Nonlinear Dyn
https://doi.org/10.1007/s11071-021-07094-8

ORIGINAL PAPER

A data-based stability-preserving model order reduction method for hyperbolic partial differential equations

Mohammad Hossein Abbasi · **Laura Iapichino** · **Wil Schilders** · **Nathan van de Wouw**

Received: 2 May 2021 / Accepted: 22 November 2021

Abstract This paper proposes a data-based approach for model order reduction that preserves incremental stability properties. Existing data-based approaches do typically not preserve such incremental system properties, especially for nonlinear systems. As a result, instability of the constructed model commonly occurs for inputs outside the training set, which seriously limits the usefulness of such models. Therefore, we propose to construct incrementally stable or incrementally ℓ_2-gain stable reduced-order nonlinear models to ensure robustness for a broad class of (bounded) input signals. Hereto, nonlinear discrete-time state-space equations are fitted to input-state-output data, obtained by simulations with the original model. We conjecture that certain classes of hyperbolic partial differential equations enjoy such incremental stability properties. Given the fact that complexity reduction in such PDE models is desirable, we employ the developed data-based reduction method to the discretized version of the hyperbolic equations thereby preserving the incremental stability features of the original system. In particular, this method is applied to a linear advection equation, for which stability properties are proved analytically. Finally, simulation results show the successful application of the method to the nonlinear Burgers' equation.

Keywords Model order reduction · Hyperbolic partial differential equation · Data-based reduction · Stability preservation · Non-intrusive model order reduction

Mathematics Subject Classification 35L65 · 37M05 · 93A15 · 93B99 · 93C99 · 93D05

1 Introduction

Data-based approximation of nonlinear dynamical systems by mathematical models is a challenging task in science and engineering. Data-based model order reduction (MOR) and system identification are two well-known branches of science and engineering that tackle approximation of dynamical systems. These two branches differ mainly in the source of the provided data. In MOR, data come from solutions of a complex model (called snapshots), which is usually based on the physical principles [6,16]. The critical issue in MOR is to find a model of lower order and complexity that imitates accurately the high-order dynamical

M. H. Abbasi (✉) · N. van de Wouw
Department of Mechanical Engineering, Eindhoven
University of Technology, Eindhoven, The Netherlands
e-mail: m.h.abbasi@tue.nl

L. Iapichino · W. Schilders
Department of Mathematics and Computer Science,
Eindhoven University of Technology, Eindhoven,
The Netherlands
e-mail: l.iapichino@tue.nl

W. Schilders
e-mail: w.h.a.schilders@tue.nl

N. van de Wouw
Department of Civil, Environmental and Geo-Engineering,
University of Minnesota, Minneapolis, USA
e-mail: n.v.d.wouw@tue.nl

Published online: 10 January 2022

model. In system identification, however, data come from experiments [24], which bring additional problems such as measurement noise and exogenous disturbances. Therefore, the techniques that have been developed in the system identification community can potentially be exploited and tailored for MOR by neglecting the effect of noise and disturbances. In this paper, we build upon a system identification technique for nonlinear systems to develop a novel data-based MOR of nonlinear systems, thereby combining the best of both worlds.

While reducing the dimension of a high-order dynamical system, preserving essential characteristics of the original model, such as stability, is desirable. Namely, stability is a key aspect in many control systems, especially those for safety-critical applications. In (model-based) balanced truncation techniques, stability preservation is achieved globally for linear systems [15] and for systems with local nonlinearities after satisfying certain conditions [9]. In addition, stability is preserved locally in balancing of nonlinear systems [38]. In data-based MOR, stability enforcement on the trained model helps the trained model to behave more robustly to inputs absent in the training data. Well-known data-based techniques such as Proper Orthogonal Decomposition (POD) [16] and empirical balancing [11] do not necessarily (provably) preserve stability and may lead to unstable system models.

Machine Learning (ML) has recently been used for imitating dynamical system responses [37]. The most used structure in ML to generate dynamical models is the Recurrent Neural Network (RNN). In the approximation of dynamical systems by RNN, the one-step ahead prediction error is typically minimized, rather than the long-term output error between the original model and the predicted model. This might yield an unstable model for which small errors accumulate under recursive simulations [29]. RNNs are used to identify nonlinear systems [4]; however, the model class is restrictive and the identified weights, during training, should evolve in a specific way to retain stability. Stability of RNNs has received attention in many works. However, the stability has only been proved locally for the equilibrium point of the discrete-time RNN by using asymmetric weighting matrices [23,27], which restricts model performance and its trainability. The model performance for stable RNNs has been increased in [35] by applying contraction analysis for nonlinear systems; nonetheless, the model performance is still limited. The RNN model class can be slightly enlarged by only adding more hidden layers and increase the number of neurons within each layer, which in return increases computational cost for training and simulation. In this paper, we try to enlarge the model class by using polynomials. In the literature, RNNs are also used for MOR of systems in which there are no time-varying inputs to the system [37]. This input feature is also fixed while training and testing physics-informed neural networks [33]. In all of these studies, only some parameters are changed and boundary conditions (i.e., inputs) are fixed from one simulation to another. In other words, this trained RNN model acts as a nonlinear interpolation between the solutions at each time instant. Therefore, for these problems, stability is of less importance and the trained model can behave unpredictably for previously unseen inputs [29].

Hence, we pursue the goal of preserving the stability of the model through data-based order reduction in such a way that it is robustly stable for a wide class of inputs. There are many notions of stability for nonlinear systems. Among them, incremental stability notions do not depend on prior knowledge of inputs and trajectories [5,26,30]. The most important notions used in this paper are Incremental Asymptotic Stability (IAS) and Incremental ℓ_2-Gain Stability (IGS). While IAS implies that all solutions converge to each other (and 'forget' initial conditions) for different initial conditions under the same input signal, IGS characterizes the bounded sensitivity of the output to changes in the input [36]. These notions have been recently enforced in system identification for nonlinear systems [39,41], but not yet exploited for MOR problems where the full-state trajectory can be constructed via the reduced-order model, rather than only the output of the system.

The main contributions of this paper are, first, reporting a new observation on the incrementally stable responses of hyperbolic Partial Differential Equations (PDEs), second, proposing an MOR technique for hyperbolic PDEs that preserves incremental stability properties and, third, employing POD to compress the dataset and exploiting it for output estimation and also for constructing the full-state trajectories. It is noteworthy to mention that the proposed method combines data-based MOR approaches and physics of the system. In this setting, the physics of the system is its (incremental) stability properties. To the best of our knowledge, no data-based MOR technique has such a feature of stability preservation. Preserving such properties helps the ROM to behave more robustly to inputs that have not been seen during training.

The present paper extends the results of [39,41] in three aspects. Firstly, we investigate the effectiveness of the proposed method in order reduction of high-dimensional models after discretization of hyperbolic PDEs. To this end, we consider a linear and a nonlinear PDE as our motivating and illustrative case studies. While in [39,41], the system identification depends on the input-output response of a real physical system, our MOR method relies on the solution of a mathematical model of the physical system. Secondly, we introduce other incremental stability notions that are required to be preserved through data-based model complexity reduction. Thirdly, we add an intermediate step (data compression) in the optimization problem to render the procedure computationally feasible for MOR of high-fidelity models, which in addition allows the construction of full-state trajectories. Notably, the model class we assume in this paper is of the form of polynomials, which can cover a large class of nonlinear problems. Other forms of nonlinearities (such as exponential terms) can also be transformed to the polynomial form by lifting transformations [19].

The structure of the paper is as follows. In Sect. 2, mathematical models of the hyperbolic PDEs together with the corresponding discretization technique are presented. Next, the stability properties are defined and the incremental stability properties of the case studies are investigated. In Sect. 3, a general model class for MOR is introduced. In Sect. 4, the constrained optimization framework underlying the proposed data-based MOR approach is presented. Numerical results are shown in Sect. 5. Finally, Sect. 6 concludes the paper.

Notation and preliminaries: Function $\alpha : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ is of class $\mathcal{K}$ if it is continuous and strictly increasing and zero at the origin. If a class-$\mathcal{K}$ function is unbounded, it belongs to the function class $\mathcal{K}_\infty$. A function $\sigma : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ is of class $\mathcal{L}$ if it is continuous, strictly decreasing and $\lim_{n\rightarrow\infty} \sigma(n) = 0$. A function $\beta : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ is of class $\mathcal{KL}$ if it is of class $\mathcal{K}$ in its first argument and of class $\mathcal{L}$ in its second argument. Space $\ell_{2,N}$ consists of all vector functions which are bounded and inherits the norm as $\|u\|^2_{\ell_{2,\mathcal{N}}} = \sum_{n=1}^{\mathcal{N}} |u^n|^2$. A positive definite matrix M is denoted by $M > 0$ (positive semi-definiteness by $M \geq 0$). A multivariate polynomial $z(x)$ is a Sum-Of-Squares (SOS) if there exists polynomials $z_1(x), \ldots, z_c(x)$ such that $z(x) = \sum_{i=1}^{c} z_c^2(x)$.

2 Motivating examples of hyperbolic PDE models and stability features

In this section, we introduce two sets of hyperbolic PDEs, widely used in the MOR community as benchmark problems. These equations represent often-used models for a wide range of physical phenomena, such as fluid mechanics, nonlinear acoustics, gas dynamics, and traffic flows [22]. More complex hyperbolic models are also built upon these motivating examples.

Firstly, the simplest hyperbolic PDE is an advection equation, representing the movement of a wave through a fixed spatial domain in a specific direction. This PDE has applications in physics, engineering and earth sciences [22]. Higher-order one-dimensional hyperbolic systems such as the isothermal Euler equations may be constructed by combining many advection equations with different directions of moving waves [12]. The advection equation we study in this paper is of the following form:

$$\begin{cases} \dfrac{\partial \mathrm{X}}{\partial t} + c\dfrac{\partial \mathrm{X}}{\partial \zeta} = 0, \\ \quad \mathrm{X}(0,\zeta) = \mathrm{X}^0(\zeta), \\ \quad \mathrm{X}(t,0) = \bar{u}(t), \end{cases} \quad , \zeta \in [0, L], t \in [0, T], \tag{1}$$

where $\mathrm{X}(t,\zeta)$ is the conservative variable of the system with initial condition $\mathrm{X}^0(\zeta)$ and $\bar{u}(t)$ represents the time-varying input entering through the boundary condition at $\zeta = 0$. Here, t, ζ, T, L and c are the temporal variable, the spatial variable, the time horizon, the length of the computational domain and the wave velocity, respectively. We define $\bar{y}(t) = \mathrm{X}(t,\zeta_y)$ as the conservative variable at a specific location ζ_y in the spatial domain, to be used later. Notably, this output can be any smooth function of the conservative variable X; however, for this study, we choose this specific function, $\bar{y}(t) = \mathrm{X}(t,\zeta_y)$.

Secondly, Burgers' equation is known as the scalar version of the Navier–Stokes equations [28], which has been the subject of study for many MOR approaches [3,43]. This PDE has been used in many branches of science, such as, e.g., in fluid mechanics and gas dynamics. Burgers' equation reads as follows:

$$\begin{cases} \dfrac{\partial \mathrm{X}}{\partial t} + \dfrac{\partial}{\partial \zeta}(\dfrac{1}{2}\mathrm{X}^2) = 0, \\ \qquad\qquad \mathrm{X}(0, \zeta) = \mathrm{X}^0(\zeta), & , \zeta \in [0, L], t \in [0, T], \\ \qquad\qquad \mathrm{X}(t, 0) = \bar{u}(t), \end{cases} \tag{2}$$

where variables and parameters are as defined in (1). The output function $\bar{y}(t)$ is also defined similarly. Various numerical schemes have been developed to solve (1) and (2). In this study, we use the Rusanov scheme [20] to spatially and temporally discretize the PDEs.

