Error estimation in reduced basis method for systems with time-varying and nonlinear boundary conditions

Abbasi, M. H.; Iapichino, L.; Besselink, B.; Schilders, W. H. A.; van de Wouw, N.

Published in:
Computer Methods in Applied Mechanics and Engineering

DOI:
10.1016/j.cma.2019.112688

Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 360 (2020) 112688
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

Error estimation in reduced basis method for systems with time-varying and nonlinear boundary conditions

M.H. Abbasi[a,*], L. Iapichino[a], B. Besselink[b], W.H.A. Schilders[a], N. van de Wouw[c,d]

[a] Department of Mathematics and Computer Science, Eindhoven University of Technology, The Netherlands
[b] Bernoulli Institute for Mathematics, Computer Science and Artificial Intelligence, University of Groningen, The Netherlands
[c] Department of Mechanical Engineering, Eindhoven University of Technology, The Netherlands
[d] Department of Civil, Environmental and Geo-Engineering, University of Minnesota, USA

Received 15 December 2018; received in revised form 3 September 2019; accepted 7 October 2019
Available online 21 October 2019

Abstract

Many physical phenomena, such as mass transport and heat transfer, are modeled by systems of partial differential equations with time-varying and nonlinear boundary conditions. Control inputs and disturbances typically affect the system dynamics at the boundaries and a correct numerical implementation of boundary conditions is therefore crucial. However, numerical simulations of high-order discretized partial differential equations are often too computationally expensive for real-time and many-query analysis. For this reason, model complexity reduction is essential. In this paper, it is shown that the classical reduced basis method is unable to incorporate time-varying and nonlinear boundary conditions. To address this issue, it is shown that, by using a modified surrogate formulation of the reduced basis ansatz combined with a feedback interconnection and a input-related term, the effects of the boundary conditions are accurately described in the reduced-order model. The results are compared with the classical reduced basis method. Unlike the classical method, the modified ansatz incorporates boundary conditions without generating unphysical results at the boundaries. Moreover, a new approximation of the bound and a new estimate for the error induced by model reduction are introduced. The effectiveness of the error measures is studied through simulation case studies and a comparison with existing error bounds and estimates is provided. The proposed approximate error bound gives a finite bound of the actual error, unlike existing error bounds that grow exponentially over time. Finally, the proposed error estimate is more accurate than existing error estimates.

Keywords: Model order reduction; Error estimate; Boundary conditions; Hyperbolic equations; Local nonlinearities; Single-phase flow

1. Introduction

Simulations of high-fidelity dynamic models, especially for real-time and many-query analyses, entail highly expensive computations, rendering direct numerical simulation infeasible. To circumvent the so-called curse of dimensionality in such simulations, model order reduction has been employed in the literature and it is becoming an essential tool for controller design, multi-query, real-time simulations and model-based optimization [1]. Several techniques for generating reduced-order models have been proposed such as, e.g., the reduced basis method [2],

* Corresponding author.
E-mail address: m.h.abbasi@tue.nl (M.H. Abbasi).

https://doi.org/10.1016/j.cma.2019.112688
0045-7825/

balanced truncation [3], moment matching [1], and interpolation techniques based on input–output behavior such as IRKA [4]. In particular, the reduced basis method is an efficient approach for dealing with parameterized systems, where the parameters can be, for instance, the physical parameters of a system, geometry properties or boundary conditions [5].

The reduced basis method consists of an offline and online phase. In the offline phase, the essence of the underlying model is extracted by obtaining a, usually small, problem-specific set of basis functions. This can be carried out by computing the solution to the full-order model for a representative set of parameter values. The basis functions can be computed by different methods; the most popular ones are proper orthogonal decomposition (POD) and the greedy algorithm [6,7]. After generating the basis functions, by exploiting the affine dependency of the full-order model on the varying parameters, a fast, easy-to-compute and reasonably accurate model is built in order to be used in the online phase. In the online phase, the computationally cheap yet accurate model can be solved for each new set of parameters with far less computational effort compared to the full-order model.

