass source [$\text{kg}/(\text{m}^3 \cdot \text{s})$], with $S_{g,i}$ denoting the contribution from i th species, so that $S_g = \sum_i S_{g,i}$, assuming all species contributions are accounted for. The vector \mathbf{S}_g is a general gas-phase momentum source [$\text{kg}/(\text{m}^2 \cdot \text{s}^2)$], accounting for interphase momentum exchange. In [Eq. \(3\)](#), the stress tensor $\boldsymbol{\tau}_g$ is modeled in the gas phase as Newtonian fluids with a linear relation between stress and rate of strain, $\boldsymbol{\tau}_g = -\frac{2}{3} \mu_g (\nabla \cdot \mathbf{v}_g) \boldsymbol{\delta} + \mu_g [(\nabla \mathbf{v}_g) + (\nabla \mathbf{v}_g)^\top]$. Here, μ_g is the gas dynamic viscosity [$\text{kg}/(\text{m} \cdot \text{s})$], and $\boldsymbol{\delta}$ represents the Kronecker delta tensor, characterized by elements $\delta_{j,k} = 0$, if $j \neq k$, and $\delta_{j,k} = 1$, if $j = k$. Additional notations are defined in the nomenclature in [Appendix A](#).

2.1.2. Lagrangian solids phase model

The Lagrangian solids phase model is governed by the conservation of mass, i th species mass, momentum, and internal energy for the p th parcel as:

$$\frac{d}{dt}(W_p m_p) = W_p \sum_{i=1}^{N_g} R_{p,i}, \quad (5)$$

$$\frac{d}{dt}(W_p m_p Y_{p,i}) = W_p R_{p,i}, \quad (6)$$

$$W_p m_p \frac{d\mathbf{v}_p}{dt} = W_p \left(m_p \mathbf{g} + \frac{m_p}{\epsilon_s \rho_s} \nabla \cdot \boldsymbol{\tau}_p \right), \quad (7)$$

$$W_p m_p c_{p,p} \frac{dT_p}{dt} = -W_p \left(\sum_{i=1}^{N_g} h_{p,i} R_{p,i} \right) + S_p. \quad (8)$$

The solid volume fraction is $\epsilon_s = 1 - \epsilon_g$, and ρ_s represents the solid density. Note that the statistical weight of particles W_p is considered in [Eqs. \(5\)–\(8\)](#).

2.1.3. Chemical reactions

In the char combustion, the heterogeneous gas-solid and homogeneous gas-gas chemistry are considered. First, the heterogeneous reaction occurs under the effect of char that reacts with oxygen, producing carbon monoxide as



This *char combustion* process, where char is burned as a fuel, consists of three distinct phases [42,43]. Initially, when a large char particle enters the bed, the reaction rate significantly exceeds the bulk diffusion rate. In this phase, bulk diffusion governs the combustion process. As combustion progresses, the char particle shrinks, making the diffusion rate comparable to the reaction rate. Lastly, the char particles become so small that the diffusion rate far exceeds the kinetic rate, making the kinetic reaction rate an influential factor in determining the combustion rate. The diffusion rate R_{diff} and the kinetic reaction rate R_{chem} are computed following [38,44] as

$$R_{\text{diff}} = \frac{24ShD_o}{d_p R_u T_m}, \quad R_{\text{chem}} = A_{\text{pre}} \exp\left(-\frac{E_0}{R_u T_p}\right).$$

Considering the interaction between R_{diff} and R_{chem} , the char combustion rate [kmol/s] across the aforementioned three phases is determined as

$$\frac{dm_{\text{ci}}}{dt} = -\pi d_{p,i}^2 p_{\text{oxy}} \left(\frac{1}{R_{\text{diff}}} + \frac{1}{R_{\text{chem}}} \right)^{-1}.$$

Here, the char particle diameter $d_{p,i}$ is computed as $\left(\frac{6m_p}{\pi\rho_p}\right)^{1/3}$, and the partial pressure of oxygen in the gas mixture p_{oxy} is computed as $p_g Y_{\text{O}_2} M_{\text{mix}}/M_{\text{O}_2}$. To compute the Sherwood number, we take the Frössling-type correlation for the mass transfer of particles in a fluidized bed as $Sh = 2.0\epsilon_g + 0.7(Re_g/\epsilon_g)^{1/2} \cdot Sc^{1/3}$, where the Reynolds number $Re_g = \rho_g \bar{U} d_p / \mu_g$, and the Schmidt number $Sc = \mu_g / (\rho_g D_o)$; see [45,46]. The homogeneous reaction is the CO combustion, governed by



The reaction rate of the CO combustion is modeled as in [47] as

$$r_{\text{CO}} = 3.98 \times 10^{14} \exp\left(-\frac{1.67 \times 10^5}{R_u T_g}\right) [c_{\text{CO}}][c_{\text{H}_2\text{O}}]^{0.5} [c_{\text{O}_2}]^{0.25}. \quad (11)$$

Note, since Eq. (11) uses c (molar concentration) with a unit of [mol/cm³], the resulting rate of CO combustion r_{CO} must be converted to be a unit consistent with the char combustion rate of [kmol/s].

2.2. Computational domain of the char combustion simulation

The computational domain of the fluidized bed boiler for char combustion is shown in Fig. 1. The small inlet dotted on the left wall of the boiler (yz -plane, with $x = 0$ [mm]) is fed with solid char that has a constant mass flow rate of 5.921×10^{-6} [kg/m], and the entire bottom inlet (xz -plane, with $y = 160$ [mm]) is fed with air. The boiler outlet A_o (xz -plane, with $y = 0$ [mm]) has a constant atmospheric pressure. The physical parameters and initial/boundary conditions used for the simulation are tabulated in Table 1. We set a slightly lower initial background temperature than the inflowing char temperature to obtain a transient thermal ramp-up during the initial few seconds of operation. Our simulation setup is based on the approaches presented in [48] and [44].

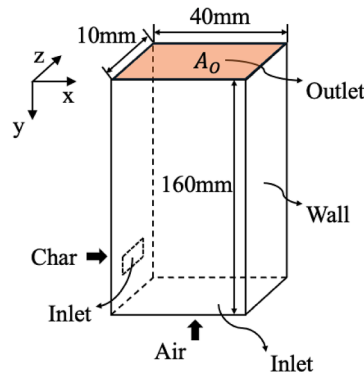


Fig. 1. Computational domain of a laboratory-scale fluidized bed for char combustion.

Table 1
Physical properties and parameters used for the char combustion simulation.

Parameters	Values
Dimensions, $x \times y \times z$ [mm]	$40 \times 160 \times 10$
Char inflow diameter [m]	4.906×10^{-4}
Char inflow density [kg/m ³]	1,300
Char mass flow rate [kg/s]	5.921×10^{-6}
Char inflow temperature [K]	300
Sand initial diameter	3.705×10^{-4}
Sand density [kg/m ³]	2,650
Initial background gas temperature [K]	1,073
Initial background solid temperature [K]	293.150
Bottom inlet gas pressure [Pa]	101,325
Bottom inlet gas temperature [K]	1,073
Bottom inlet gas y -velocity [m/s]	-0.873
Bottom inlet O ₂ mass fraction	0.230
Bottom inlet N ₂ mass fraction	0.760
Bottom inlet H ₂ O mass fraction	0.010
Outlet pressure [Pa]	101,325
Boundary wall temperature [K]	1,073

2.3. MFIX-PIC discretization details and dataset

To simulate the char combustion process, MFIX-PIC employs the finite volume method to spatially discretize the governing PDEs (1)–(8). This results in an N -dimensional system of nonlinear ODEs in terms of the full state $\mathbf{q} = \mathbf{q}(t) \in \mathbb{R}^N$, describing the time evolution of the n_v variables with

$$\frac{d\mathbf{q}}{dt} = \mathbf{F}(\mathbf{q}), \quad \mathbf{q}(t_0) = \mathbf{q}_0 \quad (12)$$

for $0 < t \leq T_f$. Due to the constant boundary conditions in the combustion model, as shown in Table 1, the nonlinear system represented by Eq. (12) is autonomous. We choose the $n_v = 10$ gas phase primitive variables— p_g, v_x, v_y, v_z, T_g , and Y_i , where $i = 1$ for O₂, $i = 2$ for N₂, $i = 3$ for CO, $i = 4$ for CO₂, and $i = 5$ for H₂O—as solution variables. In the time-discretized version of Eq. (12), the full state \mathbf{q}_k approximates $\mathbf{q}(t_k)$ for each time step k , and it includes all the solution vectors as:

$$\mathbf{q}_k = \left[\mathbf{p}_k^\top \ \mathbf{v}_{x,k}^\top \ \mathbf{v}_{y,k}^\top \ \mathbf{v}_{z,k}^\top \ \mathbf{T}_k^\top \ \mathbf{Y}_{\text{O}_2,k}^\top \ \mathbf{Y}_{\text{N}_2,k}^\top \ \mathbf{Y}_{\text{CO},k}^\top \ \mathbf{Y}_{\text{CO}_2,k}^\top \ \mathbf{Y}_{\text{H}_2\text{O},k}^\top \right]^\top \in \mathbb{R}^N. \quad (13)$$

Here, $N = n_v \cdot n_x$, and the bold symbol elements in \mathbf{q}_k indicate the vectorized solutions corresponding to each solution variable on the discretized n_x cells. Denote the ℓ th state vector within \mathbf{q}_k as

$$\mathbf{q}_k^{(\ell)} = \mathbf{q}_k[(\ell - 1)n_x : \ell n_x - 1], \quad \mathbf{q}_k^{(\ell)} \in \mathbb{R}^{n_x}. \quad (14)$$

Here, $\ell \in \{1, 2, \dots, n_v\}$, and $(\ell - 1)n_x : \ell n_x - 1$ indicates the range of indices pertaining to the ℓ th variable in the full state \mathbf{q}_k . The index ℓ in our case follows the order of variables in Eq. (13), e.g., $\ell = 1$ for pressure, $\ell = 2$ for x velocity, ..., and $\ell = 10$ for H₂O mass fraction. The training snapshot matrix is constructed as

$$\mathbf{Q} = [\mathbf{q}_0 \ \mathbf{q}_1 \ \dots \ \mathbf{q}_{K-1}] \in \mathbb{R}^{N \times K} \quad (15)$$

where the subscript of \mathbf{q} represents the time step and K denotes the number of snapshots in the training regime. Later, we denote each ℓ th state snapshot matrix as $\mathbf{Q}^{(\ell)} \in \mathbb{R}^{n_x \times K}$. This notation is frequently used in Section 3.1.

2.4. Convergence study of the char combustion simulation

We conduct a convergence study to verify that the numerical simulation of the char combustion provides reliable solutions that become independent of the spatial discretization parameter n_x . For the quantity of interest of the analysis, we choose the time-integrated species mass fraction that is averaged at the boiler outlet, denoted as \bar{Y} , of O₂ and CO₂, defined as

$$\bar{Y} = \sum_k \left(\frac{1}{|M|} \sum_{j \in M} \mathbf{Y}_{j,k} \right), \quad (16)$$

where $\mathbf{Y}_{j,k}$ denotes the mass fraction in the spatial index j and the temporal index k . Here, M is the set of spatial indices corresponding to the boiler outlet A_O (see Fig. 1), and $|M|$ is the cardinality of that set. The quantity of interest \bar{Y} provides a comprehensive understanding of the mass fractions within the region designated for thermal energy collection, integrated over time. We consider the simulation over a temporal interval ranging from 0 to 2.5s, after which the variables reach a quasi-steady state. Fig. 2 shows the quantity of interest versus the spatial discretization dimension n_x , as well as the computation time required, measured in CPU hours, for each n_x . The results for different mesh sizes are saved with the constant time step of 10^{-3} s, and the same final time is considered.

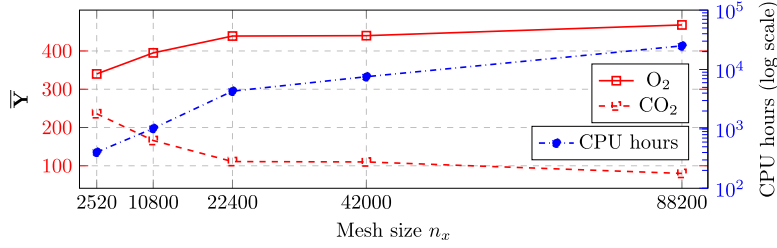


Fig. 2. Convergence study. Mesh sizes $n_x \in \{2520, 10800, 22400, 42000, 88200\}$ are evaluated. The left y-axis corresponds to integrated mass fractions for O_2 and CO_2 . The right y-axis corresponds to CPU hours on a logarithmic scale.

Hence, the difference in values of \bar{Y} arises only from the differences in the species mass fractions. For increasing mesh sizes n_x , the observed values of \bar{Y}_{O_2} are 340, 395, 439, 440, and 468, while the corresponding \bar{Y}_{CO_2} values are 236, 166, 111, 110, and 80. These results indicate that the integrated mass fractions of O_2 and CO_2 begin to converge at a spatial discretization of $n_x = 22,400$, making further mesh-refinement not cost-effective. Consequently, we select $n_x = 22,400$ as the spatial discretization dimension, leading to an overall ODE dimension of $N = n_v \cdot n_x = 224,000$. Nevertheless, as shown in Fig. 2, the FOM simulation with the chosen mesh size 22,400 still requires an extensive amount of computational cost, over 4,000 CPU hours, to obtain data for a physical time of 2.5s.

3. Operator inference reduced-order model with state constraints

The char combustion model is governed by the gas phase conservation Eqs. (1)–(4), the solid phase conservation Eqs. (5)–(8), and the chemical reactions (9)–(10). However, similar to other legacy codes or commercial software, the complexity in the MFiX-PIC source code renders the derivation of high-fidelity operators associated with the nonlinear function F in Eq. (12) impractical. For exactly those settings, OpInf has emerged as an efficient scientific machine learning technique to construct predictive ROMs of high-dimensional dynamical systems. In particular, OpInf solely requires high-fidelity data to develop underlying low-fidelity models by learning polynomial ROMs.

In Sections 3.1 and 3.2, we present data pre-processing and dimension reduction techniques essential for constructing ROMs of multiphysics, multiscale applications. Section 3.3 provides the theoretical background of the standard OpInf with regularization. In Section 3.4, we introduce state constraints and their implementation details in the OpInf ROM. We also illustrate the stabilizing effect of state constraints associated with the use of a global POD basis and block-structured POD basis.

3.1. Data pre-processing

The training and testing datasets are drawn from the FOM simulation data, with $t \in [2, 14]$ s used for training and $t \in [14, 40]$ s reserved for testing. The variables in the training dataset encompass a broad range of numerical scales, as shown in Table 2. Specifically, pressure and temperature are on the orders of 10^5 and 10^3 , respectively, whereas the mass fractions of certain species have magnitudes around 10^{-4} . Such multiphysics and multiscale problems pose substantial challenges for model training, as larger magnitude variables can dominate those with smaller scales. To avoid numerical issues, it is advantageous to pre-process the FOM data to scale all variables to be in comparable orders of magnitude. Among the state variables, pressure and temperature (including their mean values) have relatively large magnitudes among all the variables considered. To address this, we center each pressure and

Table 2

Statistics of state variables within the MFiX-PIC training dataset ($t \in [2, 14]$ s). The solutions on the 22,400 cells over 12,000 snapshots are considered for all $n_v = 10$ state variables.