2.1 Spatial and temporal discretization

Finite-volume discretization is commonly employed to solve hyperbolic PDEs. Assume that we are interested in the solution at the i-th spatial grid cell located at the center of the spatial interval $((i-1)\Delta\zeta, i\Delta\zeta), i \in \{1, \ldots, \mathcal{I}\}$, and at the discrete time instant $t^{n+1} = (n+1)\Delta t, n \in \{0, \ldots, \mathcal{N}-1\}$, with Δt and $\Delta\zeta$ denoting the temporal and spatial discretization step sizes. Here, $\mathcal{N}$ is the total number of time-steps and $\mathcal{I}$ is the number of grid cells, which is usually large. First-order Godunov-type schemes numerically solve (1) and (2) by

$$\bar{\mathrm{x}}_i^{n+1} = \bar{\mathrm{x}}_i^n - \frac{\Delta t}{\Delta\zeta}\left(Q(\bar{\mathrm{x}}_i^n, \bar{\mathrm{x}}_{i+1}^n) - Q(\bar{\mathrm{x}}_{i-1}^n, \bar{\mathrm{x}}_i^n)\right), \tag{3}$$

where $\bar{\mathrm{x}}_i^n$ is the spatial average of the conservative variable X over i-th cell at the time instant $t^n := n\Delta t$. The numerical flux function $Q(\cdot, \cdot)$ in (3) is a scheme-dependent function of the conservative variables. There are a variety of schemes in the literature to define $Q(\cdot, \cdot)$. As a special case, the classical Rusanov scheme [20] employs a flux function as below:

$$Q(\bar{\mathrm{x}}_i^n, \bar{\mathrm{x}}_{i+1}^n) = \frac{q(\bar{\mathrm{x}}_{i+1}^n) + q(\bar{\mathrm{x}}_i^n)}{2} \\ -\lambda_{i+1/2}^n(\bar{\mathrm{x}}_{i+1}^n - \bar{\mathrm{x}}_i^n), \tag{4}$$

with

$$q(\bar{\mathrm{x}}_i^n) = c\bar{\mathrm{x}}_i^n, \quad \lambda_{i+1/2}^n = \frac{1}{2}c, \tag{5}$$

for the advection equation (1) and

$$q(\bar{\mathrm{x}}_i^n) = \frac{1}{2}(\bar{\mathrm{x}}_i^n)^2, \quad \lambda_{i+1/2}^n = \frac{1}{2}\max(\bar{\mathrm{x}}_i^n, \bar{\mathrm{x}}_{i+1}^n), \tag{6}$$

for Burgers' equation (2). Here, the operator "max" gives the maximum value of its arguments. Augmenting (3) for all given $i \in \{1, \ldots, \mathcal{I}\}$, we obtain:

$$\begin{cases} \bar{\mathrm{x}}^{n+1} = \bar{f}(\bar{\mathrm{x}}^n, \bar{u}^n), \\ \quad \bar{y}^n = \bar{g}(\bar{\mathrm{x}}^n), \end{cases} \tag{7}$$

where $\bar{\mathrm{x}}^n := [\bar{\mathrm{x}}_1^n \ \cdots \ \bar{\mathrm{x}}_{\mathcal{I}}^n]^T \in \mathbb{R}^{\mathcal{I}}$ with superscript T denoting the transpose action. In addition, $\bar{u}^n := \bar{\mathrm{x}}_0^n$ appears in the right-hand side of equation (3) when we set $i = 1$. We also set $\bar{\mathrm{x}}_{\mathcal{I}+1}^n := \bar{\mathrm{x}}_{\mathcal{I}}^n$ when setting $i = \mathcal{I}$ in (3). In (7), $\bar{f}$ is a nonlinear function which encodes all the relationships in (3) and (4). Moreover, $\bar{y}^n \in \mathbb{R}^m$ with m number of outputs and $\bar{g}$ is a function to define the outputs. In our case study, $\bar{g}(\bar{\mathrm{x}}^n) = Z_{\zeta_y}\bar{\mathrm{x}}^n$, where Z_{ζ_y} is a linear operator which extracts the values of $\bar{\mathrm{x}}^n$ at the location ζ_y. The dimension of functions $\bar{f}, \bar{g}$ is as high as the number of grid cells in case of the scalar PDEs; here, the dimension is $\mathcal{I}$, which is typically large.

Note that the above discretization typically leads to high-dimensional models (due to the large number of grid cells) that obstruct efficient multi-query simulations and controller design. Therefore, we aim to construct a low-order, finite-dimensional, discrete-time model that well approximates the discretized version of (1) and (2), which is (7).

We have observed specific features in the numerical simulation of PDEs (1) and (2). These specific features invoke the definition of incremental stability properties, which are detailed in the next section.

2.2 Stability notions

Here, we give definitions and Lyapunov characterizations of two incremental stability notions: incremental asymptotic stability and incremental ℓ_2-gain stability for system (7).

Definition 1 [40] System (7) is IAS if there exists $\beta \in \mathcal{KL}$ such that $\left\|\bar{\mathrm{x}}_1^n - \bar{\mathrm{x}}_2^n\right\| \leq \beta(\left\|\bar{\mathrm{x}}_1^0 - \bar{\mathrm{x}}_2^0\right\|, n)$ holds for any two initial conditions $\bar{\mathrm{x}}_1^0$ and $\bar{\mathrm{x}}_2^0$ and the corresponding state trajectories $\bar{\mathrm{x}}_1^n$ and $\bar{\mathrm{x}}_2^n$, given the same input sequence $\bar{u}^n$.

Remark 1 In other words, as a result of Definition 1, IAS systems forget their initial conditions. A good example of such systems is hyperbolic (PDE) systems, where initial conditions finally exit the spatial domain, and henceforth, the boundary conditions fully govern the dynamics.

The Lyapunov characterization of IAS systems is given below.

Lemma 1 [40] *System* (7) *is IAS if and only if there exists a smooth incremental Lyapunov function* $V(\bar{\mathrm{x}}_1^n, \bar{\mathrm{x}}_2^n)$ *and functions* $\alpha_1, \alpha_2 \in \mathcal{K}_\infty$ *and* $\alpha_3 : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ *positive definite such that*

$$\begin{aligned} &\alpha_1(|\bar{\mathrm{x}}_1^n - \bar{\mathrm{x}}_2^n|) \leq V(\bar{\mathrm{x}}_1^n, \bar{\mathrm{x}}_2^n) \leq \alpha_2(|\bar{\mathrm{x}}_1^n - \bar{\mathrm{x}}_2^n|), \\ &V(\bar{\mathrm{x}}_1^{n+1}, \bar{\mathrm{x}}_2^{n+1}) - V(\bar{\mathrm{x}}_1^n, \bar{\mathrm{x}}_2^n) \leq -\alpha_3(|\bar{\mathrm{x}}_1^n - \bar{\mathrm{x}}_2^n|), \end{aligned} \tag{8}$$

hold for all $\bar{\mathrm{x}}_1^n, \bar{\mathrm{x}}_2^n$.

Now, we discuss the definition of the IGS property and its corresponding Lyapunov characterization. This definition and the corresponding lemma are discrete-time equivalents of the continuous-time counterparts in [8,36].

Definition 2 System (7) is IGS with an ℓ_2-gain bound of less than γ if, for any two initial conditions $\bar{\mathrm{x}}_1^0$ and $\bar{\mathrm{x}}_2^0$ and two input sequences $\bar{u}_1^n, \bar{u}_2^n \in \ell_{2,\mathcal{N}}$, the corresponding output trajectories $\bar{y}_1^n$ and $\bar{y}_2^n$ satisfy $\|\bar{y}_1 - \bar{y}_2\|^2_{\ell_{2,\mathcal{N}}} \leq \gamma^2 \|\bar{u}_1 - \bar{u}_2\|^2_{\ell_{2,\mathcal{N}}} + \tau(\bar{\mathrm{x}}_1^0, \bar{\mathrm{x}}_2^0)$, with τ a bounded positive function satisfying $\tau(0,0) = 0$.

Remark 2 Loosely speaking, Definition 2 says that if the input signals vary slightly, the output signals also vary only slightly.

Lemma 2 *System (7) is IGS with an* ℓ_2*-gain bound of less than* γ *if and only if there exists a positive semi-definite incremental storage function* $V(\bar{\mathrm{x}}_1^n, \bar{\mathrm{x}}_2^n)$ *such that:*

$$\begin{aligned} &V(\bar{\mathrm{x}}_1^{n+1}, \bar{\mathrm{x}}_2^{n+1}) - V(\bar{\mathrm{x}}_1^n, \bar{\mathrm{x}}_2^n) \leq \gamma^2 |\bar{u}_1^n - \bar{u}_2^n|^2 \\ &\quad - |\bar{y}_1^n - \bar{y}_2^n|^2 \end{aligned} \tag{9}$$

holds for all $\bar{u}_1^n, \bar{u}_2^n$ *and the corresponding* $\bar{\mathrm{x}}_1^n, \bar{\mathrm{x}}_2^n, \bar{y}_1^n, \bar{y}_2^n$ *along trajectories of (7).*

These definitions and lemmas will be extensively used later to *a priori* guarantee the stability of the obtained reduced-order models. In the next section, we report some observations about the responses of the discretized version of the advection and Burgers' equation. These observations, based on the above definitions, suggest incremental stability properties of these PDEs that we wish to preserve in reduced complexity models.

2.3 Evaluation of stability properties

While incremental stability has been thoroughly studied for nonlinear ordinary differential equations [5] and for a class of delay differential equations [10,31], the theoretical characterizations of incremental stability properties for (hyperbolic) PDEs are lacking in the literature and this is a topic for research outside the scope of this paper. Therefore, we consider high-fidelity (high-order) discretized versions of the PDEs obtained via (7) and assess the incremental stability properties for those discretized models. This is also motivated by the fact that such discretized models will be used as a basis for data-based MOR.

2.3.1 Stability properties of the advection equation

Consider the advection equation (1) with a constant initial condition along the spatial domain, $X^0(\zeta) = X^0$. Using the Laplace transform, the analytical solution of (1) is as follows:

$$X(t,\zeta) = \begin{cases} X^0, & t \leq \zeta/c, \\ \bar{u}(t - \zeta/c), & t > \zeta/c. \end{cases} \tag{10}$$

Apparently, at the spatial location ζ after time instant $t = \zeta/c$, the effect of the initial condition vanishes and the state evolution is governed only by the control input $\bar{u}(t)$, in a delayed fashion. Clearly this implies that, independent of the initial condition, all solutions of (1) converge to each other, and therefore, the advection equation exhibits the incremental stability property. Moreover, by changing the inputs, the state evolution experiences the same change as the input after the transient response of the system. This is an indication that the advection equation is also incrementally ℓ_2-gain stable. This behavior can also be observed for the case of spatially varying initial conditions.

Due to the linear nature of (3), (4) and (5), the discretized advection equation in form of (7) leads to a linear discrete-time system. Therefore, the stability notions for the discretized advection equation are only related to the system matrix obtained after spatial and temporal discretization. This system matrix is indeed Schur (if the well-known CFL condition, $c\Delta t/\Delta\zeta < 1$, is satisfied). Therefore, Lemma 1 and 2 can be satisfied with a quadratic Lyapunov and storage functions, and subsequently, the discretized advection equation is guaranteed to be IAS and IGS. This observation for the advection equation motivates us to also study the behavior of nonlinear Burgers' equation since both equations are hyperbolic.

Remark 3 This simple advection equation can easily be represented by a simple delay differential equation, which can be efficiently reduced by other methods. However, the advection equation here serves as a illustrative, though simple, example. This illustrates how incremental stability (and thus the robust stability for a wide class of inputs) can be preserved for a representative system with wave propagation, which is considered challenging [34]. In this regard, it is noteworthy to mention that methods such as shifted-POD [34] are problem specific and mainly work with systems with periodic boundary conditions. In contrast, our method works for any generic bounded input signals at the boundaries of the system.

2.3.2 Stability properties of Burgers' equation

For highly nonlinear infinite-dimensional systems such as Burgers' equation and also the corresponding discretized version, it is hard to assess incremental stability notions analytically since no formal definitions and (Lyapunov-based) characterization of incremental stability properties for such PDEs exist to the best of the authors' knowledge. Hence, we test the hypothesis that Burgers' equation is IAS and IGS by means of comprehensive simulation-based studies. Since the discretized Burgers' equation in (7) is also highly nonlinear, it is hard to find Lyapunov and storage functions for Lemma 1 and 2. However, we use simulations of (7) to infer such stability properties.