While the theory of the reduced basis method for a single, parabolic and elliptic partial differential equation (PDE) has already reached a mature stage [2,8–10], systems of PDEs and single hyperbolic PDEs still present an open research field [11–14]. For hyperbolic problems, shock-capturing model order reduction techniques such as the freezing method [12] and shifted-POD [14] are yet far from generic and still require many problem-specific adaptations and are not considered in this paper.

One critical issue to address in the reduced basis method is the incorporation of the boundary conditions. The reduced basis method is a powerful method for the case of constant boundary conditions. Due to the global nature of the basis functions and their independency over time, handling time- or state-dependent boundary conditions can be challenging [15]. Moreover, dealing with nonlinear and implicit boundary conditions in the reduced-order model of a system of PDEs is still an open issue [16]. Nonetheless, boundary conditions of any system, either induced by the physical effects or by a controller, play a crucial role in the system dynamics.

In the reduced-order model, different approaches for dealing with parameterized Dirichlet boundary conditions have been introduced [16,17]. In addition, the formulation of boundary conditions as a differential algebraic equation (DAE) has been discussed in [18]. Nevertheless, the correct implementation of boundary conditions of a hyperbolic system requires the use of characteristic waves [19,20]. The governing equations of these waves include nonlinear ordinary differential equations (ODEs) [19,21], introducing extra dynamics at the boundaries. This challenges the construction of reduced-order models. So, the correct implementation of the boundaries in the general case should be investigated more thoroughly.

Besides problems occurring in nonlinear, time-varying boundary conditions in the reduced basis method, an easy-to-compute a posteriori error bound (or estimate) is required, firstly, to speed up the basis generation in the greedy algorithm and, secondly, to assess the accuracy of the reduced solution generated by the reduced basis method. Numerous works are directed towards deriving a residual-based error bound (or estimate) for parabolic PDEs [2,7,9,22,23]. The error bounds in the parabolic context are well-developed while very few works have focused on developing an error bound (or estimate) for hyperbolic PDEs [11,24]. The error bound in [11] is dependent on the norm of the state matrix; when this norm is larger than one, the error bound grows exponentially over time and is not useful for assessing the quality of the reduced-order model. The error estimate in [24] is derived by defining a dual system and building a connection between the norm of the residuals and the error in approximating the states. Moreover, this method is also based on the norm of the state matrix and this norm should satisfy some constraints. These constraints are not satisfied in the numerical examples shown in the current paper. So the ultimate problem is that in many cases for hyperbolic problems, no finite error bound can be constructed using existing approaches. Therefore, novel approaches for the construction of error bounds and estimates are needed. Hereto, we first propose a new approximate error bound that does not pertain to the norm of the state matrix. Second, in order to mitigate the conservativeness of the proposed error bound, a new error estimate is introduced by exploiting this error bound. The main contributions of the paper are described below.

In this paper, we propose an efficient strategy for dealing with time-varying and nonlinear boundary conditions. A modified ansatz representation for the reduced basis method, interconnected with a boundary condition solver, is proposed and tested on a hyperbolic system. This method draws inspiration from methods presented in [25–27] in the scope of balanced truncation for nonlinear systems. Furthermore, to certify the accuracy of the reduced-order model, a new approximate error bound is proposed. However, this approximate error bound may still lead to conservative results. To sharpen the error bound, an error estimate is introduced based on the obtained approximate

error bound and the simulations in the offline phase, similar to the one presented in [24], however, in a different norm realization. Two test cases are considered in this paper. First, a linear advection equation containing source terms with time-varying boundary conditions is investigated. The error bound is applied on this system and we show that, for a certain class of inputs, the proposed error bound is one of the sharpest possible error bounds. Second, the linearized isothermal Euler equations with nonlinear, time-varying boundary conditions are considered. The effect of the boundary conditions on the solution of the reduced-order model is investigated. The isothermal Euler equations with the utilized boundary conditions are commonly used in the simulation of gas and liquid flows in pipelines [28] and for hydraulics drilling simulations in single phase cases [26].