Variable	Minimum	Mean	Maximum
Pressure [Pa]	9.876×10^4	1.014×10^5	2.600×10^5
x velocity [m/s]	-1.752×10	2.203×10^{-4}	1.742×10
y velocity [m/s]	-2.752×10	-9.470×10^{-1}	1.261×10
z velocity [m/s]	-1.507×10	1.888×10^{-4}	1.542×10
Temperature [K]	5.748×10^2	1.133×10^3	2.361×10^4
O_2 mass fraction	0	1.756×10^{-1}	2.300×10^{-1}
N_2 mass fraction	2.166×10^{-1}	7.455×10^{-1}	7.600×10^{-1}
CO mass fraction	0	8.933×10^{-4}	7.507×10^{-1}
CO_2 mass fraction	0	6.820×10^{-2}	3.799×10^{-1}
H_2O mass fraction	2.851×10^{-3}	9.810×10^{-3}	1.000×10^{-2}

temperature snapshot around zero via mean subtraction:

$$\mathbf{q}_k^{(\ell),\text{shifted}} = \mathbf{q}_k^{(\ell)} - \bar{\mathbf{q}}^{(\ell)}, \quad \bar{\mathbf{q}}^{(\ell)} = \frac{1}{K} \sum_{k=0}^{K-1} \mathbf{q}_k^{(\ell)}, \quad (17)$$

where ℓ indicates the state indices as in Eq. (14), $\mathbf{q}_k^{(\ell)} \in \mathbb{R}^{n_x}$ is the k th snapshot of the ℓ th state, and $\bar{\mathbf{q}}^{(\ell)} \in \mathbb{R}^{n_x}$ is its mean over K snapshots. We denote the shifted snapshot matrix as $\mathbf{Q}^{(\ell),\text{shifted}} \in \mathbb{R}^{n_x \times K}$, populated by the centered snapshots $\mathbf{q}_k^{(\ell),\text{shifted}}$, where $k = 0, 1, \dots, K-1$. In contrast, the x , y , and z velocities have relatively smaller magnitudes, with their means close to zero. Thus, we do not shift them. After shifting, if necessary, the pressure, temperature, and velocity snapshots are then scaled to ensure that all variables fall within the range of $[-1, 1]$. We scale variables by dividing all elements by the maximum absolute value of the entire shifted matrix of each variable $\mathbf{Q}^{(\ell),\text{shifted}}$. The scaling process results in the scaled snapshot matrix of the ℓ th variable as:

$$\mathbf{S}^{(\ell)} = \frac{\mathbf{Q}^{(\ell),\text{shifted}}}{\max(|\mathbf{Q}^{(\ell),\text{shifted}}|)} \in \mathbb{R}^{n_x \times K}. \quad (18)$$

We adopt this nondimensionalization approach from [26], where mapping variables to the range $[-1, 1]$ was chosen after careful study. This strategy has since become standard practice in subsequent OpInf studies, as it has been shown to improve numerical stability [27,37]. Note that since the mass fractions fall within the interval $[0, 1]$, we do not pre-process them. Specifically, $\mathbf{S}^{(\ell)} = \mathbf{Q}^{(\ell)} \in \mathbb{R}^{n_x \times K}$ for snapshots of species mass fractions, where $\ell = 6$ corresponds to O_2 mass fraction, $\ell = 7$ to N_2 mass fraction, $\ell = 8$ to CO mass fraction, $\ell = 9$ to CO_2 mass fraction, and $\ell = 10$ to H_2O mass fraction. After all pre-processing, the original training snapshot $\mathbf{Q} \in \mathbb{R}^{N \times K}$ is transformed into the scaled snapshot $\mathbf{S} \in \mathbb{R}^{N \times K}$.

3.2. Dimensionality reduction

Traditionally, OpInf adopts an approach in which snapshots of all state variables are aggregated, and subsequently, a global basis is derived via proper orthogonal decomposition (POD) from the comprehensive training snapshot matrix. We thus compute a global POD basis from the transformed snapshot matrix $\mathbf{S} \in \mathbb{R}^{N \times K}$ of the stacked variables. The reduced singular value decomposition (SVD) of \mathbf{S} , when $N \geq K$, is

$$\mathbf{S} = \mathbf{V}\mathbf{\Sigma}\mathbf{W}^T \quad (19)$$

where the columns of $\mathbf{V} \in \mathbb{R}^{N \times K}$ and $\mathbf{W} \in \mathbb{R}^{K \times K}$ are the left and right singular vectors of \mathbf{S} , respectively. The singular values of \mathbf{S} , denoted as $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_K \geq 0$, are on the diagonal of a diagonal matrix $\mathbf{\Sigma} \in \mathbb{R}^{K \times K}$. Note that the deterministic SVD of $\mathbf{S} \in \mathbb{R}^{N \times K}$ has a complexity of $O(N^2K + K^3)$, making it impractical due to the computational load that grows with the high degree-of-freedom N . Thus, we perform randomized SVD via the Python package `Scikit-learn` [49]. This significantly reduces the complexity to approximately $O(NK \log \omega + \omega^2(N + K))$, where ω is the target rank and is typically chosen such that $\omega \ll \min(N, K)$ [50]. The r leading columns of the left singular vectors \mathbf{V} form the rank- r linear POD basis as $\mathbf{V}_r = [\mathbf{v}_1, \dots, \mathbf{v}_r] \in \mathbb{R}^{N \times r}$ whose each column is orthonormal, i.e., $\mathbf{V}_r^T \mathbf{V}_r = \mathbf{I}_r \in \mathbb{R}^{r \times r}$. We project the scaled snapshot matrix \mathbf{S} as

$$\hat{\mathbf{S}} = \mathbf{V}_r^T \mathbf{S} = [\hat{\mathbf{s}}_0 \quad \dots \quad \hat{\mathbf{s}}_{K-1}] \in \mathbb{R}^{r \times K}. \quad (20)$$

The full-order scaled state $\mathbf{s}_k \in \mathbb{R}^N$ is approximated as $\mathbf{s}_k \approx \mathbf{V}_r \hat{\mathbf{s}}_k$. The low-dimensional snapshot matrix $\hat{\mathbf{S}}$ contains the data that are used to construct the ROM, which we describe next.

3.3. Operator inference with regularization

OpInf learns the best-fit polynomial ROM from the projected data $\hat{\mathbf{S}}$ in Eq. (20), potentially with added regularization or constraints. For the specific combustion problem considered in [37], including the cubic term in the right-hand side of the ROM provided a stabilizing effect. However, forming all possible cubic combinations of the reduced state vector requires $O(r^3)$ operations. Then multiplying the resultant vector by the reduced cubic operator $\hat{\mathbf{G}} \in \mathbb{R}^{r \times r^3}$ involves $O(r^4)$ multiplications and $O(r(r^3 - 1))$ additions. Thus, the overall complexity at every time step becomes $O(r^4)$ in an online ROM computation. Choosing a quadratic polynomial is often a good trade-off between model expressiveness and computational runtime for the ROM simulation [9–11,15,26,27,36]. Thus, we opt to learn a quadratic OpInf model as

$$\frac{d\hat{\mathbf{s}}}{dt} = \hat{\mathbf{c}} + \hat{\mathbf{A}}\hat{\mathbf{s}} + \hat{\mathbf{H}}(\hat{\mathbf{s}} \otimes \hat{\mathbf{s}}), \quad t \in [t_0, t_{K-1}]. \quad (21)$$

Here, the operator \otimes indicates a compact Kronecker product that calculates the element-wise multiplication of two vectors while removing redundant quadratic terms. For example, when $\mathbf{a} = [a_1, a_2, a_3]$, the compact Kronecker product is computed as $\mathbf{a} \otimes \mathbf{a} = [a_1^2, a_1 a_2, a_1 a_3, a_2^2, a_2 a_3, a_3^2]$. The $\hat{\mathbf{c}} \in \mathbb{R}^r$ is a constant term, $\hat{\mathbf{A}} \in \mathbb{R}^{r \times r}$ is a linear operator, $\hat{\mathbf{H}} \in \mathbb{R}^{r \times r(r+1)/2}$ is a quadratic operator. Note that a lifting transformation as in [26,36] is not used in this work. This is because directly accessing MFIX-PIC source terms is very involved, and the governing equations feature strong couplings between gas and solid phases, which makes the construction of an explicit lifted representation challenging.

In the model learning process, it is crucial to avoid overfitting the quadratic model to the data due to potential noise or under-resolution of the data, the truncated global POD modes that result in unresolved system dynamics, and the potential for misspecification of the ROM as fully quadratic, despite the highly nonlinear FOM encapsulated by the governing Eqs. (1)–(8) and the

chemical Eqs. (9)–(10). Therefore, we adopt Tikhonov regularization for OpInf, as in [27]. The quadratic OpInf with regularization learns the reduced operators $\hat{\mathbf{c}}$, $\hat{\mathbf{A}}$, and $\hat{\mathbf{H}}$ in the least-squares problem

$$\min_{\hat{\mathbf{c}}, \hat{\mathbf{A}}, \hat{\mathbf{H}}} \left\{ \sum_{k=0}^{K-1} \left\| \hat{\mathbf{c}} + \hat{\mathbf{A}}\hat{\mathbf{s}}_k + \hat{\mathbf{H}}(\hat{\mathbf{s}}_k \otimes \hat{\mathbf{s}}_k) - \dot{\hat{\mathbf{s}}}_k \right\|_2^2 + \lambda_1 \left(\|\hat{\mathbf{c}}\|_2^2 + \|\hat{\mathbf{A}}\|_F^2 \right) + \lambda_2 \|\hat{\mathbf{H}}\|_F^2 \right\}, \quad (22)$$

where $\hat{\mathbf{s}}_k \equiv \hat{\mathbf{s}}(t_k)$ represents the reduced state at the time step k . Note that the Kronecker product introduces scaling differences between the entries of the quadratic operator $\hat{\mathbf{H}}$, and those in the constant operator $\hat{\mathbf{c}}$ and the linear operator $\hat{\mathbf{A}}$, when subjected to a single regularization parameter. This motivates us to use two regularization parameters $\lambda_1 > 0$ and $\lambda_2 > 0$, such that λ_1 penalizes $\hat{\mathbf{c}}$ and $\hat{\mathbf{A}}$, and λ_2 penalizes $\hat{\mathbf{H}}$. Here, $\dot{\hat{\mathbf{s}}}_k \equiv \dot{\hat{\mathbf{s}}}(t_k)$ denotes the time derivative of the reduced state at the time step k , and it can be computed using a suitable time derivative approximation scheme. We use a 6th-order central finite difference method, with biased forward and backward differencing applied at the initial and final time steps. Equation (22) can be written compactly as

$$\min_{\mathbf{O}} \left\| \mathbf{D}\mathbf{O}^\top - \hat{\mathbf{S}}^\top \right\|_F^2 + \|\mathbf{A}\mathbf{O}^\top\|_F^2, \quad \text{where } \mathbf{O} = \begin{bmatrix} \hat{\mathbf{c}} & \hat{\mathbf{A}} & \hat{\mathbf{H}} \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} \mathbf{1}_K & \hat{\mathbf{S}}^\top & (\hat{\mathbf{S}} \otimes \hat{\mathbf{S}})^\top \end{bmatrix}, \quad (23)$$

and where $\mathbf{O} \in \mathbb{R}^{r \times d(r)}$ and $\mathbf{D} \in \mathbb{R}^{K \times d(r)}$. Here, $\hat{\mathbf{S}} \in \mathbb{R}^{r \times K}$ is a matrix whose columns are $\hat{\mathbf{s}}_k$, and $\mathbf{1}_K \in \mathbb{R}^K$ is a column vector of length K with all entries set to one. Also, $\mathbf{A} = \text{diag}(\lambda_1, \lambda_1 \mathbf{I}_r, \lambda_2 \mathbf{I}_{r(r+1)/2}) \in \mathbb{R}^{d(r) \times d(r)}$ serves as a diagonal regularization matrix, where \mathbf{I}_r and $\mathbf{I}_{r(r+1)/2}$ are identity matrices with dimensions r and $r(r+1)/2$, respectively, making $d(r) = 1 + r + r(r+1)/2$. In practice, we must carefully choose the regularization parameters λ_1 and λ_2 ; see Section 4.3 for details. The reduced operators inferred by solving Eq. (23) provide the best fits of reduced state trajectories of the ROM Eq. (21) within the training regime in a minimum residual sense.

3.4. State constraints to enforce physical consistency of the operator inference reduced-order model

Low-dimensional surrogate modeling is challenging for nonlinear multiscale and multiphysics problems. Consequently, there have been many research advances to ensure better predictive capabilities of these models. As discussed in Section 1, various structure-preserving methods offer an approach to enforce physical consistency in OpInf ROMs. While these approaches have successfully addressed issues of mathematical stability, energy preservation, and conservation of mechanistic principles, they are not guaranteed to provide ROM predictions that are consistent with the physical constraints inherent in the FOM. In this section, we propose a method of maintaining these intrinsic physical constraints within the state prediction via OpInf ROMs.

3.4.1. Limitation of the standard operator inference reduced-order model

The standard OpInf propagates states in the low-dimensional space from the initial reduced state $\hat{\mathbf{s}}_0$ to the final reduced state $\hat{\mathbf{s}}_{K_f-1}$ through a numerical time integration. Here, K_f represents the total number of snapshots forecasted through OpInf beyond the training regime. This is illustrated in the dashed arrows in Fig. 3, which highlights the contrast in state propagation when using standard OpInf compared to the proposed scheme with state constraints. Once the temporal evolution of all the reduced states $\hat{\mathbf{s}}_k$ is done, the high-dimensional states are reconstructed by $\tilde{\mathbf{S}} = \mathbf{V}_r \hat{\mathbf{S}} \in \mathbb{R}^{N \times K_f}$. However, the reconstructed high-dimensional state $\tilde{\mathbf{s}}_k$ is not guaranteed to comply with the potential constraints posed by the physics of the FOM. Particularly for the char combustion data considered herein, some predicted species mass fractions, which are in $\tilde{\mathbf{S}}^{(\ell)}$, where $\ell = 6, 7, 8, 9, 10$, see Section 3.1, may not fall within the range $[0, 1]$. In fact, even the reconstruction of the initial reduced state, which is $\mathbf{V}_r \hat{\mathbf{s}}_0$, is not assured to provide mass fractions consistent with such a constraint.

3.4.2. Design of state constraints

In the proposed framework of OpInf ROM with state constraints, we enforce the state constraints in the reconstructed high-dimensional state $\tilde{\mathbf{s}}_k \in \mathbb{R}^N$ at every time step during the state propagation. Since the available FOM data of the char combustion does not provide sufficient information to determine whether pressure, temperature, and velocities inherit physical constraints, we limit our state constraints to the species mass fraction variables. This is problem-specific; thus, if physical limitations regarding other variables are available from prior knowledge of the FOM, different constraints may have to be applied. By embedding state constraints for the species mass fraction variables, we aim to enhance the physical consistency of the predictions made by the OpInf ROM, thereby improving its ability to stably and accurately capture the complex dynamics of char combustion processes while maintaining computational efficiency.