To analyze IAS, we numerically solve Burgers' equation with different initial conditions but with the same input signal. Burgers' equation indeed exhibits the behavior of an IAS system (see Fig. 1), since all output solutions converge to each other (and the same is verified for the full state evolution). Three different initial conditions and the corresponding outputs in Fig. 1 are denoted by the subscripts 1,2 and 3. Moreover, this property has been tested for various input signals (which are not reported in this paper) and all results confirm such behavior of the model.

For IGS, we numerically simulate Burgers' equation with slightly different input signals, and as observed in Fig. 2, the corresponding outputs change slightly. This can be an empirical testifier for IGS. Again, this behavior has been observed for many different input signals of different nature. We acknowledge that the above is by no means a proof for IAS/IGS of Burgers' equation. We hope that this inspires further research in the direction of incremental stability for hyperbolic PDEs and emphasize that this is not the focus of this paper, which is on the preservation of such properties for discretized reduced-order models based on high-fidelity discretizations of Burgers' equation.

In the following section, we recall the definition of incremental stability properties and corresponding Lyapunov characterizations for discrete-time models (being either discretized PDE models of the form (7) or their reduced-order variants).

3 Model class for model order reduction

If PDEs (1) and (2) are discretized by a finite difference technique [1], polynomial discrete-time equations will be obtained. As a Galerkin projection in classical MOR preserves the structure of polynomials [32], classical MOR techniques (applied to the discrete model obtained after finite difference discretization) lead to the same structure in the reduced-order model. Therefore, polynomial discrete-time equations are a good candidate for approximating the dynamics in (1) or (2), or their corresponding discretized forms in (7). Moreover, lifting transformations can be applied to general nonlinear systems to convert them to a polynomial form [19]. Therefore, we focus on polynomial discrete-time models within the proposed MOR approach and use the incremental stability properties for such models.

Hence, smooth, nonlinear, implicit discrete-time dynamical systems of the following form with $\hat{\mathrm{x}}^n \in \mathbb{R}^{\hat{\mathcal{I}}}$ are considered for the low-order model to be trained:

$$\begin{cases} h(\hat{\mathrm{x}}^{n+1}) = f(\hat{\mathrm{x}}^n, u^n), \\ \quad\;\; \hat{y}^n = g(\hat{\mathrm{x}}^n) \end{cases} \tag{11}$$

with

$$h(\hat{\mathrm{x}}^{n+1}) := \sum_{i=0}^{k} \theta_i h_i(\hat{\mathrm{x}}^{n+1}), \quad f(\hat{\mathrm{x}}^n, u^n) := \sum_{i=0}^{k} \theta_i f_i(\hat{\mathrm{x}}^n, u^n), \\ := \sum_{i=0}^{k} \theta_i g_i(\hat{\mathrm{x}}^n), \tag{12}$$

where $\theta = \{\theta_i, i = \{0, \ldots, k\}\}$ is the vector of coefficients of the polynomials h_i, f_i, g_i of fixed degrees, which linearly parameterizes the polynomials h, f, g. The coefficients θ_i are the decision variables to be tuned

[1] Not the Rusanov scheme due to the "max" operator.

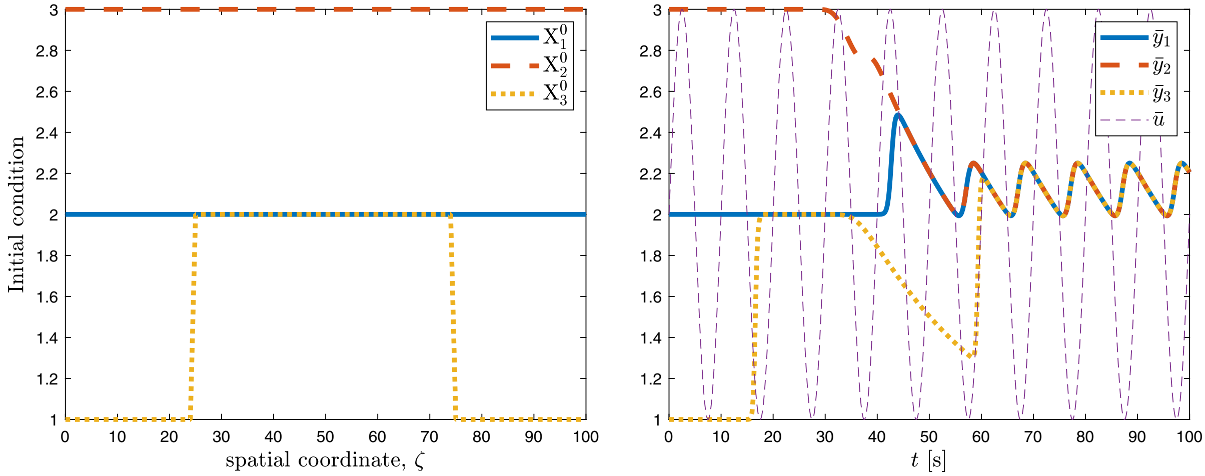

Fig. 1 Different initial conditions for Burgers' equation (left), input signal and output signal corresponding to different initial conditions (right)

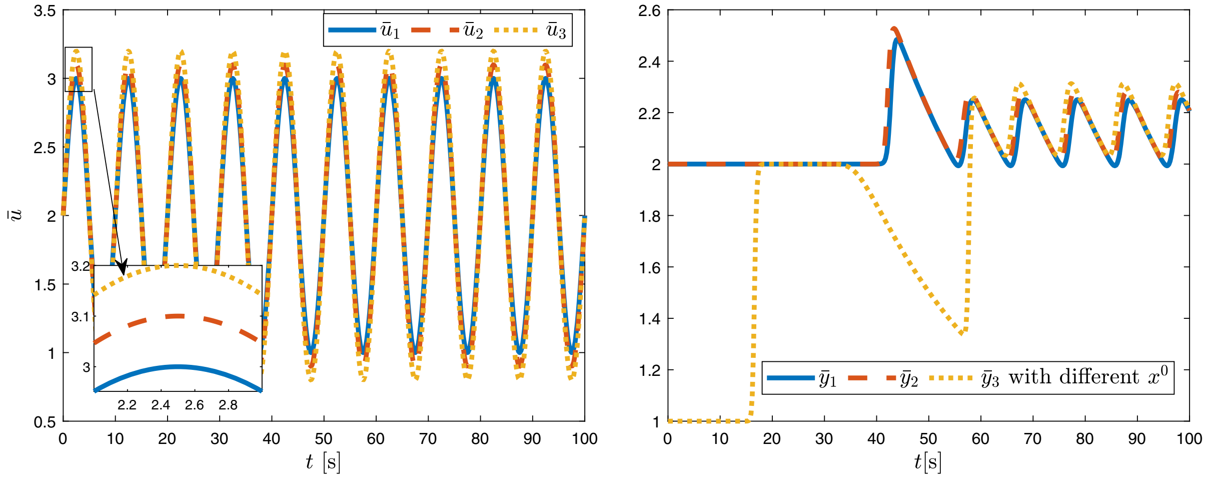

Fig. 2 Slightly different input signals (left), slight change in the output of Burgers' equation (right)

later to ensure model quality and the preservation of stability properties. The number of these coefficients, $k+1$, increases by increasing the polynomial orders of h, f, g (which increase the number of functions h_i, f_i, g_i). Note that we choose h_i, f_i, g_i from a library of the combination of all possible polynomials below a fixed degree. Moreover, $\hat{\mathrm{x}}^n$ is the approximation of x^n, which is the scaled and compressed version of $\bar{\mathrm{x}}^n$ obtained via numerical solution of (7) at discrete time instant t^n. Finally, u^n is the discrete scaled version of $\bar{u}(t)$ in (1) and (2). Notably, scaling helps the trained model to generalize better to a broader set of inputs (to be introduced in Sect. 4.1). It should be noted that the dimension of $\hat{\mathrm{x}}^n$, $f(\cdot,\cdot)$ and $g(\cdot)$ is low, namely $\hat{\mathcal{I}}$, much smaller than $\mathcal{I}$.

In the next section, we introduce the proposed data-based model reduction approach, where we require that the constructed reduced-order model satisfies one of the incremental stability conditions.

4 Data-based model order reduction with incremental stability guarantees

In this section, we explain the steps toward building a reduced-order model from data. In the first step, we generate snapshots of the discrete model (7) invoked by a rich (persistently exciting) input data. Here, we aim at approximating the full-state trajectory of the original model (7) based on the states of the reduced-order model (11). As the dimension of (7) is large (namely $\mathcal{I}$) and the dimension of (11) should be low (namely $\hat{\mathcal{I}}$), data compression should be performed to render our approach computationally feasible. From this compressed data, we construct a low-order model of the form (11) such that (11) is guaranteed to be IAS or IGS and also accurately represents the state and output evolution of the original system (model quality). To compare the state trajectories of the two models, the state trajectory of the reduced-order model (11) should be lifted to the dimension of the original model (7). This is explained in Sect. 4.1. In Sect. 4.2, we formulate the model quality in terms of an objective function to be minimized and the incremental stability guarantee as constraints, jointly leading to a constrained optimization problem used as basis for construction of the reduced-order model.

4.1 Data compression

The objective of data compression in this section is twofold. First, the state evolution of system (7) is of high-dimensional nature. The data compression procedure is employed to reduce the size of the trained model. Second, to compare the trajectories of the full-order model and the reduced-order model, we lift the states of the reduced-order model to the dimension of the original model by using the reverse procedure of the data compression. This is similar to the encoder–decoder notion in Convolutional Neural Networks (CNN) [13]. On the snapshots $\bar{X} = [\bar{\mathrm{x}}^1, \ldots, \bar{\mathrm{x}}^{\mathcal{N}}] \in \mathbb{R}^{\mathcal{I}\times\mathcal{N}}$ (recall $\bar{\mathrm{x}}^n = [\bar{\mathrm{x}}^n_1, \ldots, \bar{\mathrm{x}}^n_{\mathcal{I}}]^T$) corresponding to the training input $\bar{u}^n$, we apply POD [17] to compress the data and also obtain basis functions ϕ according to:

$$\bar{X} = USV \rightarrow \phi = U(:, 1 : \hat{\mathcal{I}}) \tag{13}$$

where $\hat{\mathcal{I}}$ is the desired number of states in the reduced-order model (11), which specifies the dimension of the reduced-order model and ϕ represents the first $\hat{\mathcal{I}}$ columns of U. In (13), $U \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ is a unitary matrix, $S \in \mathbb{R}^{\mathcal{I}\times\mathcal{N}}$ is a rectangular diagonal matrix with singular values of the matrix $\bar{X}$ on its diagonal and $V \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}}$ is again a unitary matrix. Then, the compressed version $\bar{\mathrm{x}}^n_c \in \mathbb{R}^{\hat{\mathcal{I}}}$ of $\bar{\mathrm{x}}^n$ obtained from (7) would be

$$\bar{\mathrm{x}}^n_c = \phi^T \bar{\mathrm{x}}^n, \tag{14}$$

for each time instant n. The output data $\bar{y}^n$ for training are generated directly by (7).