The outline of this paper is as follows. In Section 2, the general linear PDE together with the applied finite-volume scheme and the resulting full-order model are explained. In Section 3, the ingredients of the reduced basis method, the proposed ansatz and the reduced-order model are introduced. In Section 4, the error bound is formulated and the connection between the error bound and the error estimate is explained. In Section 5, numerical results for an advection equation with source term and the linearized isothermal Euler equations are presented. Finally, Section 6 closes the paper with some conclusions and perspectives on future work.

2. Problem statement

In this work, the following class of linear PDEs is studied:

$$\begin{cases} \dfrac{\partial u}{\partial t} + J\dfrac{\partial u}{\partial x} = S(u,t,x;\mu), \\ u(0,x;\mu) = u_0(x;\mu), \\ u(t,x_\partial;\mu) = u_\partial(t;\mu), \end{cases} \qquad t \in [0,T], \quad x \in [0,L], \tag{1}$$

where $u = u(t,x;\mu)$ is called conservative variable (since this variable should be conserved in each grid cell after the discretization) at time t and spatial location x for a set of parameters $\mu \in \mathcal{D} \subset \mathbb{R}^b$. Here, $\mathcal{D}$ is the domain of varying parameters and b is the number of varying parameters in the system. This system may also contain fixed parameters that do not vary from one simulation to another in a multi-query context. In addition, $J = J(\mu)$ is the Jacobian of the analytical flux function $f(u;\mu)$, i.e., $J(\mu) = \partial f/\partial u$ (J should be independent of u such that the PDE becomes linear), and $S(u,t,x;\mu)$ is the source term. Moreover, the final time of the simulation and the spatial domain length are denoted by T and L, respectively. To analyze the response, an initial condition $u_0(x;\mu)$ and a set of boundary conditions $u_\partial(t;\mu)$ on the boundary points x_∂ are required, which will be elaborated upon in Sections 5.1.1 and 5.1.2 for the specific cases of an advection equation with source term and the linearized isothermal Euler equations.

Definition 1. The function $S(u,t,x;\mu)$ is affine with respect to parameters μ if it can be written as $S(u,t,x;\mu) = \sum_j S_u^j(u,t,x)\, S_\mu^j(\mu)$, with state-dependent functions $S_u^j(u,t,x)$ and parameter-dependent functions $S_\mu^j(\mu)$.

Assumption 1. In (1), J only depends on the parameters μ, not the states. J and S are affine with respect to parameters. Moreover, the eigenvalues of J are all real and distinct such that the system is hyperbolic. Finally, to have a linear system, all S_u^j in the expansion of Definition 1 are assumed to be a linear function of conservative variables u.

Linear PDEs as in (1) with properties mentioned in Assumption 1 can be discretized with a linear scheme, which is described next.

2.1. Discretization using the upwind scheme

First-order Godunov-type schemes together with forward Euler integration numerically solve PDE (1) by (parameter dependency is not mentioned here for the ease of notation) [29,30]

$$U_i^{n+1} = U_i^n - \frac{\Delta t}{\Delta x}\Big(\mathcal{F}(U_{i+1}^n, U_i^n) - \mathcal{F}(U_i^n, U_{i-1}^n)\Big) + \Delta t\; S(U_i^n, t^n, x_i;\mu), \tag{2}$$

where $U_i^n,\ i = 1,\dots,N_\delta,\ n = 0,\dots,N_t - 1$, is the average of the conservative variables u over the ith spatial grid cell at the time instant $t^n = n\Delta t$, with N_δ the number of spatial grid cells and N_t the number of time steps.

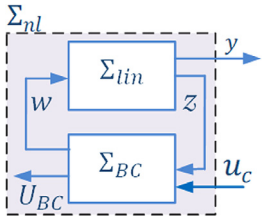

Fig. 1. The schematic representation of the full-order nonlinear model Σ_{nl} consisting of a feedback interconnection of the linear subsystem Σ_{lin} and a boundary subsystem Σ_{BC}.