To design state constraints on species mass fractions, we use the knowledge of the variable ranges as well as boundary conditions of the FOM, as shown in Table 1, and chemical reactions (9)–(10). Concerning the air inflow boundary conditions of the combustion model, the maximum permissible values of the mass fractions of O_2 , N_2 , and H_2O are dictated by the corresponding constituents within the air inflow, given that these components are not produced from both Eqs. (9)–(10). We denote those upper bounds as Y^U . Those for O_2 , N_2 , and H_2O are $Y_{\text{O}_2}^U = 0.23$, $Y_{\text{N}_2}^U = 0.76$, and $Y_{\text{H}_2\text{O}}^U = 0.01$, see Table 2. However, the CO and CO_2 do not constitute the inlet air inflow and are the products of the two chemical reactions considered. Thus, we set Y_{CO}^U and $Y_{\text{CO}_2}^U$ to 1, the possible maximum value of species mass fractions. Moreover, mass fractions in the FOM data, and in general, are always non-negative, see Table 2 for the range of FOM states in the training regime. Thus, we enforce a lower bound of zero for all species mass fractions ($Y_{\text{O}_2}^L = Y_{\text{N}_2}^L = Y_{\text{CO}}^L = Y_{\text{CO}_2}^L = Y_{\text{H}_2\text{O}}^L = 0$) throughout both the training and testing regimes. In sum, we formulate the state constraints of

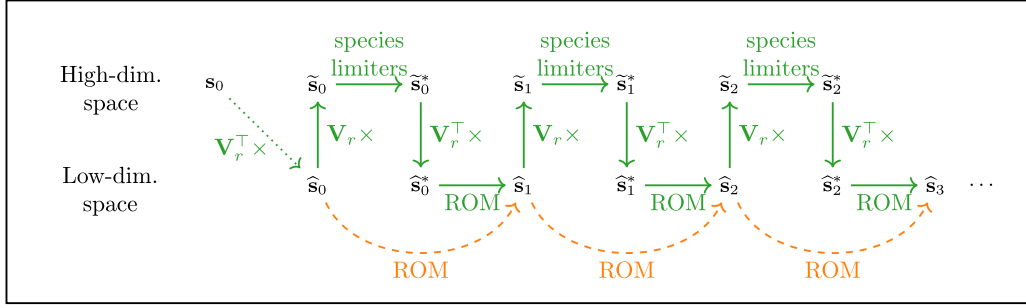


Fig. 3. Schematic of the state propagation in the ROM with species limiters, which are state constraints for the species mass fractions. The dashed arrows represent the reduced state propagation via the standard OpInf ROM, and the solid arrows represent the state propagation via the OpInf with species limiters. The orthogonal basis $\mathbf{V}_r \in \mathbb{R}^{N \times r}$ is used for projection between the low-dimensional space \mathbb{R}^r and the high-dimensional space \mathbb{R}^N .

all species mass fractions as

$$Y_{O_2} \leftarrow \max \left(\min \left(Y_{O_2}, Y_{O_2}^U \right), Y_{O_2}^L \right), \quad (24)$$

$$Y_{N_2} \leftarrow \max \left(\min \left(Y_{N_2}, Y_{N_2}^L \right), Y_{N_2}^L \right), \quad (25)$$

$$Y_{CO} \leftarrow \max \left(\min \left(Y_{CO}, Y_{CO}^U \right), Y_{CO}^L \right), \quad (26)$$

$$Y_{CO_2} \leftarrow \max \left(\min \left(Y_{CO_2}, Y_{CO_2}^U \right), Y_{CO_2}^L \right), \quad (27)$$

$$Y_{H_2O} \leftarrow \max \left(\min \left(Y_{H_2O}, Y_{H_2O}^U \right), Y_{H_2O}^L \right). \quad (28)$$

3.4.3. ROM state propagation with state constraints

In Fig. 3, we illustrate that states propagate across two spaces within the OpInf ROM with species limiters. We start from the true scaled high-dimensional state s_0 and project it to the initial reduced state \hat{s}_0 . To maintain the physical consistency from the initial time step, the initial reduced state \hat{s}_0 first must be lifted to the high-dimensional state \tilde{s}_0 that contains all variables. The state constraints, formulated as Eqs. (24)–(28), are directly enforced to \tilde{s}_0 , leading to the modified high-dimensional state \tilde{s}_0^* . This subsequently needs to be projected back to \hat{s}_0^* , so that the ROM propagates \hat{s}_0^* to \hat{s}_1 . In this manner, we propagate states over time within the low-dimensional space, thereby maintaining computational efficiency. This iterative procedure continues until the final time. Once the entire temporal propagation is done, we unscale the modified high-dimensional states \tilde{s}_k^* ($k = 0, 1, \dots, K_f - 1$), which contain scaled variables such as pressure, temperature, and velocities, via the inversion of Eqs. (17)–(18), obtaining the high-dimensional states in the original scale \tilde{q}_k^* . Note that since we only constrain the species mass fraction variables, which are not scaled initially, we obviate the necessity to restore the original scales of \tilde{s}_k at every time step during the state propagation. Nevertheless, should limiting conditions pertain to pre-processed variables, restoring their original scales in the high-dimensional space becomes necessary at each time step prior to enforcing state constraints.

The presented approach departs from the conventional offline-online decomposition in model reduction by incorporating online evaluations that partially reintroduce FOM information through state corrections. While the reduced dynamics evolve efficiently via a ROM, the correction step can be interpreted as providing a *closure mechanism* that enforces physical constraints the ROM alone may violate. This establishes a hybrid simulation: the ROM captures and evolves dominant dynamics efficiently, and the FOM-informed correction ensures physicality. Algorithm 1 provides the algorithmic process of OpInf with state constraints. Note that the model learning process, including data pre-processing, SVD, and regularization, requires K training snapshots, and the temporal propagation with state constraints can go beyond this training regime, to K_f time steps.

3.4.4. Stabilizing effects of ROM with state constraints and a global basis

The species limiters that act on a part of the state have a stabilizing effect on the entire state vector. This stabilizing effect is further reinforced by the global POD basis, which plays a key role in enhancing overall stability. Fig. 4 provides a graphical illustration of the state propagation, embedded with state constraints.

Since we use the global POD basis computed from stacked state snapshots, see Section 3.2, the linear combination of the r columns of $\mathbf{V}_r[(\ell - 1)n_x : \ell n_x - 1, :] \in \mathbb{R}^{n_x \times r}$ and the r entries of \tilde{s}_k reconstructs each ℓ th state vector, which is denoted as $\tilde{s}_k[(\ell - 1)n_x : \ell n_x - 1]$. For example, the reconstructed scaled pressure vector at time step k , $\tilde{\mathbf{p}}_k^{\text{scaled}} \in \mathbb{R}^{n_x}$, is computed as a linear combination of the columns of \mathbf{V}_{p_g} and the entries of \hat{s}_k , where \mathbf{V}_{p_g} is the first n_x rows of the global basis $\mathbf{V}_r \in \mathbb{R}^{N \times r}$. In other words, when $\mathbf{V}_{p_g} = [\mathbf{v}_{p_g,1}, \dots, \mathbf{v}_{p_g,r}] \in \mathbb{R}^{n_x \times r}$, $\tilde{\mathbf{p}}_k^{\text{scaled}}$ is computed as $\sum_{\eta=1}^r \hat{s}_{k,\eta} \cdot \mathbf{v}_{p_g,\eta}$. After all the states are reconstructed, leading to \tilde{s}_k , we enforce the state constraints to the mass fraction states. Thus, $\tilde{Y}_{O_2,k}$, $\tilde{Y}_{N_2,k}$, $\tilde{Y}_{CO,k}$, $\tilde{Y}_{CO_2,k}$, and $\tilde{Y}_{H_2O,k}$ are corrected, producing a modified high-dimensional state \tilde{s}_k^* . In Fig. 4, we indicate the modified states with a shaded area. Afterward, we compress \tilde{s}_k^* through the global POD basis \mathbf{V}_r , resulting in the low-dimensional state \hat{s}_k^* . We can view the numerical aspect of this projection in two ways. First, it is

Algorithm 1 Operator inference with state constraints.

- 1: **procedure** OPINF WITH STATE CONSTRAINTS
- 2: Transform original snapshots $\mathbf{Q} \in \mathbb{R}^{N \times K}$ into $\mathbf{S} \in \mathbb{R}^{N \times K}$ as in Eqs. (17)– (18)
- 3: Compute randomized SVD with a target rank r : $\mathbf{S} = \mathbf{V}_{r_t} \boldsymbol{\Sigma}_{r_t} \mathbf{W}_{r_t}^\top$ where $\mathbf{V}_{r_t} \in \mathbb{R}^{N \times r_t}$, $\boldsymbol{\Sigma}_{r_t} \in \mathbb{R}^{r_t \times r_t}$, and $\mathbf{W}_{r_t} \in \mathbb{R}^{K \times r_t}$
- 4: Select a rank- r low-dimensional basis $\mathbf{V}_r = \mathbf{V}_{r_t}[:, :r]$
- 5: Project \mathbf{S} to the linear low-dimensional subspace: $\hat{\mathbf{S}} = \mathbf{V}_r^\top \mathbf{S} \in \mathbb{R}^{r \times K}$
- 6: Evaluate the time derivative of the reduced snapshot $\hat{\mathbf{S}}$
- 7: **procedure** REGULARIZATION
- 8: Search hyperparameters λ_1^* and λ_2^* based on the key performance indicator criterion as in Eq. (32)
- 9: **end procedure**
- 10: **procedure** TIME PROPAGATION OVER K_f TIME STEPS WITH STATE CONSTRAINTS
- 11: **for** $k = 0, \dots, K_f - 1$ **do**
- 12: Reconstruct the high-dimensional state: $\tilde{\mathbf{s}}_k = \mathbf{V}_r \hat{\mathbf{s}}_k \in \mathbb{R}^N$
- 13: Apply the state constraints Eqs. (24)–(28), to $\tilde{\mathbf{s}}_k$, obtaining a modified full-state $\tilde{\mathbf{s}}_k^*$
- 14: Project $\tilde{\mathbf{s}}_k^*$ to the low-dimensional subspace: $\hat{\mathbf{s}}_k^* = \mathbf{V}_r^\top \tilde{\mathbf{s}}_k^*$
- 15: Propagate from $\hat{\mathbf{s}}_k^*$ to $\hat{\mathbf{s}}_{k+1}$ by time integration of Eq. (21)
- 16: **end for**
- 17: **end procedure**
- 18: Restore high-dimensional states $\tilde{\mathbf{q}}_k^*$ in the original scales from $\hat{\mathbf{s}}_k^*$
- 19: **return** $\mathbf{Q}^* = [\tilde{\mathbf{q}}_0^* \quad \tilde{\mathbf{q}}_1^* \quad \dots \quad \tilde{\mathbf{q}}_{K_f-1}^*] \in \mathbb{R}^{N \times K_f}$
- 20: **end procedure**

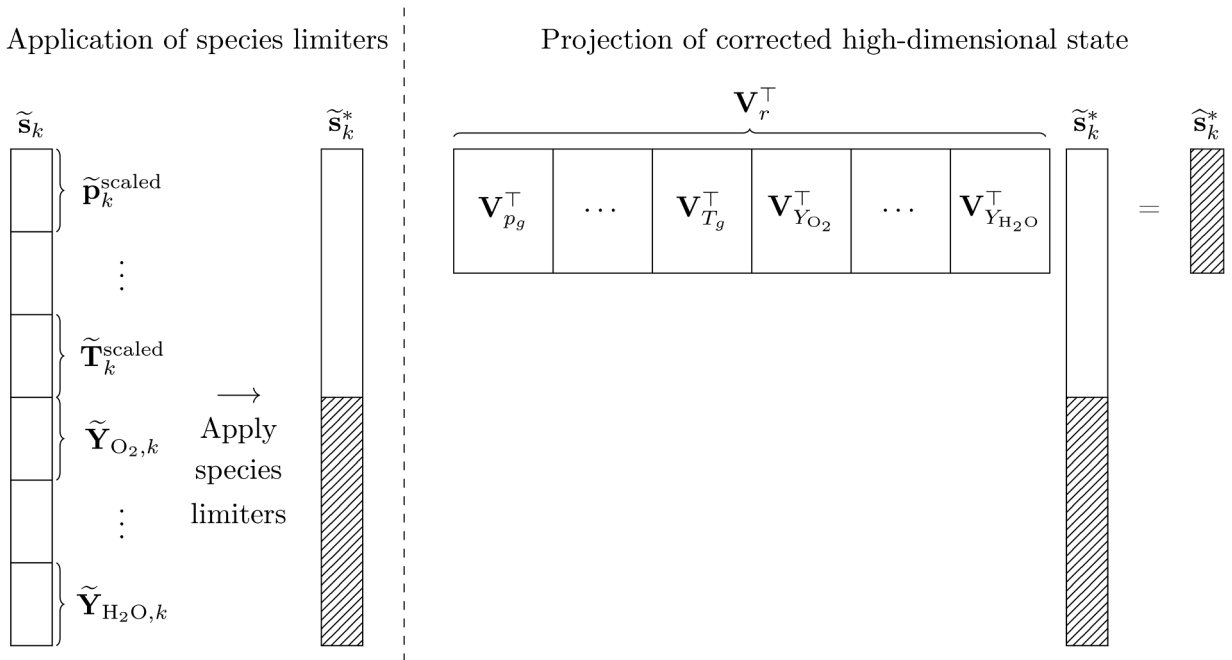


Fig. 4. Interpretation of the state constraints-embedded state propagation, as presented in Fig. 3. The process is illustrated for a single time step. The left-hand side of the vertical dotted line represents the enforcement of species limiters, and the shaded area of $\tilde{\mathbf{s}}_k^*$ represents the corrected values. The right-hand side of the vertical dotted line represents the projection of the partially corrected high-dimensional state, modifying all the entries of the reduced state $\hat{\mathbf{s}}_k^*$.

a linear combination of all the columns of \mathbf{V}_r^\top and all the entries of $\tilde{\mathbf{s}}_k^*$, meaning $\hat{\mathbf{s}}_k^* = \sum_{n=1}^N \tilde{\mathbf{s}}_{k,n}^* \cdot \mathbf{V}_r^\top[:, n]$. Another is that each entry of $\hat{\mathbf{s}}_k^*$, denoted as $\hat{s}_{k,\eta}^*$ ($\eta = 1, 2, \dots, r$), is the outer product between a row vector of \mathbf{V}_r^\top , which is $\mathbf{V}_r^\top[\eta, :]$, and a column vector $\tilde{\mathbf{s}}_k^*$. Either way, all the entries of the resultant low-dimensional state $\hat{\mathbf{s}}_k^*$ are modified. Because the ROM advances the reduced states as a coupled system, these changes propagate to all variables, providing an additional source of stabilization. Consequently, in the next time step $k + 1$, even the reconstructed high-dimensional states not directly constrained by species limiters are affected. Thus, the proposed approach benefits from two complementary stabilizing mechanisms: (i) the state constraints when using the global POD basis, and

(ii) the inherent coupling of the ROM dynamics. In our case, these combined effects stabilize the evolution of all state variables, as further demonstrated in the numerical results in Section 4.4.4.