It is common to normalize the input and output signals in the machine learning community to enhance the generalization of the trained model. For the training to preserve IAS, we use the maximum over time of input signals, compressed states (obtained from (14)) and outputs in the training data for normalization. The training data and validation data are normalized as follows:

$$\begin{aligned} &\mathrm{x}^n = \bar{\mathrm{x}}^n_c / \bar{C}_{\mathrm{x}}, \quad y^n = \bar{y}^n / \bar{C}_y, \\ &u^n = \bar{u}^n / \bar{C}_u, \end{aligned} \tag{15}$$

where $\bar{u}^n$ is the discretized version of $\bar{u}(t)$ and $\bar{C}_{\mathrm{x}}$, $\bar{C}_y$ and $\bar{C}_u$ are, respectively, the maximum over all time instants and all elements of $\bar{\mathrm{x}}^n_c$ (compressed states), $\bar{y}^n$ (output of (7)) and $\bar{u}^n$ (inputs of (7)):

$$\bar{C}_{\mathrm{x}} = \max_{n,i} \bar{\mathrm{x}}^n_{c,i}, \quad \bar{C}_y = \max_{n,i} \bar{y}^n_i, \bar{C}_u = \max_{n} \bar{u}^n. \tag{16}$$

Notably, in this study, we only consider positive signals. During training to preserve IGS, we replace (16) with the maximum values of the data used to find γ. We define function g in (11) for later use (see Theorem 3):

$$g(\hat{\mathrm{x}}^n) := \left(\bar{C}_{\mathrm{x}} / \bar{C}_y\right) \phi_y \hat{\mathrm{x}}^n, \tag{17}$$

where $\phi_y \in \mathbb{R}^{m\times\hat{\mathcal{I}}}$ is a matrix containing the value of basis functions ϕ at the output location ζ_y (recall m is the number of outputs), i.e., $\phi_y = Z_{\zeta_y}\phi$ with Z_{ζ_y} defined for (7).

In the next section, we formulate the optimization problem formalizing the data-based MOR approach, preserving incremental stability properties.

4.2 Constrained optimization problem for MOR

In this section, we first introduce an objective function expressing the model quality and second formulate stability properties as constraints to the optimization problem.

4.2.1 Convex cost function for model quality

Let us consider the scaled and compressed snapshots x^n and scaled output y^n corresponding to the input signal

$\bar{u}^n$ generated by (7) and (15) for each PDE. We aim to minimize the one-step ahead prediction cost J_s defined as follows:

$$J_s = \overbrace{\sum_{n=2}^{\mathcal{N}} \left(\left| h(\mathrm{x}^n) - f(\mathrm{x}^{n-1}, u^{n-1}) \right|^2 \right)}^{J_{s1}} + \sum_{n=1}^{\mathcal{N}} \left(\left| y^n - g(\mathrm{x}^n) \right|^2 \right), \tag{18}$$

by choosing optimization variables θ_i, the polynomial coefficients parametrizing the functions h, f, and g as in (12). We will pursue such optimization of model quality while preserving incremental stability properties of (11). The objective function J_s is convex with respect to the decision variables θ_i since functions h, f, g are linearly parametrized as in (12).

The convex objective function (18), however, does not take into account the correct estimation of the compressed and scaled states while we do have such simulation data on the state evolution at our disposal in the context of MOR. In the next section, we propose an alternative to the objective function (18) that considers this aspect.

4.2.2 Convex upper bound for one-step ahead prediction cost

It has been shown in [41] that objective function (18) leads to a bias in the trained trajectories. Here, we introduce an alternative for J_{s1} (first part in (18)) such that the scaled and compressed version of the state evolution (generated by the high-fidelity discretized model (7) followed by (15)) is also accounted for in the objective function. To find the alternative, we introduce the following minimization problem (to be used later to replace the minimization of J_{s1} in (18)):

$$\begin{aligned} \min_{\theta, \hat{\mathrm{x}}^n} \quad & \left| \mathrm{x}^n - \hat{\mathrm{x}}^n \right|^2, \\ \text{s.t.} \quad & h(\hat{\mathrm{x}}^n) = f(\mathrm{x}^{n-1}, u^{n-1}). \end{aligned} \tag{19}$$

The above constrained minimization problem is equivalent to the minimization of J_{s1} as it has the same objective and feasible set. The constraint in (19) is, however, not jointly convex in θ and $\hat{\mathrm{x}}$. To find a *convex* approximation to this problem, we use Lagrangian relaxation similar to [41] and define

$$J^n := \sup_{\hat{\mathrm{x}}^n} \left\{ \left| \mathrm{x}^n - \hat{\mathrm{x}}^n \right|^2 - 2 \left(\lambda^n (\hat{\mathrm{x}}^n) \right)^T \left(h(\hat{\mathrm{x}}^n) - f(\mathrm{x}^{n-1}, u^{n-1}) \right) \right\}, \tag{20}$$

where $\lambda^n(\hat{\mathrm{x}}^n) = \hat{\mathrm{x}}^n - \mathrm{x}^n$ is a candidate Lagrange multiplier [39]. This choice of multiplier simplifies computation of the supremum, see (24), and ensures perfect model recovery when the scaled and compressed snapshots x^n are truly generated from a model of the form (11). However, other choices for this multiplier can be analyzed, which will not be pursued here. J^n is truly an upper bound for the objective function in (19) since if $h(\hat{\mathrm{x}}^n) = f(\mathrm{x}^{n-1}, u^{n-1})$ in (20), then J^n equates (19), so the supremum over $\hat{\mathrm{x}}^n$ cannot be smaller than (19). Now, instead of J_{s1}, we define the following objective function [39]:

$$\hat{J} := \sum_{n=2}^{\mathcal{N}} J^n. \tag{21}$$

To find a *convex* upper bound on $\hat{J}$, we use a Sum-Of-Squares (SOS) [21] technique by defining a slack variable s^n such that

$$\left\{ s^n - \sup_{\hat{\mathrm{x}}^n} \left\{ \left| \mathrm{x}^n - \hat{\mathrm{x}}^n \right|^2 - 2 \left(\hat{\mathrm{x}}^n - \mathrm{x}^n \right)^T \left(h(\hat{\mathrm{x}}^n) - f(\mathrm{x}^{n-1}, u^{n-1}) \right) \right\} \right\} \in \mathrm{SOS}, \tag{22}$$

where SOS denotes the cone of sum-of-squares polynomials [39]. It is challenging to find the supremum in the statement in (22). Therefore, we use the following linearization to find the supremum in (22):

$$h(\hat{\mathrm{x}}^n) \approx h(\mathrm{x}^n) + H(\mathrm{x}^n)(\hat{\mathrm{x}}^n - \mathrm{x}^n), \tag{23}$$

where $E(\cdot)$ is the Jacobian of $h(\hat{\mathrm{x}}^n)$ with respect to $\hat{\mathrm{x}}^n$. After substituting (23) in (22), the part with supremum is replaced by

$$\begin{aligned} &\sup_{\hat{\mathrm{x}}^n} \left\{ \left| \mathrm{x}^n - \hat{\mathrm{x}}^n \right|^2 - 2 \left(\hat{\mathrm{x}}^n - \mathrm{x}^n \right)^T \left(h(\hat{\mathrm{x}}^n) - f(\mathrm{x}^{n-1}, u^{n-1}) \right) \right\} \approx \\ &\sup_{\hat{\mathrm{x}}^n} \left\{ \left| \mathrm{x}^n - \hat{\mathrm{x}}^n \right|^2 - 2 \left(\hat{\mathrm{x}}^n - \mathrm{x}^n \right)^T \left(H(\mathrm{x}^n) \left(\hat{\mathrm{x}}^n - \mathrm{x}^n \right) + \epsilon^n \right) \right\} \end{aligned} \tag{24}$$

where $\epsilon^n = h(\mathrm{x}^n) - f(\mathrm{x}^{n-1}, u^{n-1})$. By differentiating with respect to $\hat{\mathrm{x}}^n$, the value of $\hat{\mathrm{x}}^n$ to find the supremum in (24) can be evaluated as follows:

$$\begin{aligned}
&(\mathrm{x}^n - \hat{\mathrm{x}}^n)^T - \left((\mathrm{x}^n - \hat{\mathrm{x}}^n)^T H^T(\mathrm{x}^n) + \epsilon^{nT}\right) - \\
&\quad (\mathrm{x}^n - \hat{\mathrm{x}}^n)^T H(\mathrm{x}^n) = 0 \Rightarrow \\
&(\mathrm{x}^n - \hat{\mathrm{x}}^n)^T \left(\mathrm{I} - H^T(\mathrm{x}^n) - H(\mathrm{x}^n)\right) = \epsilon^{nT} \Rightarrow \\
&(\mathrm{x}^n - \hat{\mathrm{x}}^n)^T = \epsilon^{nT} \left(\mathrm{I} - H^T(\mathrm{x}^n) - H(\mathrm{x}^n)\right)^{-1},
\end{aligned} \tag{25}$$

where I is the identity matrix with appropriate dimensions. By substituting $(\mathrm{x}^n - \hat{\mathrm{x}}^n)^T$ in (24) by $\epsilon^{nT} \left(\mathrm{I} - H^T(\mathrm{x}^n) - H(\mathrm{x}^n)\right)^{-1}$, we have:

$$\begin{aligned}
&\sup_{\hat{\mathrm{x}}^n}\Big\{ |\mathrm{x}^n - \hat{\mathrm{x}}^n|^2 - 2\left(\hat{\mathrm{x}}^n - \mathrm{x}^n\right)^T \\
&\quad \left(h(\hat{\mathrm{x}}^n) - f(\mathrm{x}^{n-1}, u^{n-1})\right)\Big\} \approx \\
&\quad \epsilon^{nT}\left(H(\mathrm{x}^n) + H^T(\mathrm{x}^n) - \mathrm{I}\right)^{-1}\epsilon^n.
\end{aligned} \tag{26}$$

Replacing the supremum value obtained from (26) into (22), we obtain:

$$s^n - \epsilon^{nT}\left(H(\mathrm{x}^n) + H^T(\mathrm{x}^n) - \mathrm{I}\right)^{-1}\epsilon^n \in \mathrm{SOS}. \tag{27}$$

The above term should be always positive to be contained in the SOS polynomials. As an approximation, the constraint (27) on the slack variables s^n after applying Schur decomposition can be written as a Linear Matrix Inequality (LMI):

$$\begin{aligned}
&s^n - \epsilon^{nT}\left(H(\mathrm{x}^n) + H^T(\mathrm{x}^n) - \mathrm{I}\right)^{-1}\epsilon^n \geq 0 \Rightarrow \\
&\quad \begin{bmatrix} s^n & \epsilon^{nT} \\ \epsilon^n & H(\mathrm{x}^n) + H^T(\mathrm{x}^n) - \mathrm{I} \end{bmatrix} \geq 0.
\end{aligned} \tag{28}$$

This LMI is solved using the SOS tools [41]. For this LMI to be solvable, it is required that $H(\mathrm{x}^n) + H^T(\mathrm{x}^n) - \mathrm{I} > 0$. Finally, the convex objective function becomes

$$J = \sum_{n=2}^{N}\left(s^n + \left|y^n - g(\mathrm{x}^n)\right|^2\right). \tag{29}$$

Here, J is an approximate upper bound (not a true upper bound due to the linearization in (23)) of $\hat{J}$. Minimizing J in (29) with the constraint (28) might give a better result compared to minimizing (18) in terms of the closeness of the trajectories obtained from the full-order and the reduced-order models. Objective function (29) makes use of the state evolution while it enforces an approximate bound on the closeness of the state trajectories. Although objective function (18) does not take into account the closeness of the state trajectories, it is a convex function. The effect of objective functions on the trained reduced-order models is compared in Sect. 5.

To enforce incremental stability properties of the trained models, the objective functions (18) and (29) should be constrained by Lyapunov characterizations of such properties, which is made explicit below.

4.2.3 Incremental stability constraint

Here, we will formulate incremental stability characterizations as constraints to the objective functions and we rewrite these stability constraints in a convex manner to convexify the total constrained optimization problem underlying the proposed MOR approach.

Due to the nonlinearity of (11), it is challenging to analyze its (incremental) stability properties. Therefore, in order to assess the incremental stability properties of (11) (to ultimately guarantee that the reduced complexity model that we construct has these properties), we analyze the differential dynamics associated with (11).