Also, Δt and Δx denote the temporal and spatial resolution of the discretization. Next, $\mathcal{F}$ is the numerical flux function (discretized counterpart to the analytical flux function f in (1)) that is scheme-dependent. For instance, for the upwind scheme [29,30], the numerical flux function is given by

$$\mathcal{F}(U_{i+1}, U_i) = A^+ U_i + A^- U_{i+1}, \tag{3}$$

where $A^+ = R\Lambda^+ R^{-1}$ and $A^- = R\Lambda^- R^{-1}$ with R matrix composed of the right eigenvectors of the Jacobian matrix J, and

$$\Lambda^+ = \begin{bmatrix} {\lambda_1}^+ & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & {\lambda_r}^+ \end{bmatrix}, \quad \Lambda^- = \begin{bmatrix} {\lambda_1}^- & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & {\lambda_r}^- \end{bmatrix}. \tag{4}$$

Here, $\lambda_k,\ k = 1, \ldots, r$ are the eigenvalues of the Jacobian matrix J with r as the dimension of the system (1). For every $k = 1, \ldots, r$, we denote $\lambda_k^+ = \max(\lambda_k, 0)$ and $\lambda_k^- = \min(\lambda_k, 0)$. If the Jacobian matrix J is not diagonalizable (i.e., Assumption 1 is not valid) , other schemes such as the Lax-Friedrichs or Rusanov scheme [31] can be used.

The finite-volume discretization leads to a high-dimensional (though finite-dimensional) model, which will be referred to as the full-order model. This model is introduced in the upcoming section.

2.2. Full-order model

In many cases, the equations governing the boundary conditions are time-varying, coupled with the state variables, and/or nonlinear. As a result, the full-order model becomes nonlinear even if the internal dynamic is linear. As an example, since the system of PDEs of interest as in (1) is linear and the upwind scheme (3) is also linear, a finite-dimensional linear time-invariant (LTI) system is obtained after discretization. However, substituting $i = 1$ and $i = N_\delta$ in (2), the conservative variables U_0^n and $U_{N_\delta+1}^n$ are required to compute the solutions. These variables are the boundary values that can be regarded as inputs to the LTI system that render the entire system nonlinear. On the other hand, all nonlinearities enter the system at the boundaries. So, it is natural to decompose the full-order nonlinear model Σ_{nl} into a linear subsystem Σ_{lin} (representing the discretized PDEs in the internal domain) and a nonlinear subsystem Σ_{BC} (representing the boundary dynamics). This is schematically depicted in Fig. 1 and the models Σ_{nl}, Σ_{lin} and Σ_{BC} are of the form

$$\Sigma_{nl}: \begin{cases} \Sigma_{lin}: \begin{cases} U^{n+1} = A(\mu)U^n + B(\mu)w^n, \\ y^n = C_y U^n, \\ z^n = C_z U^n, \end{cases} \\ \Sigma_{BC}: \begin{cases} V^{n+1} = G(V^n, z^n, u_c^n; \mu), \\ w^n = \mathcal{G}(V^n, u_c^n; \mu), \\ U_{BC}^n = C_{BC}\left[(V^n)^T \quad (w^n)^T\right]^T. \end{cases} \end{cases} \tag{5}$$

Here, the superscript T denotes the transpose operator. As depicted in Fig. 1 and apparent from (5), the linear subsystem Σ_{lin} solves for the states U^n. The input to this system comes from boundaries which are governed by

Σ_{BC}. This subsystem takes auxiliary outputs z from the linear subsystem and the control inputs u_c from the user and then computes the variables w, which represents the input of the linear dynamics. The actual output y of the system is defined by the user while the auxiliary outputs z are determined based on the interconnection of the two subsystems. In addition, the boundary solver calculates the conservative variables at the boundary location U_{BC}, which will be used in the reduced-order model. Although the boundary values of the conservative variables, U_{BC}, do not play a role in the full-order model, these help us later in the next section to incorporate the boundary conditions in the reduced-order model correctly.