3.4.5. ROM prediction with state constraints and a block-structured basis

Constructing an individual basis for each variable, rather than computing a stacked global POD basis from the entire training snapshot matrix $\mathbf{S} \in \mathbb{R}^{N \times K}$, as illustrated in the previous section, leads to more interpretable ROM coefficients. This approach is discussed in detail in [51, Sec. 4.3.2], where the authors used a block-diagonal projection matrix to address the topological structure of a highly coupled nonlinear problem. Specifically in our case, we can compute a separate basis for the training snapshot matrix of the ℓ th state variable, $\mathbf{S}^{(\ell)} \in \mathbb{R}^{n_x \times K}$ (see Eq. (18)). When $n_x \geq K$, the reduced SVD of $\mathbf{S}^{(\ell)}$ is given by

$$\mathbf{S}^{(\ell)} = \mathbf{V}^{(\ell)} \mathbf{\Sigma}^{(\ell)} \mathbf{W}^{(\ell)\top},$$

where $\mathbf{V}^{(\ell)} \in \mathbb{R}^{n_x \times K}$, $\mathbf{\Sigma}^{(\ell)} \in \mathbb{R}^{K \times K}$, and $\mathbf{W}^{(\ell)} \in \mathbb{R}^{K \times K}$. The linear POD basis for the ℓ th state variable is obtained as the leading $r^{(\ell)}$ columns of the left singular vector matrix $\mathbf{V}^{(\ell)}$, denoted as $\mathbf{V}_r^{(\ell)} \in \mathbb{R}^{n_x \times r^{(\ell)}}$. Each reduction dimension $r^{(\ell)}$ is chosen separately for each state.

After computing the individual bases for all state variables, each basis is used to project its corresponding high-dimensional state into a reduced state. To clarify, referring to Eq. (13), we define the bases for all variables as follows: $\mathbf{V}_{p,r} = \mathbf{V}_r^{(\ell=1)}$ for pressure, $\mathbf{V}_{v_x,r} = \mathbf{V}_r^{(\ell=2)}$ for x -velocity, ..., $\mathbf{V}_{Y_{H_2O},r} = \mathbf{V}_r^{(\ell=10)}$ for H_2O mass fraction. Using these bases, the high-dimensional states are projected into their reduced states: $\hat{\mathbf{s}}_{p,k} = \mathbf{V}_{p,r}^\top \mathbf{p}_k^{\text{scaled}} \in \mathbb{R}^{r^{(1)}}$ for pressure, $\hat{\mathbf{s}}_{v_x,k} = \mathbf{V}_{v_x,r}^\top \mathbf{v}_{x,k}^{\text{scaled}} \in \mathbb{R}^{r^{(2)}}$ for x -velocity, ..., $\hat{\mathbf{s}}_{Y_{H_2O},k} = \mathbf{V}_{Y_{H_2O},r}^\top \mathbf{Y}_{H_2O,k} \in \mathbb{R}^{r^{(10)}}$ for H_2O mass fraction. Since each variable is projected differently, we construct a block-structured basis

$$\mathbf{V}_r = \text{diag}(\mathbf{V}_{p,r}, \mathbf{V}_{v_x,r}, \dots, \mathbf{V}_{Y_{H_2O},r}) \in \mathbb{R}^{N \times r},$$

where $r = \sum_{\ell=1}^{10} r^{(\ell)}$. This basis projects the entire full state \mathbf{s}_k into the reduced state $\hat{\mathbf{s}}_k \in \mathbb{R}^r$ via $\hat{\mathbf{s}}_k = \mathbf{V}_r^\top \mathbf{s}_k$. The variables remain decoupled when reconstructing the full state $\tilde{\mathbf{s}}_k$ from the entire reduced state $\hat{\mathbf{s}}_k$ through the block-structured basis \mathbf{V}_r as

$$\tilde{\mathbf{s}}_k = \begin{bmatrix} \tilde{\mathbf{p}}_k^{\text{scaled}} \\ \tilde{\mathbf{v}}_{x,k}^{\text{scaled}} \\ \vdots \\ \tilde{\mathbf{Y}}_{H_2O,k} \end{bmatrix} \approx \begin{bmatrix} \mathbf{V}_{p,r} & & & \\ & \mathbf{V}_{v_x,r} & & \\ & & \ddots & \\ & & & \mathbf{V}_{Y_{H_2O},r} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{s}}_{p,k} \\ \hat{\mathbf{s}}_{v_x,k} \\ \vdots \\ \hat{\mathbf{s}}_{Y_{H_2O},k} \end{bmatrix}. \tag{29}$$

We next discuss the effect of using a block-structured projection matrix in the context of state constraints. Fig. 5 illustrates the projection of the corrected high-dimensional states $\tilde{\mathbf{s}}_k^*$ using the block-structured POD basis, resulting in the modified reduced states $\hat{\mathbf{s}}_k^*$. Similar to Fig. 4, the shaded areas in $\tilde{\mathbf{s}}_k^*$ represent the parts corrected by the state constraints on mass fractions. Because of the

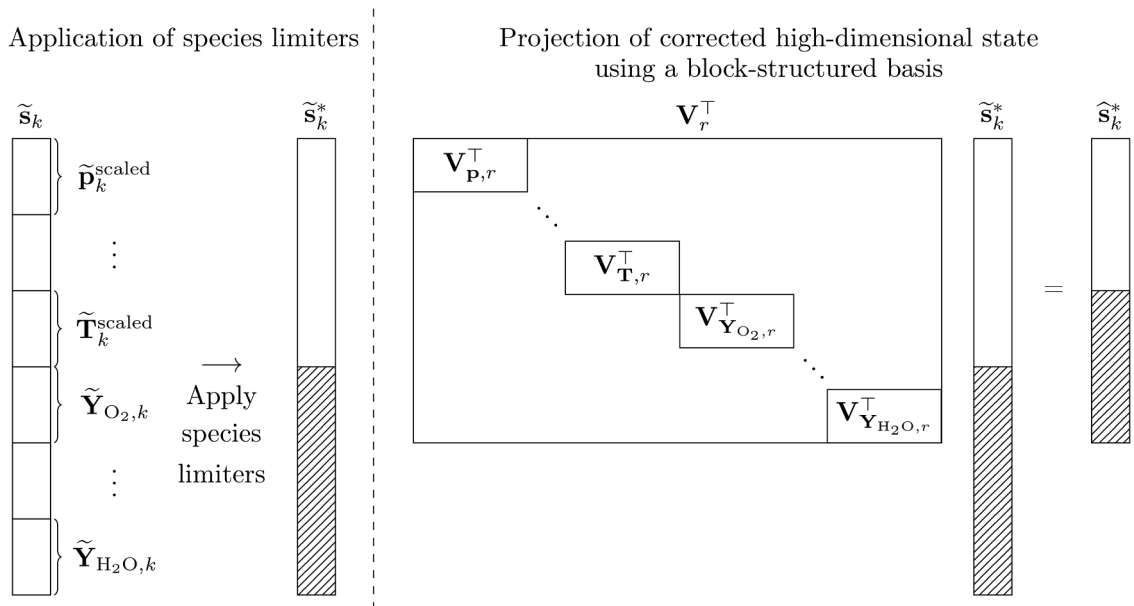


Fig. 5. Interpretation of state-constraints-embedded state propagation, when using a block-structured POD basis. The left-hand side of the vertical dotted line represents the enforcement of species limiters, and the shaded area of $\tilde{\mathbf{s}}_k^*$ represents the corrected values. The right-hand side shows the projection of the partially corrected high-dimensional state, modifying only the entries of the reduced state $\hat{\mathbf{s}}_k^*$ corresponding to the mass fraction variables.

block structure of \mathbf{V}_r^\top , the coefficients in the reduced states $\hat{\mathbf{s}}_k^*$ associated with variables other than species mass fractions remain unchanged. In particular, the first $\sum_{\ell=1}^5 r^{(\ell)}$ entries of $\hat{\mathbf{s}}_k^*$ (corresponding to unmodified states such as pressure, temperature, velocities in our case) are unaffected by the state constraints. However, once the ROM evolves the *partially modified* reduced state $\hat{\mathbf{s}}_k^*$ in time, its coupled dynamics cause all state variables to be indirectly affected. Consequently, similar to the case of the global basis, at the next time step $k + 1$, the components of the reduced state $\hat{\mathbf{s}}_k^*$ for pressure, temperature, and velocities differ from those that would have been obtained without mass-fraction correction at time step k . In other words, with a block-structured basis, state constraints still provide a stabilizing effect on all variables through the coupled ROM dynamics, though not as strongly as in the global basis case, which has two stabilizing mechanisms, as discussed in Section 3.4.4.

4. Numerical results

This section presents the numerical results of our study on OpInf ROMs with state constraints for the char combustion problem. Section 4.1 presents the details of the MFIX-PIC FOM data from which the ROM is constructed. In Section 4.2, we discuss the ROM dimension selection process. Section 4.3 presents the regularization hyperparameters selection process, where we propose a new selection criterion via a key performance indicator. In Section 4.4, we analyze the effect of enforcing state constraints to OpInf on stability, accuracy, and computational efficiency by comparing it with standard OpInf and other stability-enhancing OpInf approaches. As discussed in Sections 3.4.4 and 3.4.5, the global basis provides stronger stabilizing effects due to the combination of state constraints and coupled ROM dynamics. Thus, for the results shown here, we use the global POD basis.

4.1. FOM data details

The training data generated by MFIX-PIC consist of solution variables from Eq. (13) over $n_x = 22,400$ cells, with $N = n_v \times n_x = 224,000$ being the degrees of freedom of the high-fidelity system. The FOM simulation generates solutions over 38s (from $t = 2$ to $t = 40$ s), using an adaptive time step ranging from 10^{-8} s to 10^{-4} s. The snapshots are saved every $\delta t = 10^{-3}$ s, providing $K_f = 38,000$ snapshots. Of these, the initial $K = 12,000$ snapshots (corresponding to $t \in [2, 14]$ s, 32% of the data) are used for training, while the remaining 26,000 snapshots (corresponding to $t \in [14, 40]$ s, 68% of the data) are reserved for testing. Thus, the testing interval extends approximately 200% beyond the length of the training interval. The high-dimensional data generation took approximately 56 days with an MPI-based solver on 45 CPUs, corresponding to 60,480 CPU hours (Intel Xeon W-3175X CPU @ 3.10 GHz). The construction of the OpInf ROM utilizes the training snapshot matrix $\mathbf{Q} \in \mathbb{R}^{224,000 \times 12,000}$. Table 2 shows the range of state variables within the training regime, which is essential for the data pre-processing step, as elaborated in Section 3.1. The ranges and means of the state variables in the testing regime (not shown here) are similar to those in the training regime.

4.2. Selection of the ROM dimension

Conventionally, POD considers an energy metric to determine the ROM dimension r . The cumulative energy $\mathcal{E}_{\text{cum}}(r)$ is defined as the ratio of the sum of squared singular values up to rank r to the total sum of squared singular values: $\mathcal{E}_{\text{cum}}(r) = \sum_{\eta=1}^r \sigma_\eta^2 / \sum_{\eta=1}^{r_t} \sigma_\eta^2$, where the σ_η ($\eta = 1, 2, \dots, r_t$) are the singular values of the scaled snapshot matrix \mathbf{S} . Here, r_t is the number of singular values to extract when we use randomized SVD in practice. This cumulative energy connects the low-rank dimension r to the relative projection error of the scaled snapshot \mathbf{S} as

$$\mathcal{E}_{\text{proj}}(r) = \frac{\|\mathbf{S} - \mathbf{V}_r \mathbf{V}_r^\top \mathbf{S}\|_F^2}{\|\mathbf{S}\|_F^2} = 1 - \mathcal{E}_{\text{cum}}(r). \quad (30)$$

For this data set, we found that $\mathcal{E}_{\text{proj}}(1) = 0.005$, meaning the cumulative energy is 0.995 with $r = 1$. Despite the first mode containing most of the system energy, choosing only one global mode for all variables is insufficient to capture the complex dynamics and sensitivities of the combustion model.

Instead of evaluating the single projection error of the scaled snapshot matrix \mathbf{S} that contains all the scaled variables, we can also calculate the projection error for each variable in its original scales, with respect to the low-dimensional rank- r basis \mathbf{V}_r . This involves reconstructing the scaled snapshots as $\tilde{\mathbf{S}} = \mathbf{V}_r \mathbf{V}_r^\top \mathbf{S}$, followed by restoring the original scales of all $n_v = 10$ variables through the inversion of Eqs. (17)–(18). Denoting the reconstructed snapshot matrix of the ℓ th state in its original scale as $\tilde{\mathbf{Q}}^{(\ell)} \in \mathbb{R}^{n_x \times K}$ (see Section 3.1 for notation), the relative error is computed with respect to the true ℓ th state snapshot matrix $\mathbf{Q}^{(\ell)} \in \mathbb{R}^{n_x \times K}$.

Fig. 6 shows the variable-specific projection errors in their original scales with respect to the low-rank dimension r . Pressure, N_2 , and H_2O mass fractions exhibit notably low errors, in turn indicating high energy capture. Conversely, CO mass fraction and velocity components (v_x , v_y , v_z) display higher errors. This discrepancy is in part due to their relatively small scales, which skew the errors when considered in the denominator of the relative error metric. These observations highlight the strong dependence of projection errors on the specific variable under consideration. This variability suggests that the effectiveness of the rank- r basis in capturing the system's dynamics is not uniform across all variables.

In conclusion, neither the projection error of the individual variables nor the energy associated with the entire scaled snapshot matrix \mathbf{S} provides a comprehensive representation of the complex and multiscale characteristics of this dataset. After several tests, we select $r = 30$ because it reasonably captures the high-frequency fluctuating dynamics of the FOM while providing a manageable model dimension. For this dataset, selecting more than 30 modes is ineffective. This is because the regularization hyperparameter selection process, which is discussed next, substantially increases the computational time.

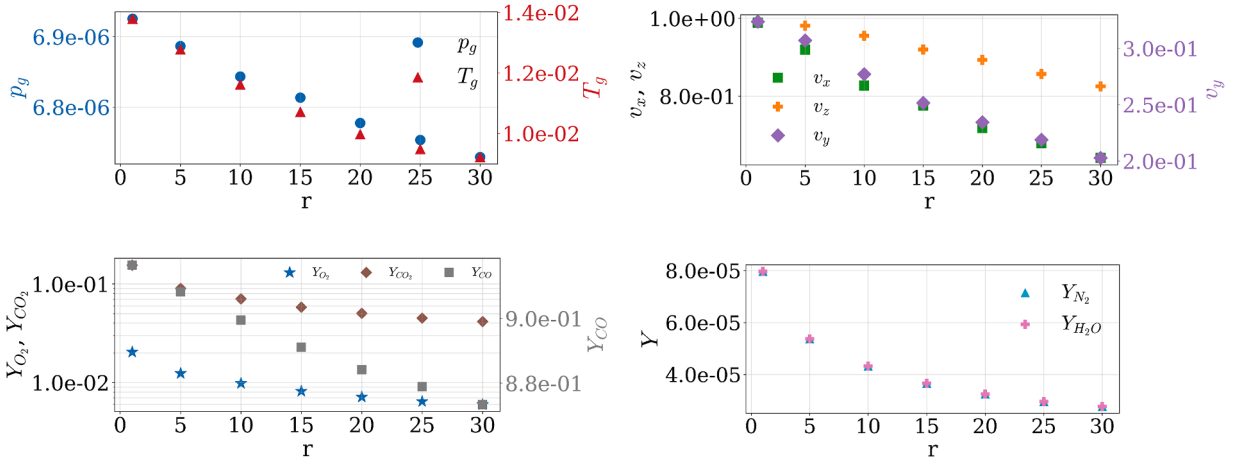


Fig. 6. Projection errors of each variable. The errors are computed with the variables in their original scales as $\|\mathbf{Q}^{(\ell)} - \tilde{\mathbf{Q}}^{(\ell)}\|_F^2 / \|\mathbf{Q}^{(\ell)}\|_F^2$.