To characterize IAS, we consider the differential dynamics:

$$\begin{cases} H(\hat{\mathrm{x}}^{n+1})\Delta^{n+1} = F_{\hat{\mathrm{x}}}(\hat{\mathrm{x}}^n, u^n)\Delta^n, \\ \qquad\qquad \Delta^n_{\hat{y}} = G(\hat{\mathrm{x}}^n, u^n)\Delta^n, \end{cases} \tag{30}$$

with Δ and $\Delta_{\hat{y}}$ the infinitesimal variations between two neighboring state and output trajectories, respectively, of the original system (11). Here, $H = \partial h/\partial \hat{\mathrm{x}}$, $F_{\hat{\mathrm{x}}} = \partial f/\partial \hat{\mathrm{x}}$ and $G = \partial g/\partial \hat{\mathrm{x}}$. Notably, the stability of the differential system (30) and the incremental stability of the primal system (11) are equivalent for smooth systems [14,42]. As a consequence, instead of constructing an incremental Lyapunov function that satisfies (8), we search for a differential Lyapunov function $V(\hat{\mathrm{x}}, \Delta)$ that satisfies the inequality

$$V(\hat{\mathrm{x}}^{n+1}, \Delta^{n+1}) - V(\hat{\mathrm{x}}^n, \Delta^n) \leq -\alpha_3(\left|\Delta^n\right|), \tag{31}$$

for all $\hat{\mathrm{x}}^{n+1}$, $\hat{\mathrm{x}}^n$ satisfying (11) and all Δ^{n+1}, Δ^n satisfying (30). Hence, if the dissipation inequality (31) holds for all solutions of (11) and the differential dynamics (30), then the model (11) is IAS. The next theorem provides a sufficient condition for IAS of (11).

Theorem 1 *Assume there exists a $P = P^T > 0$ and a $\mu > 0$ such that the following LMI is satisfied for any pair of $(\hat{\mathrm{x}}^n, \Delta^n)$:*

$$M_{IAS} := \begin{bmatrix} H + H^T - P - \mu I & F_{\hat{\mathrm{x}}}^T \\ F_{\hat{\mathrm{x}}} & P \end{bmatrix} \geq 0. \tag{32}$$

Then, (11) is IAS.

Proof By substituting the quadratic differential Lyapunov function $V(\hat{x}, \Delta) = \Delta^T H(\hat{x})^T P^{-1} H(\hat{x})\Delta$ and $\alpha_3(|\Delta|) = \mu\Delta^T\Delta$ in (31), and by taking into account (30), we have that

$$\Delta^T \left(F_{\hat{x}}^T P^{-1} F_{\hat{x}} - H^T P^{-1} H\right)\Delta \leq -\mu\Delta^T\Delta.$$

Using the inequality $-H^T P^{-1} H \leq P - H - H^T$, the above inequality is implied by the inequality

$$H + H^T - P - F_{\hat{x}}^T P^{-1} F_{\hat{x}} - \mu \mathrm{I} \geq 0.$$

Using a Schur decomposition around the first term leads to the desired LMI (32). □

Remark 4 It should be noted that the degree of polynomials in (11) is fixed before starting the optimization problem. Therefore, the Schur decomposition of LMI (32) is a polynomial of a fixed degree.

The differential dynamics used to study IGS is as follows:

$$\begin{cases} H(\hat{x}^{n+1})\Delta^{n+1} = F_{\hat{x}}(\hat{x}^n, u^n)\Delta^n + F_u(\hat{x}^n, u^n)\Delta_u^n, \\ \Delta_{\hat{y}}^n = G(\hat{x}^n)\Delta^n, \end{cases} \tag{33}$$

with $F_u = \partial f/\partial u$ and Δ_u the infinitesimal variation between two neighboring input signals. Notably, the ℓ_2-gain of the differential system (33) is less than γ if and only if the incremental ℓ_2-gain of the primal system (11) is less than γ [14,42]. As a consequence, instead of constructing an incremental storage function that satisfies (9), we search for a differential storage function $V(\hat{x}, \Delta)$ that satisfies the inequality

$$V(\hat{x}^{n+1}, \Delta^{n+1}) - V(\hat{x}^n, \Delta^n) \leq \gamma^2 \left|\Delta_u^n\right|^2 - \left|G(\hat{x}^n)\Delta^n\right|^2. \tag{34}$$

In the following exposition, we provide two sets of LMIs representing sufficient conditions for incremental ℓ_2-gain stability of (11). The first LMI condition is conservative but does not restrict the choice for the output function g in (11). The second condition is less conservative but imposes restrictions on the output function g to generate a convex stability constraint.

Theorem 2 *Assume there exists a $P = P^T > 0$ such that the two following LMIs are satisfied for any pairs of $(\hat{x}^n, \Delta^n)$ and (u^n, Δ_u^n):*

$$\begin{bmatrix} H + H^T - P & \sqrt{2}F_{\hat{x}}^T & G^T \\ \sqrt{2}F_{\hat{x}} & P & \mathbf{0}^T \\ G & \mathbf{0} & I \end{bmatrix} \geq 0,$$

$$\begin{bmatrix} \gamma^2 I & \sqrt{2}F_u^T \\ \sqrt{2}F_u & P \end{bmatrix} \geq 0. \tag{35}$$

Then, system (11) is IGS with incremental ℓ_2-gain γ (where $\mathbf{0}$ is a zero matrix of appropriate dimension).

Proof Let us employ the quadratic candidate storage function $V(\hat{x}, \Delta) = \left|H(\hat{x})\Delta\right|_{P^{-1}} = \Delta^T H(\hat{x})^T P^{-1} H(\hat{x})\Delta$ in (34), which leads to the inequality

$$|H\Delta|_{P^{-1}}^2 - |F_{\hat{x}}\Delta + F_u\Delta_u|_{P^{-1}}^2 + \gamma^2 \left|\Delta_u\right|^2 - \left|G\Delta\right|^2 \geq 0. \tag{36}$$

Exploiting the inequality $|F_{\hat{x}}\Delta + F_u\Delta_u|_{P^{-1}}^2 \leq 2|F_{\hat{x}}\Delta|_{P^{-1}}^2 + 2|F_u\Delta_u|_{P^{-1}}^2$ in the inequality above leads to

$$\Delta^T \left(H^T P^{-1} H - 2F_{\hat{x}}^T P^{-1} F_{\hat{x}} - G^T G\right)\Delta + \Delta_u^T \left(\gamma^2 \mathrm{I} - 2F_u^T P^{-1} F_u\right)\Delta_u \geq 0.$$

If the matrices in between brackets in the first and the second terms in the above inequality are both positive definite, then the full inequality is satisfied. Rewriting these two conditions via a Schur decomposition leads to the LMIs in (35). □

Theorem 3 *Suppose g is a given function and is not parameterized. If there exists a $P = P^T > 0$ such that the following LMI is satisfied for any pairs of $(\hat{x}^n, \Delta^n)$ and (u^n, Δ_u^n):*

$$M_{IGS} := \begin{bmatrix} H + H^T - P - G^T G & \mathbf{0}^T & F_{\hat{x}}^T \\ \mathbf{0} & \gamma^2 I & F_u^T \\ F_{\hat{x}} & F_u & P \end{bmatrix} \geq 0, \tag{37}$$

then (11) is IGS with incremental ℓ_2-gain less than γ.

Proof Recall (36) in Theorem 2:

$$|H\Delta|_{P^{-1}}^2 - |F_{\hat{x}}\Delta + F_u\Delta_u|_{P^{-1}}^2 + \gamma^2 \left|\Delta_u\right|^2 - \left|G\Delta\right|^2 \geq 0.$$

Detailing the above inequality leads to:

$$\Delta^T \left(H^T P^{-1} H - F_{\hat{x}}^T P^{-1} F_{\hat{x}} - G^T G\right)\Delta + \Delta_u^T \left(\gamma^2 \mathrm{I} - F_u^T P^{-1} F_u\right)\Delta_u - \Delta_u^T F_u^T P^{-1} F_{\hat{x}}\Delta - \Delta^T F_{\hat{x}}^T P^{-1} F_u \Delta_u \geq 0.$$

This inequality can be written as follows:

$$\begin{bmatrix}\Delta\\ \Delta_u\end{bmatrix}^T \begin{bmatrix} H^T P^{-1} H - F_{\hat{\mathrm{x}}}^T P^{-1} F_{\hat{\mathrm{x}}} - G^T G & -F_{\hat{\mathrm{x}}}^T P^{-1} F_u \\ -F_u^T P^{-1} F_{\hat{\mathrm{x}}} & \gamma^2 \mathrm{I} - F_u^T P^{-1} F_u \end{bmatrix} \begin{bmatrix}\Delta\\ \Delta_u\end{bmatrix} \geq 0.$$

Exploiting the Schur decomposition for the middle matrix with respect to the first argument leads to:

$$H^T P^{-1} H - F_{\hat{\mathrm{x}}}^T P^{-1} F_{\hat{\mathrm{x}}} - G^T G - F_{\hat{\mathrm{x}}}^T P^{-1} F_u \left(\gamma^2 \mathrm{I} - F_u^T P^{-1} F_u\right)^{-1} F_u^T P^{-1} F_{\hat{\mathrm{x}}} \geq 0. \tag{38}$$

By denoting $M := \gamma^2 \mathrm{I} - F_u^T P^{-1} F_u$, we can use the following matrix inverse relation:

$$\begin{bmatrix}\gamma^2 \mathrm{I} & F_u^T \\ F_u & P\end{bmatrix}^{-1} = \begin{bmatrix} M^{-1} & -M^{-1} F_u^T P^{-1} \\ -P^{-1} F_u M^{-1} & P^{-1} + P^{-1} F_u M^{-1} F_u^T P^{-1} \end{bmatrix}. \tag{39}$$

By using (39), a Schur decomposition of the following matrix gives the same results as (38):

$$\begin{bmatrix} H^T P^{-1} H - G^T G & \mathbf{0}^T & F_{\hat{\mathrm{x}}}^T \\ \mathbf{0} & \gamma^2 \mathrm{I} & F_u^T \\ F_{\hat{\mathrm{x}}} & F_u & P \end{bmatrix} \geq 0.$$

To yield an LMI, we again use the inequality $-H^T P^{-1} H \leq P - H - H^T$ and we obtain the desired LMI (37). □

Remark 5 To be able to satisfy these LMIs, the degree of the polynomial in h should be greater than or equal to the degree of the polynomial in f. This is also detected by the optimization technique where, e.g., the highest identified polynomials in h with nonzero coefficient θ_i were higher than the highest identified polynomials in f with nonzero coefficient θ_i. Therefore, to avoid heavy computations, we set the degree of polynomials in h always equal to or greater than the degree of polynomials in f. Moreover, to solve the LMI (37), consider Remark 4.

Remark 6 By satisfying all stability LMIs, we automatically obtain $H + H^T - P \geq 0$ (note the diagonal terms in all LMIs should be positive definite). This ensures the bijectivity of h and therefore well-posedness of the reduced-order model (11) [41].

Remark 7 Theorem 2 enforces highly conservative stability constraints, which lead to rather unsatisfactory simulation results. Therefore, for IGS, we only study Theorem 3.

Remark 8 LMIs (32) and (37), in, respectively, Theorems 1 and 3, need to be satisfied for any pairs of $(\hat{\mathrm{x}}^n, \Delta^n)$ and (u^n, Δ_u^n), which challenges checking these conditions. To solve this problem, these LMIs are solved using SOS techniques [25], where expansion of LMIs is compared with a combination of polynomials with only even degrees and with positive coefficients, i.e., we restrict the Schur decomposition of the LMIs to be an SOS polynomial [7,21]. Although SOS polynomials are only a subset of all non-negative polynomials, using this method we can tune the free parameters to satisfy those LMIs globally for all the mentioned pairs. There are some software packages that ensure the SOS nature of a polynomial via converting the problem into a semi-definite programming problem [1,25]. As LMIs (32) and (37) are dependent on the pairs mentioned in the respective theorems, we solve $z^T M_{IAS} z \in SOS$ ($z^T M_{IGS} z \in SOS$), where z is a vector of slack dummy variables with known order of polynomials of the pairs. Then the SOS software packages find M_{IAS} (M_{IGS}) to be semi-positive definite and therefore satisfy the LMIs.

To summarize, we solve four problems in Sect. 5 for both the advection and Burgers' equation:
Problem 1: Minimize objective function in (18) constrained to (32) to obtain an IAS system.
Problem 1*: Minimize objective function in (29) constrained to (28) and (32) to obtain an IAS system.
Problem 2: Minimize objective function in (18) constrained to (37) and a fixed output function g as in (17) to obtain an IGS system.
Problem 2*: Minimize objective function in (29) constrained to (28) and (37) and a fixed output function g as in (17) to obtain an IGS system.