To introduce the notation in (5), $A(\mu) \in \mathbb{R}^{N_\delta \times N_\delta}$, $B(\mu) \in \mathbb{R}^{N_\delta \times m}$, with m the dimension of the inputs w, are parameter-dependent state and input matrices, $U^n \in \mathbb{R}^{N_\delta}$ is the vector of state variables of the linear system Σ_{lin} at time step n. For this system to be stable, $A(\mu)$ should be a Schur matrix (i.e., all eigenvalues of $A(\mu)$ lie in the unit disc in the complex plane). Moreover, V^n is the vector of state variables in the dynamics of the boundary conditions governed by G, $w^n \in \mathbb{R}^m$ is the input of the linear system Σ_{lin}, which is computed by an output function $\mathcal{G}$ in Σ_{BC}. External (control) inputs u_c act at the boundaries. Finally, C_y and C_z are the output matrices for the output of interest y^n and auxiliary outputs z^n. In general, Σ_{nl} in (5) expresses a feedback interconnection between Σ_{lin} and Σ_{BC}. However, in some cases, there is no dynamics occurring at the boundaries and the boundary conditions are specified explicitly over time; and in some other cases, the boundaries are dynamically coupled with the output z^n as in Σ_{BC} in (5).

Clearly the dimension of Σ_{lin} scales with N_δ, which can be extremely large when fine spatial discretization meshes are used. On the other hand, the dimension of Σ_{BC} is typically of low order compared to N_δ. According to this observation, we assume that most of the computational time is allocated for solving Σ_{lin} rather than Σ_{BC}. This assumption is crucial to justify the speedup after model order reduction via the interconnection approach. To reduce the number of equations to be solved in Σ_{lin}, a reduction approach will be implemented on the linear subsystem, while the number of equations in Σ_{BC} is not changed. Moreover, the boundary incorporation in the reduced basis is another challenge that has to be handled. To deal with the complexity of the linear subsystem of the full-order model (5) and the boundary incorporation, the model order reduction technique is introduced next.

Assumption 2. $A(\mu)$ and $B(\mu)$ in the linear subsystem Σ_{lin} in (5) are affine with respect to parameters μ. Moreover, $A(\mu)$ is Schur for all $\mu \in \mathcal{D}$.

Remark 1. Since $A(\mu)$ is Schur for all $\mu \in \mathcal{D}$, Σ_{lin} is stable. However, the stability of Σ_{lin} does not guarantee the stability of Σ_{nl}.

Remark 2. In this study, only 1D problems are considered. For the 2D and 3D case, the number of equations to be solved at the boundary is still much lower than for the internal grids. If the boundaries in the higher-dimensional settings are nonlinear, reduction for simulation is less promising if most of the computational time is allocated in solving the boundary conditions. One alternative in the 2D and 3D cases would be the reformulation of the system as partial differential algebraic equations (PDAEs) and reduce the entire system, which is beyond the scope of this paper. For the reduction of parameter-independent PDAEs based on input–output behavior, refer to [32].

Remark 3. In this study, by using characteristic boundary equations [20,21], boundary dynamics and the internal dynamics become consistent in the full-order model and no boundary layer appears in the solution.

3. Boundary incorporation within the reduced basis method

The reduced basis method [2,11] targets the reduction of parametrized problems requiring repeated evaluations or (faster than) real-time simulations, known as many-query and real-time analysis, respectively. This technique consists of two phases with two different objectives. The first phase is the offline phase, which captures the most dominant characteristics of the problem based on the solution of the physical full-order model, the so-called truth solutions, for specific parameter values. At this stage, the dimension N_δ of the full-order space exploited to find the truth solution is large, leading to expensive and time-consuming simulations. The reduced basis method approximates this space with an N-dimensional subspace (typically $N \ll N_\delta$). After finding this subspace, the required reduced-order operators are defined through Galerkin or Petrov–Galerkin projection onto this N-dimensional subspace. In the online phase, the reduced-order model is generated, rendering simulations

faster as $N \ll N_\delta$ [2]. In other words, the online computational cost depends on N, rather than N_δ. This fact has two advantages. First, the solution in the online phase can be found faster compared to the full-order model, due to the small dimension of the reduced basis space and, second, the basis functions used for the projection are defined in the full-order space and allow to express the reduced basis solution in the full-order space as well. The dimension of the full-order space depends on the number of grid cells of the computational domain and can be increased to attain more accurate results. Even though increasing the number of grid cells increases the offline cost, it does not influence the online cost significantly. We note that Assumption 2 enables such offline-online decomposition [2,33].