4.3. Regularization hyperparameters selection based on key performance indicator

To select the hyperparameters λ_1 and λ_2 for regularizing the least-squares problem in Eq. (22), we create a logarithmically-spaced uniform parameter grid, where $(\lambda_1, \lambda_2) \in [10^{-1}, 10^6] \times [10^3, 10^{10}]$, with each dimension discretized into 10 values. Using each set of inferred reduced operators with each regularization on the (λ_1, λ_2) grid, the ROM propagates the initial reduced state $\hat{\mathbf{s}}_0$ within the training regime through numerical time integration of Eq. (21), obtaining the ROM solution in the reduced space $\hat{\mathbf{S}}^{\text{ROM}} = [\hat{\mathbf{s}}_0^{\text{ROM}}, \dots, \hat{\mathbf{s}}_{K-1}^{\text{ROM}}] \in \mathbb{R}^{r \times K}$. Conventionally, one selects the best λ_1^* and λ_2^* values that minimize the relative state error between the FOM reduced states Eq. (20) and the ROM solution,

$$e^{\text{ROM}} = \frac{\|\hat{\mathbf{S}}^{\text{FOM}} - \hat{\mathbf{S}}^{\text{ROM}}\|_F}{\|\hat{\mathbf{S}}^{\text{FOM}}\|_F} \times 100 (\%). \quad (31)$$

While widely used, this approach is inadequate in our case because it measures accuracy only in the *scaled reduced state space*, where contributions of different variables are scaled and mixed. As a result, a model may achieve a lower relative state error yet fail to capture essential nonlinear dynamics in the physical variables.

We propose to use the key performance indicator of char combustion as an alternative way of selecting hyperparameters for regularization. During each ROM prediction in the (λ_1, λ_2) grid search, we compute a crucial derived quantity of char combustion as the KPI: the thermal energy collected at the boiler outlet A_O (see Fig. 1). This quantity serves as a ROM selection metric throughout the grid search. The thermal energy $\xi(t)[\text{J}]$ is formulated as

$$\xi(t) = \int_{t_0}^t \underbrace{\left[\frac{1}{|A_O|} \int_{A_O} c_{p,g}(\mathbf{x}, \tau) \cdot \dot{m}(\mathbf{x}, \tau) \cdot \Delta T(\mathbf{x}, \tau) d\mathbf{x} \right]}_{\xi(\tau)} d\tau. \quad (32)$$

Here, the specific heat capacity of the gas mixture, $c_{p,g}$, is computed as $\sum_i c_{p,i} \cdot X_i / M_{\text{mix}}$ where $M_{\text{mix}} = \sum_i X_i \cdot M_i$ represents molar mass of the gas mixture. The molar fraction of each species is X_i , and its molar mass is M_i , i.e., $M_1 = 31.9988[\text{g/mol}]$ (O_2), $M_2 = 28.0134[\text{g/mol}]$ (N_2), $M_3 = 28.0104[\text{g/mol}]$ (CO), $M_4 = 44.0098[\text{g/mol}]$ (CO_2), and $M_5 = 18.0153[\text{g/mol}]$ (H_2O). The mass flow rate, \dot{m} [kg/s], through a surface is computed by summing the product of the gas phase volume fraction, density, velocity component normal to the boiler outlet surface A_O , and the area: $\int \varepsilon_g \cdot \rho_g \cdot v_y \cdot dA_O$, which in turn is computed in the discretized setting as $\sum_{ijk} \varepsilon_{g,ijk} \cdot \rho_{g,ijk} \cdot v_{y,ijk} \cdot |A_{O,ijk}|$, where ijk are the spatial indices of the discretization of A_O . Here, $\xi(\tau)$ represents the thermal energy rate [J/s], averaged across the boiler outlet A_O . To compute Eq. (32), all state variables except for v_x and v_z are required. However, the computation only needs the reconstruction of the high-dimensional variables in a specific region, A_O , making it computationally feasible.

We generate two ROMs based on different regularization methods: one is chosen via the KPI-based approach in Eq. (32), where thermal energy guides hyperparameter selection, and the ROM that is trained by minimizing the relative reduced-state error of Eq. (31). Fig. 7 compares temperature and N_2 mass fraction predictions for those two ROMs. It demonstrates that our proposed KPI-based approach provides a more physically accurate ROM for this problem, as it captures key nonlinear combustion dynamics in both temperature and N_2 mass fraction better than the other ROM, which is not expressive despite minimizing Eq. (31). In particular, the conventional relative state error metric of Eq. (31) assigns the KPI-based ROM a slightly lower score of $e^{\text{ROM}} = 6.64\%$ compared to $e^{\text{ROM}} = 6.35\%$ for the other that minimizes Eq. (31). This highlights that minimizing the reduced-state error does not ensure physical fidelity for our complex, multiscale char combustion case. In contrast, the proposed KPI-based approach aggregates contributions

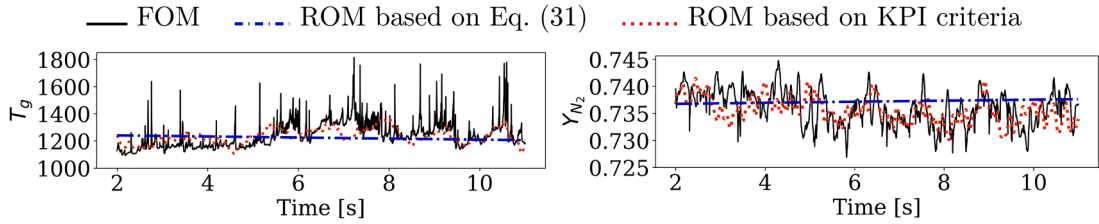


Fig. 7. Comparison of two ROM predictions of temperature and N_2 mass fraction over 9,000 snapshots spanning from 2 to 11s, averaged across the boiler outlet surface.

from multiple variables in the *full physical space*, directly reflecting the underlying combustion physics, and yields a ROM that better preserves critical combustion behavior.

For this dataset, the proposed approach is less expensive than the conventional method of selecting regularization hyperparameters based on the state error minimization metric. This is primarily because the norm computation for $\|\hat{S} - \hat{S}^{ROM}\|_F / \|\hat{S}\|_F$ in the conventional method is costly as the number of training snapshots K is high. When computing the error of the predicted KPI, we compare its full time series of the thermal energy $\xi(t)$, not the thermal energy accumulated at the final training time step, $\xi(t_{K-1})$. This avoids cases where the final energy would be similar, but the production curves thereof would have differed significantly. We also exclude regularizations that produce negative values of $\xi(t)$. For each combination of regularizations, the OpInf problem is solved, the resulting ROM is integrated in time, and $\xi(t)$ is computed from the reconstructed high-dimensional states over a partial domain—combustion outlet A_O —using the unconstrained ROM trajectory. Note that state constraints are not applied during this step, thereby avoiding the substantial offline cost of repeatedly enforcing them in the full high-dimensional space throughout the grid search. The hyperparameters, $(\lambda_1, \lambda_2) = (129.1550, 5994.8425)$, are selected via this KPI-based grid search. These hyperparameters minimize the error between $\xi^{FOM}(t)$ and $\xi^{ROM}(t)$, resulting in a relative error of 10.57%.

4.4. Comparison with other stability-enhancing operator inference reduced-order model approaches

Since the proposed OpInf with state constraints also enhances the stability of the ROM, we compare it with two other advanced stability-enhancing approaches of OpInf: eigenvalue reassignment and constrained optimization. Additional results for OpInf using log-transformed species mass fractions for positivity preservation are included in [Appendix C](#).

4.4.1. Eigenvalue reassignment

The OpInf framework finds the reduced operators by minimizing the cost function of [Eq. \(22\)](#). As a result, it does not guarantee that the inferred linear operator $\hat{A} \in \mathbb{R}^{r \times r}$ has only stable eigenvalues. Eigenvalue reassignment generates a stable \hat{A} by repositioning its unstable eigenvalues to be stable (see, e.g., [25, Sec. 3.4.1]). First, assuming \hat{A} is diagonalizable, we identify the eigenvalues of \hat{A} by computing its eigendecomposition as

$$\hat{A} = \Phi \Sigma_{\hat{A}} \Phi^{-1} \tag{33}$$

where the columns of Φ are eigenvectors of \hat{A} , and $\Sigma_{\hat{A}}$ is a diagonal matrix containing the eigenvalues of \hat{A} . Assume that the leading p eigenvalues of \hat{A} have non-negative real parts. Let these eigenvalues be $\sigma_1, \dots, \sigma_p$ and let the remaining eigenvalues be $\sigma_{p+1}, \dots, \sigma_r$. Denote $\Re(\sigma)$ and $\Im(\sigma)$ as real and imaginary parts of the eigenvalue σ . We replace $\Re(\sigma_1), \dots, \Re(\sigma_p)$ with a small tolerance $-\epsilon$, where $\epsilon > 0$. The linear reduced operator \hat{A} is then replaced by a new stable matrix $\Phi \tilde{\Sigma} \Phi^{-1}$, where $\tilde{\Sigma} = \text{diag}(-\epsilon + \Im(\sigma_1), \dots, -\epsilon + \Im(\sigma_p), \sigma_{p+1}, \dots, \sigma_r)$. As a result, all the eigenvalues in $\tilde{\Sigma}$ have strictly negative real parts.

4.4.2. Operator inference with constrained optimization

The authors in [25] considered constrained optimization within a structure-preserving OpInf framework to identify a symmetric negative definite linear matrix \hat{A} . They demonstrated the local stabilization of the OpInf ROM by constrained optimization on Burgers' equation and the reaction-diffusion equation. Yet, its performance in strongly nonlinear multiphysics problems, with conditions not close to the linear regime, is not certain. For the purpose of the comparison in this paper, we solve the constrained optimization for the OpInf with char combustion data. For numerical purposes, we implement this with a model constraint $\hat{A} + \epsilon \mathbf{I} \leq \mathbf{0}$ in [Eq. \(22\)](#), thereby finding a stable matrix \hat{A} . Here, $\mathbf{I} \in \mathbb{R}^{r \times r}$ is an identity matrix, $\hat{A} + \epsilon \mathbf{I} \leq \mathbf{0}$ means $\hat{A} + \epsilon \mathbf{I}$ is negative semidefinite, and the tolerance $\epsilon > 0$ ensures that \hat{A} is negative definite. We use the `cvxpy` 1.6 python package [52,53] for negative semidefinite programming, without a symmetry constraint and with a default tolerance of $\epsilon = 10^{-8}$.

4.4.3. Error metric

We calculate pointwise errors for all variables in their original scales as an error metric. Since pressure and temperature have large magnitudes, we use a standard relative error for these variables. The notation follows [Eq. \(14\)](#), except that the prediction is made for the $K_f = 38,000$ snapshots for each ℓ th state variable on the $n_x = 22,400$ cells, such that $\mathbf{Q}^{(\ell)} \in \mathbb{R}^{22,400 \times 38,000}$. The relative

pointwise error is computed as

$$\Upsilon_{j,k}^{\text{rel}} = \frac{|\mathbf{Q}_{j,k}^{(\ell),\text{FOM}} - \mathbf{Q}_{j,k}^{(\ell),\text{ROM}}|}{|\mathbf{Q}_{j,k}^{(\ell),\text{FOM}}|},$$

where j and k denote spatial and temporal indices. However, this error metric can often skew a small error if true values in the denominator are small. Thus, for other variables (velocity components and species mass fractions) that have relatively smaller scales and even include values close to zero, we compute a normalized absolute error

$$\Upsilon_{j,k}^{\text{nabs}} = \frac{|\mathbf{Q}_{j,k}^{(\ell),\text{FOM}} - \mathbf{Q}_{j,k}^{(\ell),\text{ROM}}|}{\max(|\mathbf{Q}^{(\ell),\text{FOM}}|)}.$$

Here, the denominator denotes the maximum of absolute values of $\mathbf{Q}^{(\ell),\text{FOM}}$ throughout the entire spatial and temporal domain. The resulting error matrix $\Upsilon \in \mathbb{R}^{n_x \times K_f}$ contains pointwise errors for all K_f snapshots throughout the training and testing regime, over the entire spatial domain. To analyze the temporal evolution of the error across the spatial domain, we calculate the spatially averaged pointwise error where each element $\bar{\Upsilon}_k = \frac{1}{n_x} \sum_{j=1}^{n_x} \Upsilon_{j,k}$ represents the spatially averaged pointwise error at the time step k .

4.4.4. Results

Stability and error comparison. Fig. 8 compares errors introduced in Section 4.4.3 for four models: (i) the proposed OpInf with species limiters, (ii) standard (unconstrained) OpInf, (iii) OpInf with eigenvalue reassignment, (iv) OpInf with constrained optimization. All ROMs propagate low-dimensional states using the RK45 time-integration scheme. For a fair comparison and to isolate the effect of each OpInf variant, we apply the same regularization hyperparameters, λ_1 and λ_2 , across all OpInf ROMs. These regularizers are selected based on the KPI criterion for the standard OpInf. This ensures that any observed performance differences arise solely from the formulation of each OpInf variant, rather than from differences in regularization. Except for the proposed OpInf with species limiters, the other ROMs implemented propagate states without state constraints, see also Fig. 3, where this is illustrated with dashed arrows.

First, the error evolution of all OpInf models is similar in the training regime, except for OpInf with eigenvalue reassignment. This reflects the contribution of appropriate regularization to accuracy, achieved through the KPI-based selection of hyperparameters during the model learning phase. As the predictions go beyond the training regime, we observe a divergence of errors for all variables with the standard OpInf. In contrast, the OpInf with state constraints on the species mass fractions maintains stable and consistently lower errors. This demonstrates that enforcing state constraints enhances accuracy by providing a stabilizing effect. The OpInf with eigenvalue reassignment produces larger errors than the standard OpInf within the training regime, eventually resulting in divergent errors in the testing regime.

This highlights that making a highly nonlinear system locally stable simply by reassigning unstable eigenvalues of its linear operator does not guarantee global ROM stability and accuracy. In addition, the divergent predictions arise because the quadratic term in the model Eq. (21) dominates the system dynamics in the testing regime. In contrast, the OpInf with constrained optimization learns the optimal stable $\hat{\mathbf{A}} < \mathbf{0}$ in the model learning process. Its error behavior in the training regime is similar to the OpInf with species limiters. In the testing regime, its errors become slightly larger than those of the OpInf with species limiters. Note that although evaluating errors after $t = 40s$ is not possible because we do not have FOM data, we can still check if the ROM predictions are stable and physically consistent. We found that predictions by OpInf with constrained optimization begin to diverge at $t = 50s$, whereas OpInf with species limiters is stable yet becomes less expressive. Similar to the OpInf with eigenvalue reassignment, the divergent ROM predictions by OpInf with constrained optimization in the time horizon far beyond the testing regime are due to the predominant influence of the quadratic term in the ROM. We also investigated cubic OpInf ROMs with KPI-based regularization. However, their offline training—especially the grid search for optimal regularization hyperparameters—was substantially more expensive than that of quadratic ROMs due to the increased number of operations (see Section 3.3). In addition, the cubic ROMs yielded less expressive solutions than the quadratic ROMs.