In the next section, we compare the trained models obtained from these four different problems, which assign different stability notions through different optimization problems. Then, we decide which of these problems lead to a more flexible reduced-order model (in terms of responding correctly to various input signals) and compare the effect of different objective functions explained in Sect. 4.2.

5 Numerical Results

In this section, we present two case studies: the advection equation and Burgers' equation. For each case study, we solve Problems 1, 1*, 2, 2* to construct accurate reduced-order models with incremental stability certificates. In Sect. 5.1, we first show how we compute the incremental ℓ_2-gain of the original model, to enable enforcing the same incremental ℓ_2-gain on the trained model. Then, we train and validate implicit models as in (11) to imitate the response of the discretized version of (1) and (2) via (7). We also compare the response of the trained models with reduced-order models obtained after the classical POD-Galerkin approach. For the advection equation, the POD-Galerkin with 10 POD basis function is used as explained in [2]. For Burgers' equation, we compare the results with the POD-Galerkin method combined with Empirical Interpolation Method (EIM) as explained in [3] with 10 POD basis functions and 10 EIM basis functions. These 10 basis functions yield a good accuracy in comparison with the full-order model solution. The trained reduced-order model is also of dimension 10. Finally, the claimed stability properties on the trained model are tested empirically. It should be noted that POD-Galerkin techniques do not guarantee stability and we compare its results with our method that guarantees stability.

In the numerical examples, we set $L = 100$ m, $c = 10$ m/s, $T = 100$ s, $\mathcal{N} = 1000$ (we collect simulation data every 0.1 s), $\mathcal{I} = 1000$, $\hat{\mathcal{I}} = 10$, $\mu = 10^{-10}$ with $\Delta t = 0.01$ s, $\Delta\zeta = 0.1$ m in (1), (2) and (3). Notably, all figures show the original variables, not the scaled ones. In case of the advection and Burgers' equation, we choose $\bar{g}(\bar{\mathrm{x}}^n) = [0 \cdots 0\ 1]\bar{\mathrm{x}}^n$. The initial condition for the simulations is $\mathrm{X}^0(\zeta) = \bar{u}(0)$.

5.1 Computation of ℓ_2-gain bound

For the advection equation, the incremental ℓ_2-gain bound is equal to the $\mathcal{H}_\infty$-norm of the linear state-space system obtained after discretization [18], due to linearity. In our numerical example, this gain is equal to 1. Since this gain is computed from the state-space equations, we do not apply data normalization as in Sect. 4.1 for the advection equation test case.

Since it is challenging to analytically obtain ℓ_2-gain bounds for Burgers' equation, we simulate Burgers' equation with many different inputs consisting of sinusoidal and step function contributions. Then, we compare the outputs for every two input pairs and compute the smallest incremental ℓ_2-gain bound γ. As shown in Fig. 3, the incremental ℓ_2-gain bound of Burgers' equation is estimated to be 1.78. In our training process, we consider $\gamma = 2$; in other words, we aim to construct a reduced-order model that is IGS with a guaranteed incremental ℓ_2 gain of 2, which is hence hardly conservative.

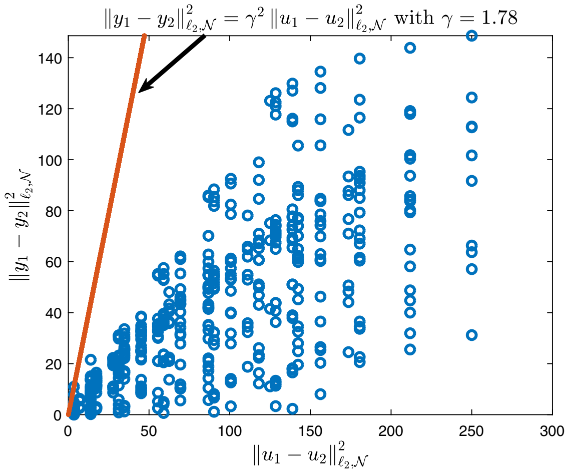

Fig. 3 Empirical evaluation of the ℓ_2-gain bound of Burgers' equation, blue circles: value of $\frac{\|y_1-y_2\|^2_{\ell_2,\mathcal{N}}}{\|u_1-u_2\|^2_{\ell_2,\mathcal{N}}}$ for long simulation with different inputs, red line: a straight line with the slope of max $\frac{\|y_1-y_2\|^2_{\ell_2,\mathcal{N}}}{\|u_1-u_2\|^2_{\ell_2,\mathcal{N}}}$ among many different simulations

Remark 9 Since the LMIs used for model reduction are conservative, for the IGS case in the optimization problem, we use two different values for γ, one equal to the gain of the original system, the other higher than that of the original system. In linear test cases, we compare the gains of the original system and the trained reduced-order model.

5.2 Training and validation results

In this section, extensive simulation studies are carried out for the four problems defined before. Simulations are divided into two sections for the advection and Burgers' equation. The validation and training errors are reported as below:

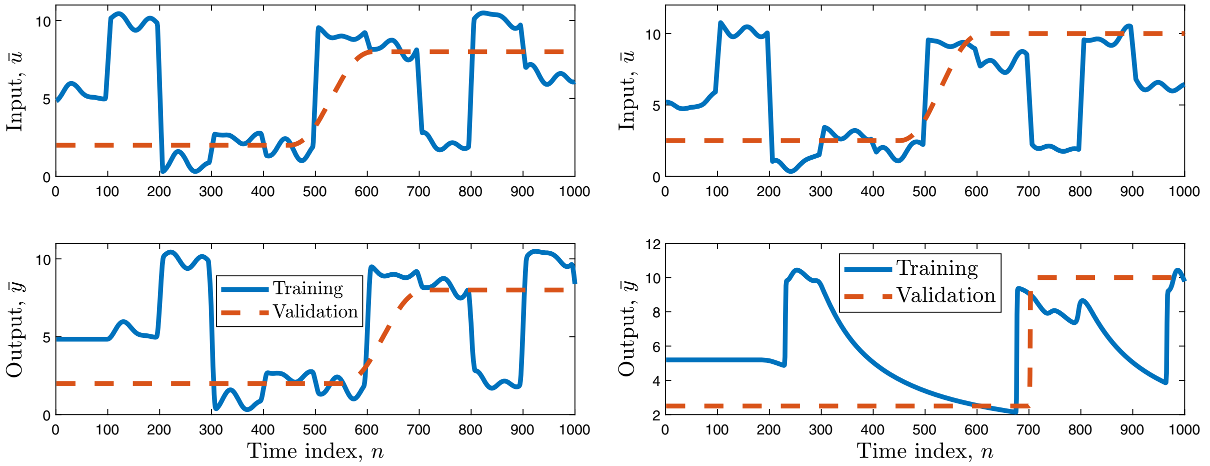

Fig. 4 Input and output signals for training and validation of the advection equation (left) and Burgers' equation (right)

$$
\text{error}\% = \frac{\|y - \hat{y}\|_{\ell_{2,\mathcal{N}}}}{\|y\|_{\ell_{2,\mathcal{N}}}} \times 100. \tag{40}
$$

The training data include an input signal as a combination of multi-sinusoidal and steps signals. Since the interested inputs for our case are a combination of steps (in fluid mechanics, the boundary conditions are usually changed in this manner), we validate our results against a step input.

5.2.1 Advection equation

Left side of Fig. 4 shows the training and validation input and output for the advection equation. Snapshots corresponding to the training input signal are used to determine basis functions for compressing the data, see Sect. 4.1, and also for applying POD-Galerkin.

Since the advection equation is linear and linear systems are included in the model class (11), we only train a linear system for this test case. Results of training and validation of a linear implicit model for the advection equation are shown in Figs. 5 and 6. Table 1 reports the training and validation errors (40) for the four problems and also the POD-Galerkin approach.

Problems 1 and 1* show a good agreement with the training and validation data. Notably, Problem 1* generates a linear model that outperforms the POD-Galerkin approach both for the training and validation data set while guaranteeing IAS. In contrast, POD-Galerkin approach does not guarantee such stability property.

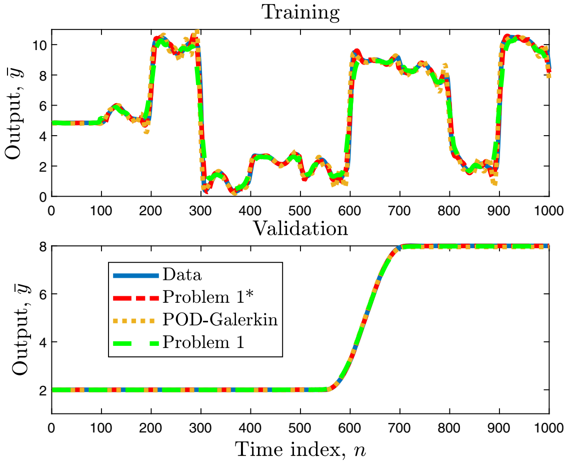

Fig. 5 Training (top) and validation (bottom) results for the advection equation, Problems 1 and 1*

Remark 10 Notably, POD is optimal in the sense that it minimizes the difference between a set of snapshots and their projection on a lower-dimensional space. It is not optimal in the sense of difference between the snapshots of the full-order and reduced-order models.

Hence, this shows that we can construct accurate reduced-order models with incremental stability certificates using the proposed approach. Notably, the dimension of the original model is 1000 while the dimension of the reduced-order model is 10. An issue with Problem 2* is mainly due to the conservativeness in the objective function (29) (using an upper bound on the real objective function, rather than using the objective

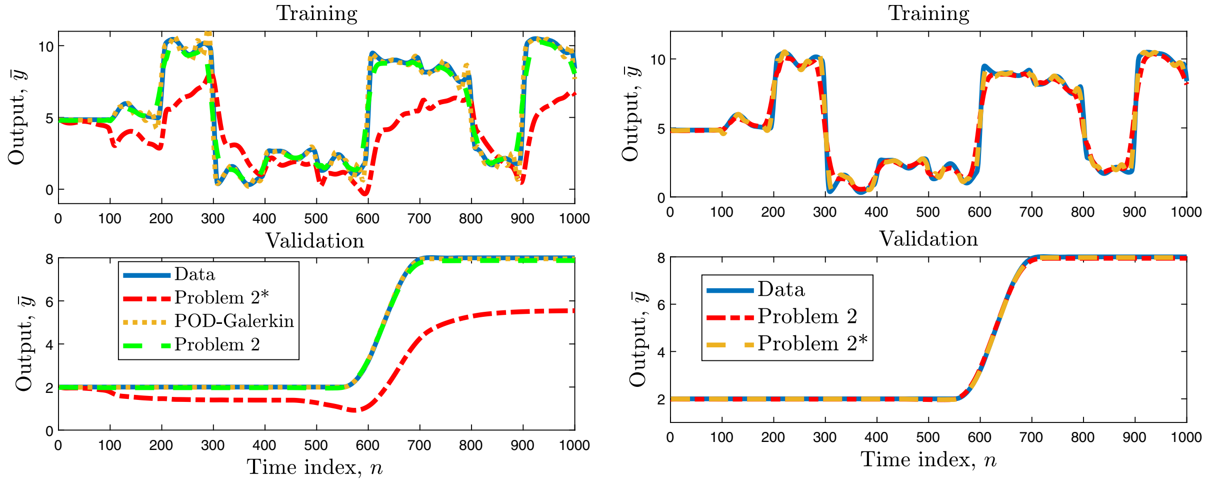

Fig. 6 Training (top) and validation (bottom) results for the advection equation, Problems 2 and 2*, left: $\gamma = 1$, right: $\gamma = 2$ during the training process

Table 1 Training and validation error, advection equation (POD-Galerkin:PG)

	Problem 1	Problem 1*	Problem 2	Problem 2*	PG
			$\gamma = 1(\ell_2$-gain)	$\gamma = 1(\ell_2$-gain)	(ℓ_2-gain)
			$\gamma = 2(\ell_2$-gain)	$\gamma = 2(\ell_2$-gain)	
Training	0.47%	0.12%	0.84%(0.98)	18.47%(0.69)	0.36%
			0.86%(0.991)	0.14% (0.998)	(0.999)
Validation	0.00013%	0.0000045%	0.028%	14.45%	0.00032%

function itself) and consequently underestimating the ℓ_2-gain bound.