Numerically, the reduced basis solution of (5) in the finite-dimensional setting collected for all grid cells at time instant t^n is denoted by $\hat{U}^n(\mu)$. The reduced basis solution, $\hat{U}^n$, is computed as a linear combination of spatial- and time-dependent vectors as

$$\hat{U}^n(\mu) = \sum_{i=1}^{N} a_i^n(\mu)\phi_i, \tag{6}$$

where $\Phi = \{\phi_i,\ i = 1, \ldots, N\}$ is the set of N reduced basis vectors (i.e., reduced basis functions evaluated at the discrete computational domain). These basis vectors are the dominant modes of specific truth solutions of (5) for certain values of the parameters μ. The basis vectors ϕ_i can be obtained by using the POD-greedy algorithm in the offline phase [2,22]. The POD approach to generate ϕ_i is illustrated in Algorithm 1 and POD-greedy is explained in Algorithm 2. The dynamics of the generalized coordinates, $a_i^n(\mu)$, can be obtained by Galerkin or Petrov–Galerkin projection of the problem (5) onto the space Φ.

Algorithm 1: POD algorithm, POD($\boldsymbol{U}, n_{\text{POD}}$)

Input: Snapshots $\boldsymbol{U} = \{U^0, \cdots U^{N_t}\} \in \mathbb{R}^{N_\delta \times N_t}$, number of basis vectors n_{POD}
Output: $\phi_i \in \mathbb{R}^{N_\delta \times n_{\text{POD}}}$
1 Perform a Singular Value Decomposition on the snapshots, $\boldsymbol{U} = MSV$,
2 $\phi_i = M(:, 1 : n_{\text{POD}})$ are the first n_{POD} vectors of the left singular vectors M.

Remark 4. If nonlinearities exist or the system is non-affine with respect to parameters, the empirical interpolation method (EIM) or the discrete empirical interpolation method (DEIM) can be performed on the system to prepare it for Galerkin or Petrov–Galerkin projection onto the space Φ [9,10,34]. However, nonlinear and non-affine systems are not studied in this work.

Remark 5. The dyadic product of (6) may not be suitable for problems with shocks. In this study, the boundary conditions change smoothly and therefore no shock occurs in the system. For systems with shocks, other approaches such as Shifted-POD [14] or the freezing method [12] should be followed; however, their applicability is still limited.

3.1. New ansatz formulation

The ansatz formulation (6) cannot capture time-varying boundary conditions [17]. This is attributed to the global and time-invariant nature of the reduced basis vectors. As an example, POD typically employs basis vectors that are constant over time. As a result, if the boundary values are constant over time and not parameter dependent, the boundary conditions are always satisfied. However, when the boundary conditions change over time, the reduced solution cannot cope with these changes since the coefficients of the basis vectors, a_i^n, are determined by the internal dynamics. This is more important when the system has two time-varying boundary conditions at both ends. For many industrial applications, such as flow inside a pipe [19] and managed pressure drilling (MPD) [26], boundary conditions are coupled with the state variables and are even nonlinear. Since external (control) inputs of such systems also usually act at the boundaries, correct implementation of the boundaries in the reduced-order model is crucial.

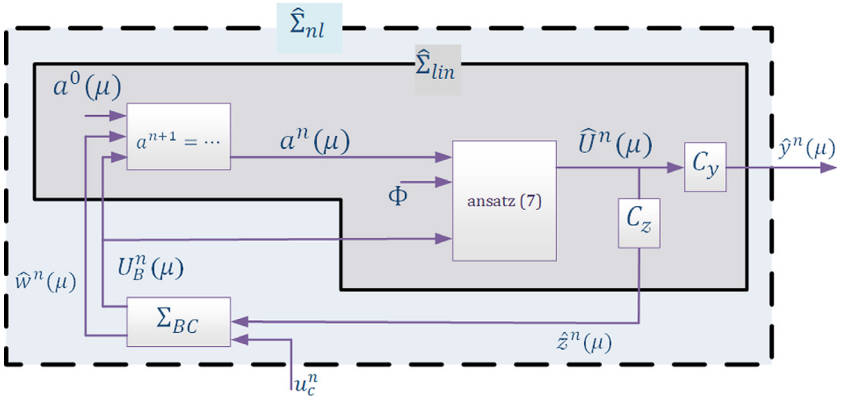

Fig. 2. The schematic representation of handling boundary conditions in the reduced-order model in (9).