A critical observation is that the state constraints applied to the species mass fractions provide a global stabilizing effect across all variables, including pressure, temperature, and velocity components that are not subject to constraints. This added benefit is due to using a global POD basis. As depicted in Fig. 4 and detailed in Section 3.4.4, at each time step k , the entire reduced state vector $\hat{\mathbf{s}}_k$ is modified to $\hat{\mathbf{s}}_k^*$, even though the constraints are applied only to species mass fractions. The modified reduced state $\hat{\mathbf{s}}_k^*$ evolves to $\hat{\mathbf{s}}_{k+1}$. The $\hat{\mathbf{s}}_{k+1}$ is then lifted to the high-dimensional scaled state $\tilde{\mathbf{s}}_{k+1}$, which contains all variables. This process ensures that the ROM always propagates entirely modified reduced states, eventually influencing the trajectories of all variables.

Conservation of physical constraints. Next, we check if the OpInf ROMs make unphysical predictions. Table 3 presents the ratio of the unphysical species mass fractions predicted with each OpInf ROM over the entire spatial and temporal regime. Predictions on species mass fractions are considered unphysical if they violate the conditions specified in Eqs. (24)–(28). The standard OpInf, OpInf with eigenvalue reassignment, and OpInf with constrained optimization generate unphysical predictions for all species, whereas the OpInf with species limiters Eqs. (24)–(28) remains physically consistent for all species across the entire spatio-temporal domain. If we further relax the upper bounds in Eqs. (24), (25), and (28) to one (which would be known a priori without FOM or the boundary condition data of this specific model), the ROM produces unphysical results for O_2 , N_2 , and H_2O . This demonstrates that incorporating any available physical information into the design of state constraints improves the physical consistency of the ROM predictions. We note that, while the KPI strategy improves model accuracy during training, it alone does not ensure physically valid predictions: the

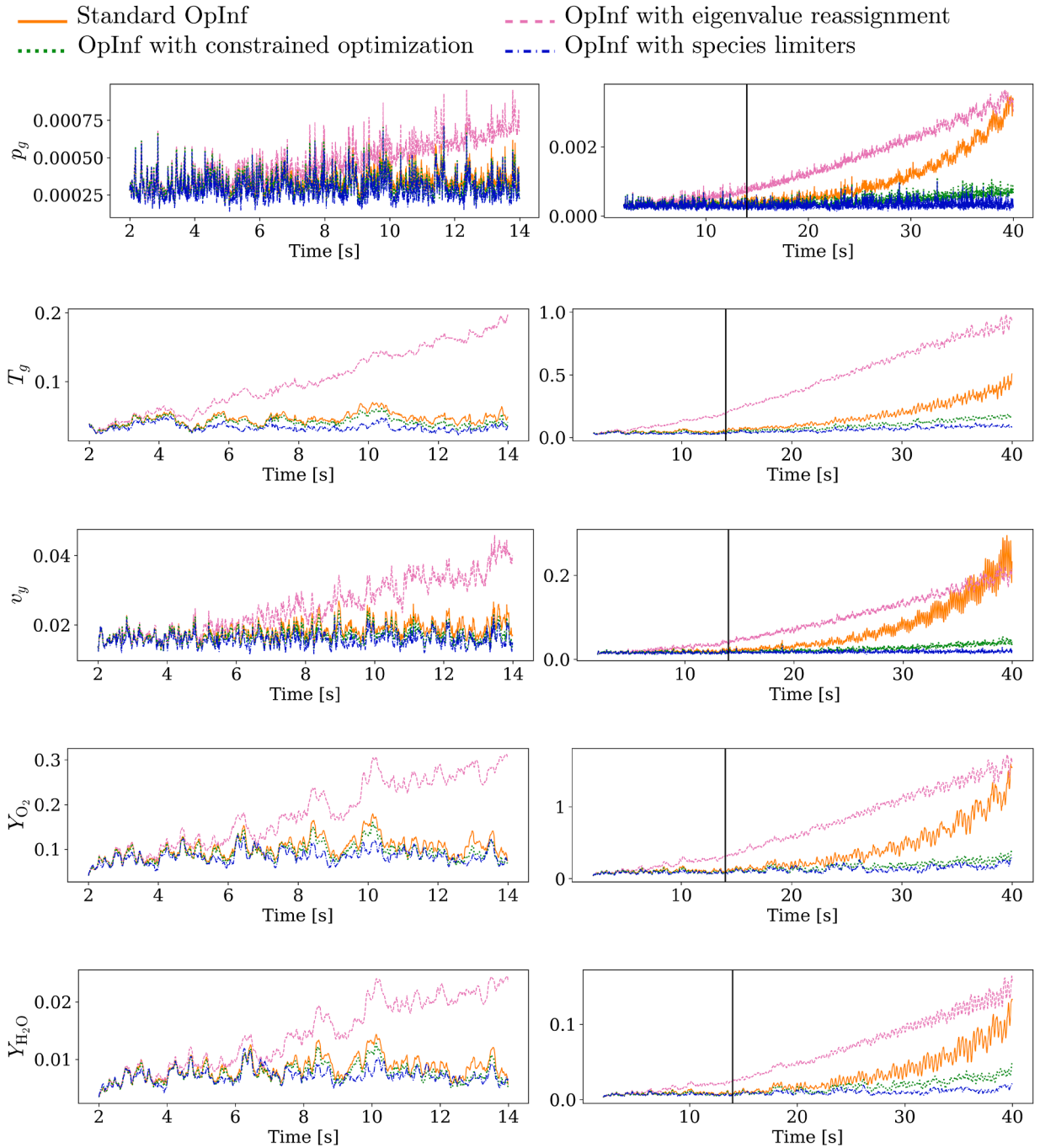


Fig. 8. Comparison of spatially averaged pointwise error \bar{Y}_k of all methodologies. Figures in the left column show error propagation over the training regime, while those in the right column show the full time horizon. The black vertical lines mark the end of the training regime, which comprises 12,000 snapshots in $t \in [2, 14]$ s, followed by the testing regime (26,000 snapshots in $t \in [14, 40]$ s). Variables not shown here are presented in [Appendix B](#).

physically consistent behavior of species-limited OpInf can be directly attributed to the enforcement of state constraints. We also observe unphysical predictions by OpInf with species limiters enforced every α th time step, see the last two rows of [Table 3](#). We provide a detailed discussion together with the computational time analysis presented later.

Additionally, we check whether each ROM conserves the sum of mass fractions, $Y_{O_2} + Y_{N_2} + Y_{CO} + Y_{CO_2} + Y_{H_2O}$, ensuring that it remains equal to 1. [Fig. 9](#) compares the spatially averaged sum of mass fractions over time for each ROM. Among the four methods, only OpInf with species limiters consistently maintains the sum of five mass fractions close to 1 across the entire spatio-temporal domain.

Table 3

The percentage (%) of unphysical species mass fractions in $n_x = 22,400$ spatial grid cells over $K_f = 38,000$ snapshots. The last three cases apply the species limiters Eqs. (24)–(28) every α th time step.

	O ₂	N ₂	CO	CO ₂	H ₂ O
Standard OpInf	26.74	26.57	30.79	18.95	26.57
OpInf with eigenvalue reassignment	45.28	40.20	46.78	25.64	40.20
OpInf with constrained optimization	8.30	18.76	18.66	6.98	18.76
OpInf with species limiters of a generic bound [0, 1]	2.64	20.06	0	0	20.06
OpInf with species limiters ($\alpha = 1$)	0	0	0	0	0
OpInf with species limiters ($\alpha = 10$)	0.64	0.15	6.35	0.86	0.15
OpInf with species limiters ($\alpha = 100$)	3.11	2.65	15.23	3.30	0.65

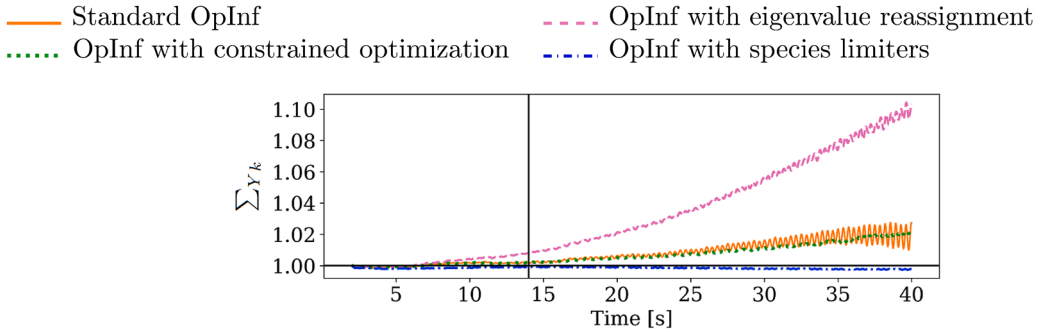


Fig. 9. The sum of mass fractions over time. The sum is computed by summing five mass fractions and then averaging over the spatial discretization dimension: $\sum Y_k = \frac{1}{n_x} \sum_{j=1}^{n_x} Y_{O_2,j,k} + Y_{N_2,j,k} + Y_{CO,j,k} + Y_{CO_2,j,k} + Y_{H_2O,j,k}$, where j and k represent spatial and temporal indices.

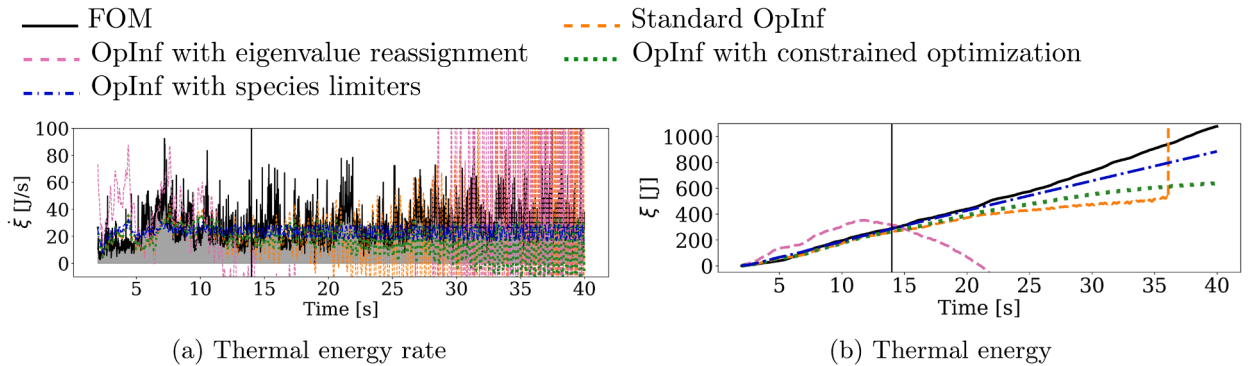


Fig. 10. Comparison of thermal energy rate and thermal energy. The shaded area under the reference ξ , representing its time-integrated quantity, corresponds to the thermal energy ξ .

Note that the proposed method does not explicitly enforce this constraint; however, the species limiters defined in Eqs. (24)–(28) help preserve this property.

Thermal energy prediction. Fig. 10 presents another comparison of the four methods regarding the prediction capabilities for the thermal energy, our KPI. As shown in Fig. 10(a), the standard OpInf, OpInf with eigenvalue reassignment, and OpInf with constrained optimization yield unstable predictions for the thermal energy rate, $\dot{\xi}$, computed using Eq. (32). Notably, these methods even predict negative values for $\dot{\xi}$ in the testing regime. This is unphysical because $\dot{\xi}$ must inherently remain strictly positive, given that the specific heat capacity $c_{p,g}$, mass flow rate \dot{m} , and the temperature gradient ΔT are all positive over the boiler outlet A_O). In particular, the OpInf with eigenvalue-reassignment predicts negative $\dot{\xi}$ even in the training regime, despite using the same regularizers as the other methods. This stems from the fact that the unstable eigenvalues of \hat{A} are reassigned after the model is trained. The instability and inaccuracy $\dot{\xi}(t)$ in the three methods lead to correspondingly unstable and inaccurate thermal energy predictions in Fig. 10(b). On the other hand, OpInf with species limiters makes stable and accurate predictions throughout training and testing, resulting in the closest predictions to the time series of the FOM thermal energy.

Comparison of computational time. Table 4 compares the four methods' computational cost. We run all four ROMs using Python 3.12.2 on an MPI-based cluster equipped with 56 Intel Xeon W-3175X CPUs @ 3.10 GHz. The offline and online computational costs are measured in wall clock time. The online speedup factor is computed by dividing the FOM simulation cost (60,480 CPU hours)

Table 4

Comparison of computational wall time for the offline stage (model learning in the training regime) and the online stage (time integration in the entire time regime) across all methods. The last three cases apply the species limiters every α th time step to investigate possible avenues for online speedup. To account for variations in ROM simulation time, we average the costs over 10 runs for each method.