Although we set $\gamma = 1$ in (37), the incremental ℓ_2-gain bounds obtained after training linear reduced-order models for Problem 2 and 2* are, respectively, 0.98 and 0.69. Due to the conservativeness mentioned above, the trained model would have a lower gain which restricts the generality of the trained model. If we assign a higher value for the ℓ_2-gain during the optimization problem, the optimization problem might not choose a model that has a higher gain than the original model since the data was generated with the original system. To compare better, we perform another set of optimization with $\gamma = 2$ in (37) without modifying other steps.

Table 1 also reports the training and validation errors (40) together with the incrementally ℓ_2-gain bounds of the trained models with $\gamma = 2$. Although we set a higher gain, the optimization process did not choose a higher value than the original one (for Problem 2 and 2*, we get $\gamma = 0.991$ and $\gamma = 0.998$, respectively). Right side of Fig. 6 shows the performance of Problems 2 and 2* with $\gamma = 2$, where Problem 2* has improved significantly.

5.2.2 *Burgers' equation*

Right side of Fig. 4 shows the non-scaled version of the training and validation input and output for Burgers' equation. The solution to the discretized version of (2) based on (7) to the training input is used to determine basis functions for compressing the data and applying POD-Galerkin and also for computing EIM basis functions.

Since Burgers' equation is quadratic, the function $\bar{f}$ in the discretized version will be quadratic. Moreover, the max operators inside (4) renders (7) more complicated and this model class is not in (11) (note Remark 5 that for satisfying stability conditions we need polynomials of higher degree in h than f in (11)). Therefore, we train many different models of different polynomial orders for this test case and compare the results. If n_h denotes the polynomial degree in h (similarly for

Table 2 Training and validation error for different problems with different polynomial orders and POD-Galerkin (PG) error, Burgers' equation

	Orders	(1,1,1) (%)	(2,1,1) (%)	(2,2,1) (%)	(3,1,1) (%)	(3,2,1) (%)	(3,3,1) (%)	PG (%)
1	Training	3.80	3.80	3.46	0.98	0.82	0.82	2.65
	Validation	2.38	2.37	2.5	3.72	2.33	1.55	1.00
1*	Training	4.42	4.42	3.90	0.98	0.89	0.90	2.65
	Validation	2.17	2.17	2.64	1.98	2.18	1.75	1.00
2	Training	4.25	4.25	4.25	1.43	1.20	1.20	2.65
	Validation	2.51	2.51	2.51	2.52	1.91	1.91	1.00
2*	Training	3.89	3.89	3.89	1.35	1.26	1.26	2.65
	Validation	2.40	2.40	2.40	1.42	1.43	1.43	1.00

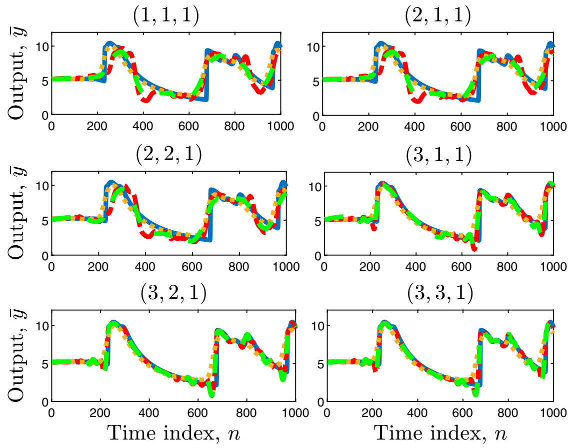

Fig. 7 Training results of Burgers' equation for Problem 1 and 1*, Data (blue solid line), Problem 1 (green dashed line), Problem 1* (red dashed-dotted line), POD-Galerkin (yellow dotted line)

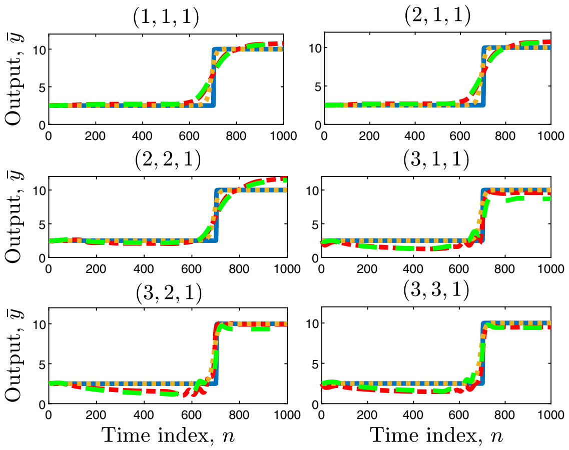

Fig. 8 Validation results of Burgers' equation for Problem 1 and 1*, Data (blue solid line), Problem 1 (green dashed line), Problem 1* (red dashed-dotted line), POD-Galerkin (yellow dotted line)

n_f and n_g), we train six models with $(n_h, n_f, n_g) = \{(1, 1, 1), (2, 1, 1), (2, 2, 1), (3, 1, 1), (3, 2, 1), (3, 3, 1)\}$. Table 2 reports the training and validation error (40) for Burgers' equation. Again, it should be noted that POD-Galerkin method does not guarantee stability.

Figures 7 and 8 show the training and validation results for Problems 1 and 1*. Results show a promising application of our method to this nonlinear problem. It seems that by increasing the polynomial orders, the reduced-order model (11) can adapt better to the highly nonlinear behavior of the actual model (7). Specifically, models (3, 3, 1) for Problems 1 and 1* have close training and validation error. Both Problems 1 and 1* work effectively in this test case.

In Figs. 9 and 10, results of training and validation for Problem 2 and 2* are reported. The gain bounds for the linear system obtained from training of Burgers' equation via Problem 2 and 2* are, respectively, 1.06 and 1.07. Since we did not have any problem with these sets of training and validation data set, we did not increase the gain during training. For this test case, orders (3, 2, 1) work the best for Problems 2 and 2*. The training and validation results are closer in case of Problem 2*, which confirms its generalization power over Problem 2.

The required simulation time for training and validation for Problem 1 is reported in Table 3. All simulations are carried out in MATLAB R2021a on a HP ZBook Studio G5 laptop equipped with Intel(R) Core(TM) i7-9750H CPU with 16 GB RAM and 1 core

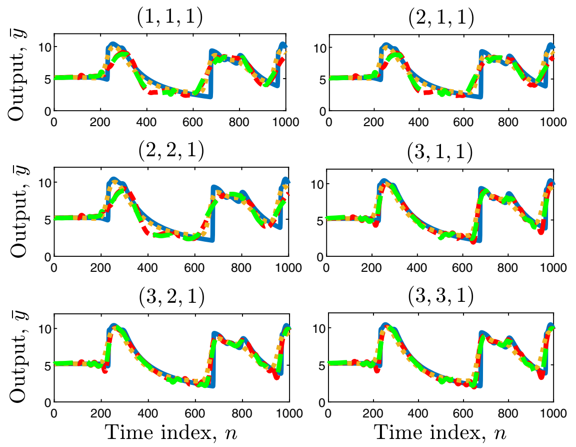

Fig. 9 Training results for Burgers' equation, Problem 2 and 2*, blue solid line: Data, green dashed line: Problem 2, red dashed-dotted line: Problem 2*, yellow dotted line: POD-Galerkin

running at 2.60 GHz. The simulation time for different problems is similar and this only varies significantly by changing the polynomial orders.

5.3 Testing the required stability properties

Since the trained advection equation is linear, all stability properties can be extracted from the state-space equation, which as expected confirmed IAS and IGS of all trained models for the advection equation. Here, we test the four trained models of Burgers' equation for Problems 1, 1*, 2 and 2*. For the initial conditions and inputs shown in Fig. 1, the trained IAS models respond as shown in Fig. 11. As observed in this figure, the IAS property for Problems 1 and 1* is clearly inherited. To check IGS in Problems 2 and 2*, different inputs and initial conditions used in Fig. 2 are used to excite the trained IGS models. Apparent from Fig. 12, the IGS property of Burgers' equation is preserved in the reduced-order model.

5.4 Discussion

According to the reported results, our approach provides accurate reduced-order models with incremental stability certificates. In the test cases we carried out, the accuracy is comparable with the POD-Galerkin approach with the extra property of incremental stability. These incremental stability certificates are instrumental in providing robustness of the trained models for broader classes of inputs (as shown in the validation results). The generated training input is a persistently exciting signal while the validation input is the one which is usually encountered in the real application of the advection and Burgers' equation. We also have tested the model behavior in case of inputs never used during the training, and the testing inputs were sufficiently different from the ones used in the training phase. The results were comparable to the POD-Galerkin with the additional property of preserving incremental stability.

As reported in Table 2, the training error and validation error are much closer in Problem 2*. This confirms that Problem 2* leads to models behaving more robustly by changing the inputs. However, it should be assured that the original model has such stability properties.

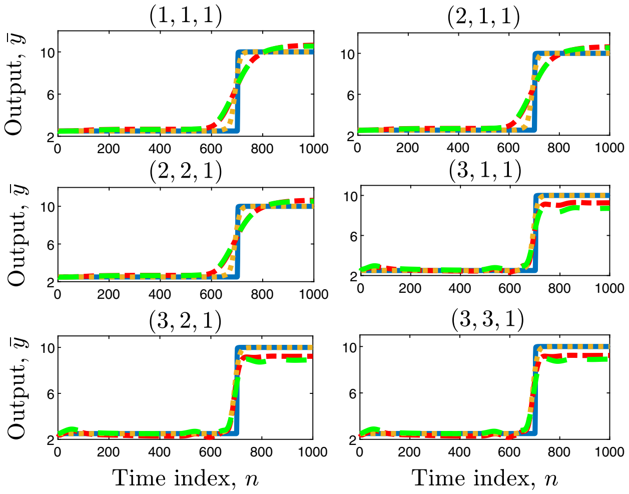

Fig. 10 Validation results for Burgers' equation, Problem 2 and 2*, blue solid line: Data, green dashed line: Problem 2, red dashed-dotted line: Problem 2*, yellow dotted line: POD-Galerkin

Table 3 Training and validation time in minutes for Burgers' equation for Problem 1

Orders	(1,1,1)	(2,1,1)	(2,2,1)	(3,1,1)	(3,2,1)	(3,3,1)
Training	0.43	2.59	2.60	133.63	233.42	250.23
Validation	0.05	0.56	0.57	11.79	12.18	13.12

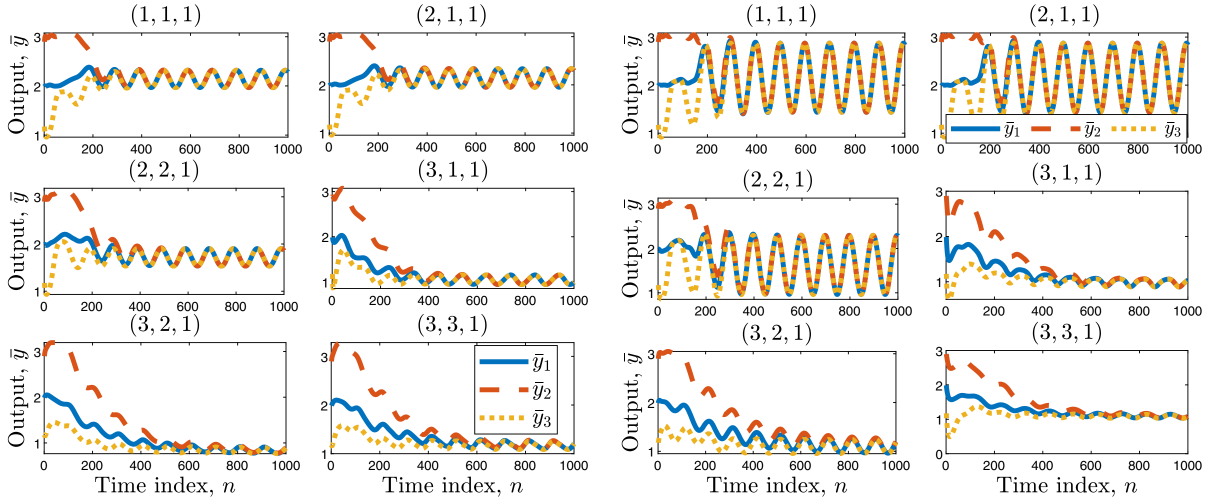

Fig. 11 Simulation of the trained models of Problems 1 (left) and 1* (right) for input and initial conditions shown in Fig. 1