To handle the time dependency and nonlinearities associated with the boundaries, the dyadic form of the reduced basis ansatz (6) is changed to

$$\hat{U}^n(\mu) = U_B(\hat{U}^n_{BC}) + \sum_{i=1}^{N} a_i^n(\mu)\phi_i, \tag{7a}$$

$$U_B(\hat{U}^n_{BC})\Big|_{x_\partial} = \hat{U}^n_{BC}, \tag{7b}$$

$$\phi_i|_{x_\partial} = 0, \tag{7c}$$

where $\hat{U}^n_{BC}$ is the reduced solution at specified locations x_∂ on the boundary and $U_B(\hat{U}^n_{BC})$ is a vector (a function evaluated at the discrete computational domain) that enables the reduced basis solution $\hat{U}^n$ to satisfy the boundary conditions. In other words, the vector U_B encodes the exact satisfaction of only the boundary conditions at the reduced level as in (7b) (it can be an interpolation inside the internal domain). Then, (7c) states that ϕ_i should vanish at the location of the specified boundary conditions.

The logic behind choosing U_B is problem-specific and some examples are given in Section 5. In general, we propose to choose U_B by considering a linear interpolation between the boundaries x_∂ of the system (1).

One requirement for the correct incorporation of the boundary conditions is that the basis vectors in (7) should vanish at the boundaries (see (7c)). In order to satisfy this requirement, the snapshots for applying the POD are modified. Instead of applying the POD to the snapshots of the solution $U^n(\mu)$, the POD is applied to the set of modified snapshots $U^{n*}(\mu)$ defined as below,

$$U^{n*}(\mu) = U^n(\mu) - U_B(U^n_{BC}), \tag{8}$$

where U^n_{BC} is the truth solution at the boundaries. The modified snapshots $U^{n*}(\mu)$ vanish at the location of the specified boundaries, enforcing the POD to generate basis vectors that are zero at these boundaries.

The methodology to obtain the reduced-order model with the modified ansatz (7a) at the discrete level is illustrated next.

3.2. Reduced-order model by Galerkin projection

As mentioned earlier, we focus on linear PDEs as in (1), discretized with linear schemes such as the upwind scheme (3). By doing so, on the one hand, the equations governing the internal domain form an LTI system and thus the dimension of the equations can be reduced by the reduced basis method. On the other hand, the dynamics of the boundary conditions are time-variant, nonlinear, and act locally. As the correct implementation of the boundaries is uncompromisable, the number of equations in Σ_{BC} in (5) are not reduced. This method shares some features with the method introduced in [25–27] in the balanced truncation setting and in [18] in the finite-element setting.

The linear subsystem Σ_{lin} in (5) can be reduced to another linear system $\hat{\Sigma}_{lin}$ of lower order by the reduced basis method, but the local and finite boundary equations Σ_{BC} remain full-order. Generally speaking, the local

nonlinearities of the boundary conditions are hard-coded in this approach. To provide some insights on this technique, the schematic view of the method is shown in Fig. 2. This figure illustrates that auxiliary outputs, $\hat{z}^n = C_z \hat{U}^n$, that are necessary for computing the outputs of Σ_{BC}, w^n and $\hat{U}^n_{BC}$ (see also Fig. 1), are provided via a feedback interconnection. Then, the outputs from Σ_{BC} are fed into the reduced linear system $\hat{\Sigma}_{lin}$ to incorporate the effect of the boundary conditions.