	Offline [mins]	Online [mins]	Online speedup factor
Standard OpInf	2.68	0.01	4.41×10^9
OpInf with eigenvalue reassignment	2.69	0.01	4.41×10^9
OpInf with constrained optimization	64.36	0.01	4.41×10^9
OpInf with species limiters ($\alpha = 1$)	2.68	20.83	3.17×10^3
OpInf with species limiters ($\alpha = 10$)	2.68	5.14	1.26×10^4
OpInf with species limiters ($\alpha = 100$)	2.68	1.64	3.95×10^4

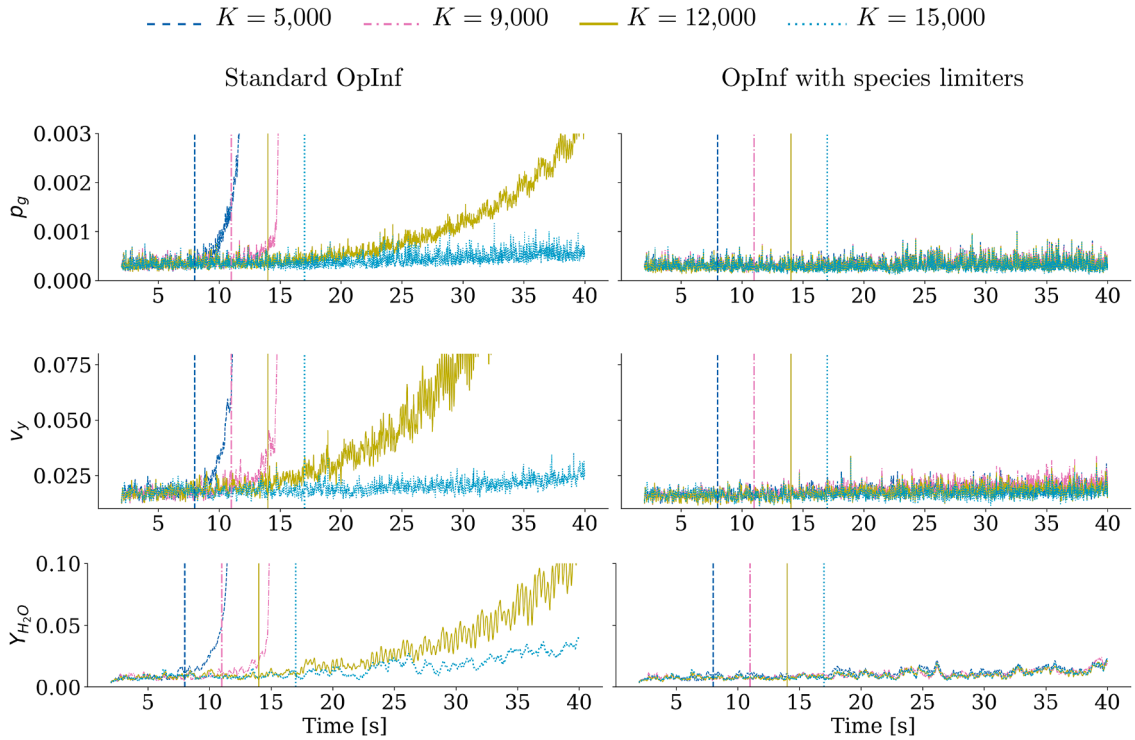


Fig. 11. Comparison across four training horizons: 2–7s ($K = 5,000$ snapshots), 2–11s ($K = 9,000$), 2–14s ($K = 12,000$), and 2–17s ($K = 15,000$). The vertical lines indicate the end of each training regime, with the line color and style matching the corresponding error curve.

by the ROM online cost, with all ROM costs converted to CPU hours. Specifically, we measure the ROM online CPU time using the `time.process_time()` function in `time` module of Python. The online phase for all methods includes the temporal evolution of the reduced states by the ROMs in the entire time domain, followed by the projection of the results into the high-dimensional space. The offline phase for all methods comprises data pre-processing, randomized SVD, and solving the optimization problem with regularization to learn the ROMs. The randomized SVD, which takes approximately 2.42 mins in wall time, is the most expensive step in the offline phase. For the OpInf with eigenvalue reassignment, the offline phase also includes reallocating unstable eigenvalues of the learned linear operator $\hat{\mathbf{A}} \in \mathbb{R}^{r \times r}$. This process is executed instantaneously, as the dimension of $\hat{\mathbf{A}}$ is only $r \ll N$.

The offline stage of the OpInf with constrained optimization includes solving a regularized optimization problem with a model constraint $\hat{\mathbf{A}} + \epsilon \mathbf{I} \leq \mathbf{0}$, which can be expensive depending on the amount of training data and the ROM dimension. In our case, this process takes more than 60 mins on average over 10 runs in wall time. Lastly, one should note that all four models are trained on the same amount of high-fidelity data with the same projection matrix. If not otherwise available, simulating that data may also factor into the offline cost.

Sensitivity of OpInf with species limiters to the amount of training data. We investigate the influence of training data length on the performance of the OpInf ROM. Fig. 11 shows the comparison of error propagation for standard OpInf ROMs and species-limited OpInf ROMs trained with varying numbers of snapshots, with all models selected based on the KPI-based regularization with parameters given in Section 4.3. The standard OpInf shows unstable error growth when trained with $K = 5,000, 9,000$, and $12,000$ snapshots,

whereas extending the training set to 15,000 snapshots improves stability and extends accuracy further into the testing horizon. In contrast, the species-limited OpInf ROMs yield consistently stable error behavior across all training lengths. The error curves for these cases nearly overlap, indicating that the performance of the species-limited OpInf ROMs is largely insensitive to the amount of training data.

Summary of ROM performance. The performance of the four OpInf ROMs can be summarized as follows. First, standard OpInf, OpInf with eigenvalue reassignment, and OpInf with constrained optimization are the fastest methods online. However, they encounter stability and accuracy issues when making long-term predictions, often resulting in physically inconsistent results and inaccurate predictions on our KPI. OpInf with constrained optimization involves considerable offline computational costs, making it more expensive to train compared to other approaches. Additionally, this method does not always produce expressive ROMs when evaluated with training data across different temporal regimes. For example, using a larger amount of training data (e.g., $K = 15,000$ snapshots) results in less expressive predictions with fewer oscillations compared to the FOM data. Conversely, unlike the species-limited ROM cases shown in Fig. 11, training with fewer snapshots (e.g., $K = 5,000$ and $K = 9,000$ snapshots) leads to instability, even with stabilization efforts. Lastly, OpInf that enforces species limiters every time step incurs the highest online cost, leading to the smallest online speedup among all methods. This is because it requires reconstructing high-dimensional states at each time step, enforcing species mass fraction constraints (as outlined in Eqs. (24)–(28)), and projecting the modified states back to the low-dimensional space, see Fig. 3. Nevertheless, it achieves a significant speedup of 3,170 times compared to the FOM computation. While enforcing species limiters every α th time step (here, $\alpha = 10$ and 100) improves online speedup, as shown in Table 3, this strategy does not ensure physically valid mass fraction predictions at intermediate time steps when the constraints are not applied. As demonstrated in the numerics above, only OpInf with species limiters at every time step ensures physically consistent predictions for species mass fractions in this application. It also provides the best accuracy and stability among all methods tested.

5. Conclusions

We proposed a new approach to enhance accuracy and stability of nonintrusive OpInf ROMs, while ensuring physical consistency, as demonstrated in a char combustion application. We incorporated state constraints into the prediction step of the OpInf ROMs, specifically for the species mass fraction variables, enabling physically consistent predictions across the entire spatio-temporal domain. Our approach combines efficient ROM state evolution with an online correction that partially reintroduces FOM information, forming a hybrid offline-online framework. We further introduced a strategy for selecting regularization hyperparameters in OpInf ROM training, guided by key performance indicators (KPIs) that can be defined for a wide range of physical and engineering applications. The proposed OpInf ROM with state constraints demonstrated superior performance in terms of stability, accuracy, and physical consistency, compared to the standard OpInf and other stability-enhancing approaches, which failed to fully preserve species mass fraction bounds and showed error growth, albeit at different rates. We further showed that combining a global POD basis with state constraints provides the strongest stabilizing effect and the highest prediction accuracy, outperforming the block-POD basis. This approach also stabilizes other variables, such as pressure, temperature, and velocity components, which are not directly constrained. Although the state-constrained OpInf ROM incurs additional online computational cost, it still achieved a significant speedup of approximately 3170 times relative to the FOM simulation.

The proposed approach offers a promising avenue for developing fast, stable, and physically consistent data-driven ROMs for large-scale nonlinear systems. Future work could explore formulating the OpInf framework in an integral or a discrete-time form to address challenges in estimating time derivatives of POD coefficients for highly noisy, oscillatory, and coarsely sampled data. Also, investigating ways to reduce the online computational cost of the current expensive state constraints step, which relies on FOM information in the high-dimensional space, would be very valuable. We also attempted to impose limiting conditions directly on the reduced states after projecting the species mass fraction variable onto its own latent space via a block-structured basis. However, the physical bounds are encoded through coupled interactions between POD modes and coefficients, making it difficult to enforce such constraints in the low-dimensional space. Finally, the very recent work [54] proposes a method to ensure the sum of mass fractions is equal to 1 in high-fidelity simulations of reacting flows. It would be interesting to impose such a constraint with our proposed state constraints in the ROM.

CRedit authorship contribution statement

Hyeonhun Kim: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Formal analysis, Data curation, Conceptualization; **Boris Kramer:** Writing – review & editing, Supervision, Project administration, Methodology, Funding acquisition, Formal analysis, Conceptualization.

Data availability

Data will be made available on request.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This research was in part financially supported by the [Korea Institute for Advancement of Technology \(KIAT\)](#) through the International Cooperative R&D program (No. [P0019804](#), Digital twin-based intelligent unmanned facility inspection solutions).

Appendix A. Nomenclature

ρ_g	gas density [kg/m ³]
\mathbf{v}_g	gas velocity vector in 3D space [m/s]
\mathbf{v}_p	solid particle velocity vector in 3D space [m/s]
N_g	number of chemical species in gas mixture
$R_{g,i}$	rate of formation per unit volume of i th gas species
$R_{p,i}$	rate of formation for the p th MFiX-PIC parcel of i th chemical species
S_g	gas source term [kg/(m ³ · s)]
S_p	solid source term [kg/(m ³ · s)]
Y_i	mass fraction of i th gas species
$Y_{p,i}$	mass fraction of i th gas species for the p th parcel
X_i	molar fraction of i th gas species
\mathbf{j}_g	diffusive flux vector
p_g	gas pressure [Pa]
\mathbf{g}	gravitational acceleration vector [m/s ²]
μ_g	dynamic viscosity [kg/(m · s)]
\mathbf{S}_g	general gas phase momentum source term vector [kg/(m ² · s ²)]
$\boldsymbol{\tau}_g$	gas stress tensor [kg/(m · s ²)]
$\boldsymbol{\tau}_p$	interparticle stress tensor [kg/(m · s ²)]
$c_{p,g}$	gas phase mixture specific heat capacity at constant pressure [J/(kg · K)]
$c_{p,p}$	solid parcel specific heat capacity at constant pressure [J/(kg · K)]
$c_{p,i}$	i th gas specific heat capacity [J/(kg · K)]
T_g	gas phase temperature [K]
T_p	solid phase temperature [K]
κ	thermal conductivity [W/(m · K)]
$h_{p,i}$	i th species specific enthalpy [J/kg]
m_{ci}	mass of unreacted char [kg]
$d_{p,i}$	diameter of a char particle [m]
m_p	particle mass [kg]
p_{oxy}	partial pressure of oxygen in gas mixture [Pa]
M	molecular weight [g/mol]
R_{diff}	diffusion rate
R_{chem}	kinetic rate
Sh	Sherwood number
D_o	binary diffusion coefficient of oxygen-nitrogen mixture [m ² /s]
R_u	universal gas constant [J/(kmol · K)]
T_m	mean temperature between gas and solid [K]
A_{pre}	pre-exponential factor
E_0	activation energy
Re_g	Reynolds number
\overline{U}	slip velocity between fluid and particles
Sc	Schmidt number
ϵ_g	gas volume fraction
W_p	statistical weight of particles

Appendix B. Error propagation of all variables

To complement the results presented in [Fig. 8](#), we include here the error plots of the remaining five variables, see [Fig. B.1](#). These results verify the same overall trends of error evolution discussed in [Section 4.4.4](#).

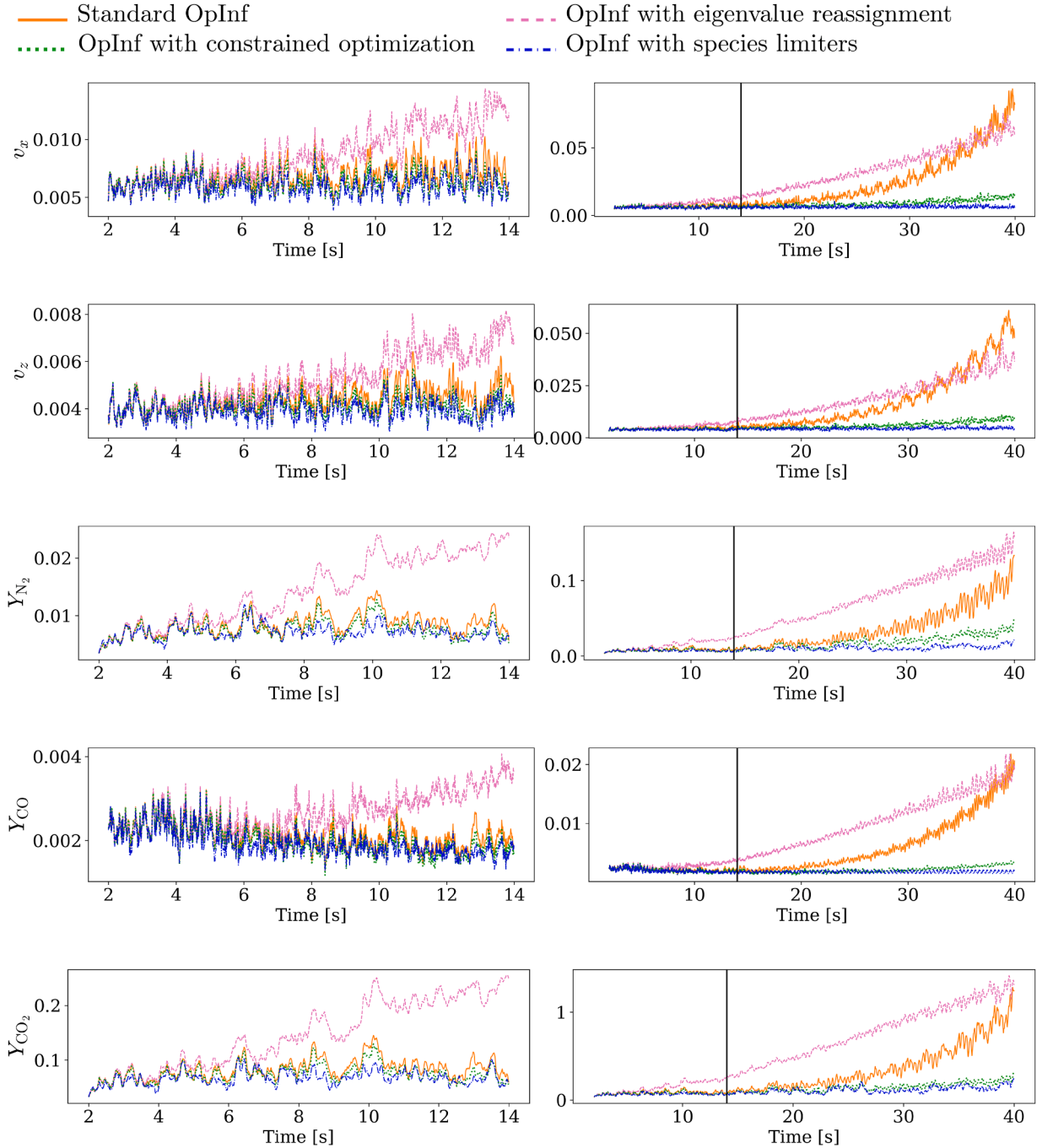


Fig. B.1. Figures in the left column show error propagation over the training regime, while those in the right column show the error propagation over the full time horizon.

Appendix C. Operator Inference using log-transformed species mass fractions for positivity preservation

In this appendix, we discuss a strategy for learning an OpInf ROM using log-transformed species mass fraction data. As noted in Section 3.1, the species mass fractions are not preprocessed in this work, since their values already fall within the target range used in data preprocessing, $[-1, 1]$. Nevertheless, a log-transformation can be applied prior to training to ensure positivity of the predicted mass fractions. Since the minimum mass fraction in the training data is zero (see Table 2), we introduce a small positive tolerance,

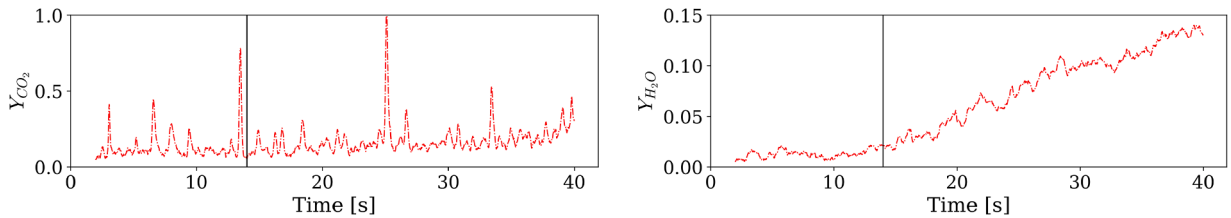


Fig. C.2. Time evolution of spatially averaged error \bar{Y}_k (see Section 4.4.3) for CO_2 and H_2O of OpInf with log-transformed mass fractions.