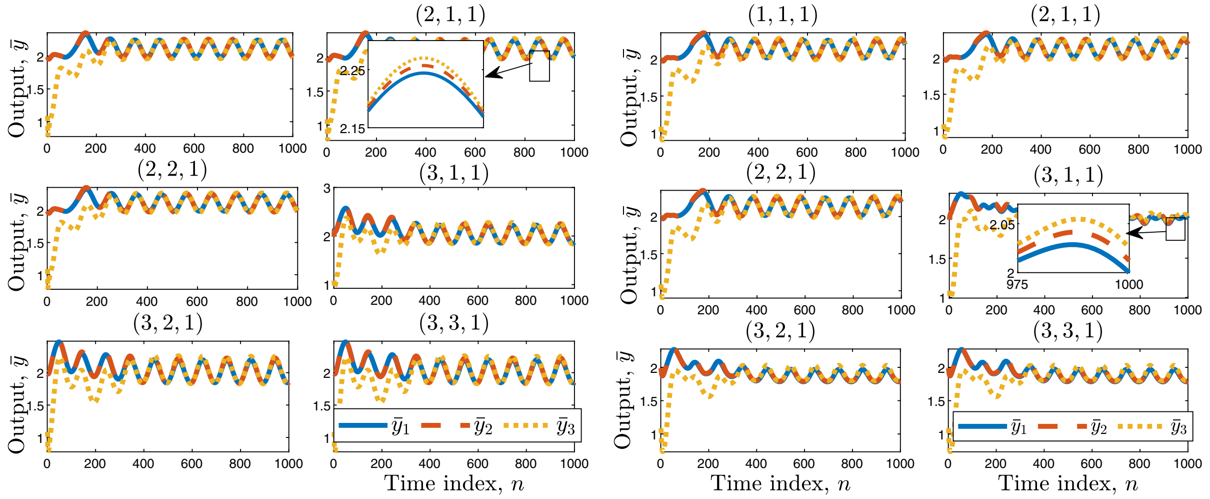

Fig. 12 Simulation of the trained models of Problems 2 (left) and 2* (right) for input and initial conditions shown in Fig. 2

6 Conclusions

Preserving stability of the reduced-order model is a key feature for safety-critical applications. The incremental stability properties help the reduced-order model to behave more robustly to general input signals. In this paper, we presented a data-based model order reduction approach, which preserves the incremental stability features (incremental asymptotic stability and incremental ℓ_2-gain stability), for two relevant hyperbolic PDE models: advection equation and Burgers' equation. First, it has been argued that these models exhibit incremental stability properties. This observation motivated to develop a data-based model reduction approach that guaranteed the constructed model to preserve these incremental stability properties. These incremental stability properties, which are valid for all bounded inputs, help the reduced-order model to be also valid (in terms of preserving these stability characteristics) when the inputs are not present in the training dataset. We developed a (constrained) optimization-based approach for data-driven model reduction to construct accurate low-complexity models preserving incremental stability properties. The proposed approach is comparable to POD-Galerkin approach in terms of accuracy with the additional advantage of incremental stability enforcement. Therefore, this approach can be used when POD-Galerkin approach yields an unstable model. This method can be extended to other systems of hyperbolic equations of higher complexity, which is investigated by the authors.

Funding This research has been carried out in the HYDRA project, which has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 675731.

Availability of data and material The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Code availability The code generated during and/or analyzed during the current study is available from the corresponding author on reasonable request.

References

1. Papachristodoulou, A., Anderson, J., Valmorbida, G., Prajna, S., Seiler, P., Parrilo, P.A.: SOSTOOLS: Sum of squares optimization toolbox for MATLAB. arxiv:1310.4716 (2013). Available from http://www.eng.ox.ac.uk/control/sostools, http://www.cds.caltech.edu/sostools and http://www.mit.edu/~parrilo/sostools
2. Abbasi, M.H., Iapichino, L., Besselink, B., Schilders, W., van de Wouw, N.: Error estimation in reduced basis method for systems with time-varying and nonlinear boundary conditions. Comput. Methods Appl. Mech. Eng. **360**, 112688 (2020)
3. Abbasi, M.H., Iapichino, L., Besselink, B., Schilders, W.H.A., van de Wouw, N.: Error estimates for model order reduction of Burgers' equation. IFAC-PapersOnLine **53**(2), 5609–5616 (2020)
4. Ahn, C.K.: $l_2 - l_\infty$ nonlinear system identification via recurrent neural networks. Nonlinear Dyn. **62**, 543–552 (2010)
5. Angeli, D.: A Lyapunov approach to incremental stability properties. IEEE Trans. Autom. Control **47**(3), 410–421 (2002)
6. Antoulas, A.: Approximation of large-scale dynamical systems. Society for Industrial and Applied Mathematics (2005)
7. Arcak, M., Meissen, C., Packard, A.: Network of Dissipative Systems. Springer Briefs in Electrical and Computer Engineering (2016)
8. Besselink, B., van de Wouw, N., Nijmeijer, H.: Model reduction of nonlinear systems with bounded incremental l_2 gain. In: IEEE 50th Conference on Decision and Control and European Control Conference (CDC-ECC), Orlando, FL, USA, pp. 7170–7175 (2011)
9. Besselink, B., van de Wouw, N., Nijmeijer, H.: Model reduction for nonlinear systems with incremental gain or passivity properties. Automatica **49**, 861–872 (2013)
10. Chaillet, A., Pogromsky, A., Ruffer, B.: A Razumikhin approach for the incremental stability of delayed nonlinear systems. In: 52nd IEEE Conference on Decision and Control CDC 2013,Dec 2013, Florence, Italy, pp. 1596–1601 (2013)
11. Condon, M., Ivanov, R.: Empirical balanced truncation of nonlinear systems. J. Nonlinear Sci. **14**, 405–414 (2004)
12. Cuong, D.H., Thanh, M.D.: A Godunov-type scheme for the isentropic model of a fluid flow in a nozzle with variable cross-section. Appl. Math. Comput. **256**, 602–629 (2015)
13. Fresca, S., Dede, L., Manzoni, A.: A comprehensive deep learning-based approach to reduced order modeling of nonlinear time-dependent parametrized pdes. arXiv:2001.04001 (2020)
14. Fromion, V., Scorletti, G.: A theoretical framework for gain scheduling. Int. J. Robust Nonlinear Control **13**(10), 951–982 (2003)
15. Gugercin, S., Antoulas, A.: A survey of model reduction by balanced truncation and some new results. Int. J. Control **77**(8), 748–766 (2004)
16. Hesthaven, J.S., Rozza, G., Stamm, B.: Certified reduced basis methods for parametrized partial differential equations. Springer Briefs in Mathematics. Springer International Publishing (2016)
17. Kerschen, G., Golinval, J.C., Vakakis, A.F., Bergman, L.A.: The method of proper orthogonal decomposition for dynamical characterization and order reduction of mechanical systems: An overview. Nonlinear Dyn. **41**, 147–169 (2005)
18. Khalil, H.K.: Nonlinear systems, 3, edition Pearson. Upper Saddle River, N.J (2001)
19. Kramer, B., Willcox, K.: Nonlinear model order reduction via lifting transformations and proper orthogonal decomposition. In: American Institute of Aeronautics and Astronautics, vol. 57 (2019)
20. Kurganov, A., Tadmor, E.: New high-resolution central schemes for nonlinear conservation laws and convection-diffusion equations. J. Comput. Phys. **160**(1), 241–282 (2000)
21. Lasserre, J.: Global optimization with polynomials and the problem of moments. SIAM J. Optim. **11**(3), 796–817 (2001)

22. LeVeque, R.: Finite Difference Methods for Ordinary and Partial Differential Equations. Soc. Ind. Appl. Math. (2007)
23. Liang, J., Cao, J., Lam, J.: Convergence of discrete-time recurrentneural networks with variable delay. Int. J. Bifurc. Chaos **15**(2), 581–595 (2005)
24. Ljung, L.: System Identification: Theory for the User, 3rd edn. Prentice-Hall, Englewood Cliffs (1999)
25. Löfberg, J.: Yalmip: a toolbox for modeling and optimization in matlab. In: Proceedings of the CACSD Conference, Taipei, Taiwan (2004)
26. Lohmiller, W., Slotine, J.: On contraction analysis for nonlinear systems. Automatica **34**, 683–696 (1998)
27. Miller, J., Hardt, M.: Stable recurrent models. In: International Conference on Learning Representations (2019)
28. Orlandi, P.: The Burgers equation. In: Fluid Flow Phenomena: A Numerical Toolkit. Fluid Mechanics and Its Applications, pp. 40–50. Springer, Netherlands (2000)
29. Pascanu, R., Mikolov, T., Bengio, Y.: On the difficulty of training recurrent neural networks. In: Proceedings of the 30^{th} International Conference on Machine Learning, Atlanta, GA, USA, vol. 28, pp. 1310–1318 (2013)
30. Pavlov, A., Pogromsky, A., van de Wouw, N., Nijmeijer, H.: Convergent dynamics, a tribute to Boris Pavlovich Demidovich. Syst. Control Lett. **52**(3–4), 257–261 (2004)
31. Pola, G., Pepe, P., Di Benedetto, M., Tabuada, P.: Symbolic models for nonlinear time-delay systems using approximate bisimulations. Syst. Control Lett. **59**(6), 365–373 (2020)
32. Qian, E., Kramer, B., Marques, A., Willcox, K.: Transform & learn: A data-driven approach to nonlinear model reduction. In: AIAA Aviation 2019 Forum (2019)
33. Raissi, M., Perdikaris, P., Karniadakis, G.: Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J. Comput. Phys. **378**, 686–707 (2019)
34. Reiss, J., Schulze, P., Sesterhenn, J., Mehrmann, V.: The shifted proper orthogonal decomposition: a mode decomposition for multiple transport phenomena. SIAM J. Sci. Comput. **40**(3), A1322–A1344 (2018)
35. Revay, M., Manchester, I.: Contracting implicit recurrent neural networks: stable models with improved trainability. In: Proceedings of Machine Learning Research, vol. 120, pp. 1–11 (2020)
36. Romanchuk, B., James, M.: Characterization of the l_p incremental gain for nonlinear systems. In: IEEE 35th Conference on Decision and Control (CDC), Kobe, Japan, pp. 3270–3275 (1996)
37. San, O., Maulik, R., Ahmed, M.: An artificial neural network framework for reduced order modeling of transient flows. Commun. Nonlinear Sci. Numer. Simul. **77**, 271–287 (2019)
38. Scherpen, J.: Balancing for nonlinear systems. Syst. Control Lett. **21**, 143–153 (1993)
39. Tobenkin, M., Manchester, I., Megretski, A.: Convex parameterizations and fidelity bounds for nonlinear identification and reduced-order modelling. IEEE Trans. Autom. Control **62**(7), 3679–3686 (2017)
40. Tran, D., Ruffer, B., Kellett, C.: Convergence properties for discrete-time nonlinear systems. IEEE Trans. Autom. Control **64**(8), 3415–3422 (2019)
41. Umenberger, J., Manchester, I.: Convex bounds for equation error in stable nonlinear identification. IEEE Control Syst. Lett. **3**(1), 73–78 (2019)
42. Verhoek, C., Koelewijn, P., Toth, R.: Convex incremental dissipativity analysis of nonlinear systems. arXiv:2006.14201v1 (2020)
43. Zhang, Y., Feng, L., Li, S., Benner, P.: An efficient output error estimation for model order reduction of parametrized evolution equations. SIAM J. Sci. Comput. **37**(6), B910–B936 (2015)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.