$0 < \epsilon \ll 1$, to avoid taking the logarithm of zero. The log-transformed mass fractions are then defined as

$$Y_{\ln} = \ln(Y + \epsilon),$$

which satisfies $Y_{\ln} < 0$ for all training mass fractions in the range $0 \leq Y < 1 - \epsilon$. We then scale the log-transformed mass fractions using the same way as in Eq. (18). After training a polynomial OpInf ROM, we propagate the low-dimensional states via the standard OpInf ROM state evolution illustrated in Fig. 3. Finally, the high-dimensional states are reconstructed from the predicted low-dimensional states, followed by unscaling and inversion of the logarithm to recover positive mass fractions. The procedure can be summarized as follows:

1. Apply the logarithm to species mass fractions: $Y_{\ln} = \ln(Y + \epsilon) < 0$.
2. Obtain the scaled log-transformed states $Y_{\ln}^{\text{scaled}} = \frac{Y_{\ln}}{\beta}$, where $\beta = \min(Y_{\ln}) < 0$. This ensures $0 \leq Y_{\ln}^{\text{scaled}} \leq 1$.
3. Train the OpInf ROM.
4. Propagate the low-dimensional states.
5. Reconstruct high-dimensional log-transformed mass fractions \tilde{Y}_{\ln} .
6. Obtain the unscaled log-transformed prediction $\tilde{Y}'_{\ln} = \beta \cdot \tilde{Y}_{\ln}$.
7. Recover the mass fraction predictions in the original domain: $Y_{\text{pred}} = e^{\tilde{Y}'_{\ln}} - \epsilon$.

Due to the inclusion of ϵ , the recovered mass fraction in step 7 satisfies $Y_{\text{pred}} \geq -\epsilon$. In other words, the predicted mass fractions are bounded below by a negative value infinitesimally close to zero. In our test, we set $\epsilon = 10^{-16}$. However, this approach still fails to achieve our goal of physical consistency of mass fractions and stability. While the preprocessed high-dimensional log-transformed mass fractions are strictly bounded ($0 \leq Y_{\ln}^{\text{scaled}} \leq 1$ at step 2), the reconstructed log-transformed states \tilde{Y}_{\ln} at step 5 are obtained through the coupled ROM dynamics and are not guaranteed to remain within the same bounds. In particular, if $\tilde{Y}_{\ln} < 0$, then the resultant \tilde{Y}'_{\ln} (after unscaling) may become positive. This, in turn, leads to $Y_{\text{pred}} > 1$, which violates the physical consistency of mass fractions and may cause instability in long-time ROM predictions. Fig. C.2 shows unstable time-evolution of spatially averaged error \bar{Y}_k of two variables, Y_{CO_2} and $Y_{\text{H}_2\text{O}}$.

References

- [1] X. Ku, T. Li, T. Lövås, CFD-DEM simulation of biomass gasification with steam in a fluidized bed reactor, *Chem. Eng. Sci.* 122 (2015) 270–283. <https://doi.org/10.1016/j.ces.2014.08.045>
- [2] A. Quarteroni, G. Rozza, et al., *Reduced Order Methods for Modeling and Computational Reduction*, 9, Springer, 2014.
- [3] P. Benner, S. Gugercin, K. Willcox, A survey of projection-based model reduction methods for parametric dynamical systems, *SIAM Rev.* 57 (4) (2015) 483–531.
- [4] J.S. Hesthaven, G. Rozza, B. Stamm, et al., *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*, 590, Springer, 2016.
- [5] O. Ghattas, K. Willcox, Learning physics-based models from data: perspectives from inverse problems and model reduction, *Acta Numer.* 30 (2021) 445–554.
- [6] B. Peherstorfer, K. Willcox, Data-driven operator inference for nonintrusive projection-based model reduction, *Comput. Methods Appl. Mech. Eng.* 306 (2016) 196–215. <https://doi.org/10.1016/j.cma.2016.03.025>
- [7] B. Kramer, B. Peherstorfer, K.E. Willcox, Learning nonlinear reduced models from data with operator inference, *Annu. Rev. Fluid Mech.* 56 (1) (2024) 521–548.
- [8] O. Issan, B. Kramer, Predicting solar wind streams from the inner-heliosphere to earth via shifted operator inference, *J. Comput. Phys.* 473 (2023) 111689.
- [9] B.G. Zastrow, A. Chaudhuri, K. Willcox, A.S. Ashley, M.C. Henson, Data-driven model reduction via operator inference for coupled aeroelastic flutter, in: *AIAA SciTech 2023 Forum*, 2023, p. 0330.
- [10] P. Benner, P. Goyal, J. Heiland, I.P. Duff, Operator inference and physics-informed learning of low-dimensional models for incompressible flows, *Electron. Trans. Numer. Anal.* 56 (2022) 28–51.
- [11] P.R.B. Rocha, J.L. de Sousa Almeida, M.S. de Paula Gomes, A.C.N. Junior, Reduced-order modeling of the two-dimensional Rayleigh-Bénard convection flow through a non-intrusive operator inference, *Eng. Appl. Artif. Intell.* 126 (2023) 106923.
- [12] I. Adibnazari, H. Sharma, J.C. Torralba, B. Kramer, M.T. Tolley, Full-body optimal control of a swimming soft robot enabled by data-driven model reduction, in: *2023 Southern California Robotics (SCR) Symposium* (2023).
- [13] S. Yildiz, P. Goyal, P. Benner, B. Karasözen, Learning reduced-order dynamics for parametrized shallow water equations from data, *Int. J. Numer. Methods Fluids* 93 (8) (2021) 2803–2821.
- [14] S.A. McQuarrie, P. Khodabakhshi, K.E. Willcox, Nonintrusive reduced-order models for parametric partial differential equations via data-driven operator inference, *SIAM J. Sci. Comput.* 45 (4) (2023) A1917–A1946.
- [15] I.-G. Farcas, R. Gundevia, R. Muniipalli, K.E. Willcox, Parametric non-intrusive reduced-order models via operator inference for large-scale rotating detonation engine simulations, in: *AIAA SciTech 2023 Forum*, 2023, p. 0172.
- [16] 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.
- [17] A.A. Kaptanoglu, J.L. Callahan, A. Aravkin, C.J. Hansen, S.L. Brunton, Promoting global stability in data-driven models of quadratic nonlinear dynamics, *Phys. Rev. Fluids* 6 (9) (2021) 094401.
- [18] M. Schlegel, B.R. Noack, On long-term boundedness of Galerkin models, *J. Fluid Mech.* 765 (2015) 325–352.

- [19] S.E. Ahmed, S. Pawar, O. San, A. Rasheed, T. Iliescu, B.R. Noack, On closures for reduced order models—A spectrum of first-principle to machine-learned avenues, *Phys. Fluids* 33 (9) (2021).
- [20] 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 (2012) 10–26.
- [21] M. Benosman, J. Borggaard, O. San, B. Kramer, Learning-based robust stabilization for reduced-order models of 2D and 3D boussinesq equations, *Appl. Math. Model.* 49 (2017a) 162–181. <https://doi.org/10.1016/j.apm.2017.04.032>
- [22] M. Benosman, J. Borggaard, B. Kramer, Robust POD model stabilization for the 3D Boussinesq equations based on Lyapunov theory and extremum seeking, in: *American Control Conference (ACC)*, IEEE, 2017b, pp. 1827–1832.
- [23] V.L. Kalb, A.E. Deane, An intrinsic stabilization scheme for proper orthogonal decomposition based low-dimensional models, *Phys. Fluids* 19 (5) (2007).
- [24] B. Kramer, Stability domains for quadratic-bilinear reduced-order models, *SIAM J. Appl. Dyn. Syst.* 20 (2) (2021) 981–996.
- [25] N. Sawant, B. Kramer, B. Peherstorfer, Physics-informed regularization and structure preservation for learning stable reduced models from data with operator inference, *Comput. Methods Appl. Mech. Eng.* 404 (2023) 115836. <https://doi.org/10.1016/j.cma.2022.115836>
- [26] R. Swischuk, B. Kramer, C. Huang, K. Willcox, Learning physics-based reduced-order models for a single-injector combustion process, *AIAA J.* 58 (6) (2020) 2658–2672. <https://doi.org/10.2514/1.j058943>
- [27] S.A. McQuarrie, C. Huang, K.E. Willcox, Data-driven reduced-order models via regularised operator inference for a single-injector combustion process, *J. R. Soc. N. Z.* 51 (2) (2021) 194–211. <https://doi.org/10.1080/03036758.2020.1863237>
- [28] H. Sharma, Z. Wang, B. Kramer, Hamiltonian operator inference: physics-preserving learning of reduced-order models for canonical Hamiltonian systems, *Physica D* 431 (2022) 133122. <https://www.sciencedirect.com/science/article/pii/S0167278921002682>. <https://doi.org/10.1016/j.physd.2021.133122>
- [29] A. Gruber, I. Tezaur, Canonical and noncanonical hamiltonian operator inference, *Comput. Methods Appl. Mech. Eng.* 416 (2023) 116334.
- [30] H. Sharma, B. Kramer, Preserving Lagrangian structure in data-driven reduced-order modeling of large-scale dynamical systems, *Physica D* 462 (2024) 134128. <https://www.sciencedirect.com/science/article/pii/S0167278924000794>. <https://doi.org/10.1016/j.physd.2024.134128>
- [31] H. Sharma, D.A. Najera-Flores, M.D. Todd, B. Kramer, Lagrangian operator inference enhanced with structure-preserving machine learning for noninvasive model reduction of mechanical systems, *Comput. Methods Appl. Mech. Eng.* 423 (2024) 116865. <https://www.sciencedirect.com/science/article/pii/S004578252400121X>. <https://doi.org/10.1016/j.cma.2024.116865>
- [32] Y. Geng, J. Singh, L. Ju, B. Kramer, Z. Wang, Gradient preserving operator inference: data-driven reduced-order models for equations with gradient structure, *Comput. Methods Appl. Mech. Eng.* 427 (2024) 117033.
- [33] Y. Filanova, I.P. Duff, P. Goyal, P. Benner, An operator inference oriented approach for linear mechanical systems, *Mech. Syst. Signal Proces* 200 (2023) 110620.
- [34] T. Koike, E. Qian, Energy-preserving reduced operator inference for efficient design and control, in: *AIAA SciTech 2024 Forum*, 2024, p. 1012.
- [35] C. Huang, K. Duraisamy, C. Merkle, Data-informed species limiters for local robustness control of reduced-order models of reacting flow, in: *AIAA SciTech 2020 Forum*, 2020. <https://doi.org/10.2514/6.2020-2141>
- [36] E. Qian, L-G. Farcas, K. Willcox, Reduced operator inference for nonlinear partial differential equations, *SIAM J. Sci. Comput.* 44 (4) (2022) A1934–A1959.
- [37] P. Jain, S. McQuarrie, B. Kramer, Performance comparison of data-driven reduced models for a single-injector combustion process, in: *AIAA Propulsion and Energy 2021 Forum*, 2021, p. 3633.
- [38] Y. Geng, D. Che, An extended DEM–CFD model for char combustion in a bubbling fluidized bed combustor of inert sand, *Chem. Eng. Sci.* 66 (2) (2011) 207–219.
- [39] S. Yang, H. Wang, Y. Wei, J. Hu, J.W. Chew, Particle-scale modeling of biomass gasification in the three-dimensional bubbling fluidized bed, *Energy Convers. Manage.* 196 (2019) 1–17.
- [40] M.A. Clarke, J.M. Musser, *The MFIX Particle-in-Cell Method (MFIX-PIC) Theory Guide*, Technical Report, National Energy Technology Laboratory (NETL), 2020.
- [41] J.M. Musser, J.E. Carney, *Theoretical Review of the MFIX Fluid and Two-Fluid Models*, Technical Report, National Energy Technology Laboratory (NETL), 2020.
- [42] F. Winter, M.E. Prah, H. Hofbauer, Temperatures in a fuel particle burning in a fluidized bed: the effect of drying, devolatilization, and char combustion, *Combust. Flame* 108 (3) (1997) 302–314.
- [43] P. Basu, *Combustion and Gasification in Fluidized Beds*, CRC press, 2006.
- [44] J. Xie, W. Zhong, Y. Shao, Study on the char combustion in a fluidized bed by CFD-DEM simulations: influences of fuel properties, *Powder Technol.* 394 (2021) 20–34. <https://doi.org/10.1016/j.powtec.2021.08.018>
- [45] R.D. La Nauze, K. Jung, J. Kastl, Mass transfer to large particles in fluidised beds of smaller particles, *Chem. Eng. Sci.* 39 (11) (1984) 1623–1633. [https://doi.org/10.1016/0009-2509\(84\)80089-5](https://doi.org/10.1016/0009-2509(84)80089-5)
- [46] F. Scala, Mass transfer around freely moving active particles in the dense phase of a gas fluidized bed of inert particles, *Chem. Eng. Sci.* 62 (16) (2007) 4159–4176. <https://doi.org/10.1016/j.ces.2007.04.040>
- [47] F.L. Dryer, I. Glassman, High-temperature oxidation of CO and CH₄, *Symp. (Int.) Combust.* 14 (1) (1973) 987–1003. [https://doi.org/10.1016/S0082-0784\(73\)80090-6](https://doi.org/10.1016/S0082-0784(73)80090-6)
- [48] D. Lee, E. Lavichant, B. Kramer, Global sensitivity analysis with limited data via sparsity-promoting D-MORPH regression: application to char combustion, *J. Comput. Phys.* 511 (2024) 113116.
- [49] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, A. Müller, J. Nothman, G. Louppe, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, É. Duchesnay, Scikit-learn: Machine Learning in Python, 2018. [arXiv:1201.0490](https://arxiv.org/abs/1201.0490)
- [50] N. Halko, P.-G. Martinsson, J.A. Tropp, Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions, *SIAM Rev.* 53 (2) (2011) 217–288.
- [51] P. Benner, P. Goyal, B. Kramer, B. Peherstorfer, K. Willcox, Operator inference for non-intrusive model reduction of systems with non-polynomial nonlinear terms, *Comput. Methods Appl. Mech. Eng.* 372 (2020) 113433.
- [52] S. Diamond, S. Boyd, CVXPY: a python-embedded modeling language for convex optimization, *The J. Mach. Learn. Res.* 17 (1) (2016) 2909–2913.
- [53] A. Agrawal, R. Verschuere, S. Diamond, S. Boyd, A rewriting system for convex optimization problems, *J. Control and Decision* 5 (1) (2018) 42–60.
- [54] A. Baumgart, G. Blanquart, Ensuring $\sum_i Y_i = 1$ in transport of species mass fractions, *J. Comput. Phys.* 513 (2024) 113199.