To obtain the reduced-order model corresponding to the full-order model (5), we drop the parameter dependency on μ for notational simplicity. The parameter dependency can be included straightforwardly using the affine property of the system, as explained in [11]. After finding the reduced basis space $\Phi \in \mathbb{R}^{N_\delta \times N}$, the Galerkin projection can be executed. After the projection, the states of the reduced system $\hat{\Sigma}_{lin}$ are the generalized coordinates a^n_i. Following the explanation above regarding the reduction strategy and considering (7) and the orthogonality of Φ, we obtain the reduced-order system as follows:

$$\hat{\Sigma}_{nl}: \begin{cases} \hat{\Sigma}_{lin}: \begin{cases} a^{n+1} = \hat{A}_a a^n + \hat{B}\hat{w}^n + \hat{A}_{BC} U^n_B - \Phi^T U^{n+1}_B, \\ \hat{U}^n = U^n_B + \Phi a^n, \\ \hat{y}^n = C_y \hat{U}^n, \\ \hat{z}^n = C_z \hat{U}^n, \end{cases} \\ \Sigma_{BC}: \begin{cases} \hat{V}^{n+1} = G(\hat{V}^n, \hat{z}^n, u^n_c), \\ \hat{w}^n = \mathcal{G}(\hat{V}^n, u^n_c), \\ \hat{U}^n_{BC} = C_{BC}\begin{bmatrix} (\hat{V}^n)^T & (\hat{w}^n)^T \end{bmatrix}^T, \\ U^n_B = U_B(\hat{U}^n_{BC}), \end{cases} \end{cases} \tag{9}$$

where $\hat{A}_a = \Phi^T A \Phi$, $\hat{A}_{BC} = \Phi^T A$, $\hat{B} = \Phi^T B$, $a = [a_1, \ldots, a_N]^T$. The second equation in $\hat{\Sigma}_{lin}$ in (9) is the ansatz (7a). For an example of U_B, see (35) in Section 5.1.2 for the linear isothermal Euler equations. In (9), $\hat{A}_a \in \mathbb{R}^{N \times N}$ and $\hat{B} \in \mathbb{R}^{N \times m}$ are both of low dimension. Recall that N is the number of basis vectors with $N \ll N_\delta$ and m is the dimension of w^n. The initial condition of the system is $a^0 = \Phi^T U^0$ where U^0 is the initial condition of the original system. However, some dependencies on the actual degree of freedom N_δ still exist in (9) due to $\hat{A}_{BC}$, Φ^T and U_B. Resolving this issue is problem-specific, which is illustrated in Sections 5.1.1 and 5.1.2 for two test cases.

Remark 6. In the reduced-order model, an inconsistency between the solution at the internal domain and the solution at the boundaries appears when Galerkin projection is applied (see Section 5.2). To resolve this, the effect of boundary conditions is subtracted from the internal domain and added later on by an input-related term to avoid boundary layers. One can also reformulate the problem with the boundary condition as a DAE problem. Note that in the current study, the boundary conditions are dynamic and not characterized by mere algebraic constraints. In a way, this type of decomposition preserves the DAE structure, but its numerical solution is derived based on the interconnection approach presented here. In addition, this also preserves the differentiation index of the DAE. If the full DAE is reduced, then the dynamics at the boundary might also change and inconsistencies might still appear. Here, only the differential part of the DAE is reduced and the boundary related part is retained in the exact form. Moreover, the numerical solution of the DAEs is hard to obtain as the problem might become stiff. These reasons have motivated us to pursue the decomposition approach.

Remark 7. System (9) still depends on the dimension of the full-order model due to the lifting in the second equation of $\hat{\Sigma}_{lin}$. This lifting can be incorporated in the output equations as $\hat{y}^n = C_y U^n_B + C_y \Phi a^n$ (similar for $\hat{z}^n$) and precomputing $C_y \Phi$ ($C_z \Phi$) for use in the online phase.

In the next section, the bound and estimate for the error induced by replacing the full-order model with the reduced-order model are discussed.

4. Error bound and estimate for the reduced-order model

In order to certify the accuracy of the reduced basis solution and to accelerate the offline phase by appropriate snapshot selection, an error bound (or estimate) has to be provided. In this study, we are interested in a reduced-order

model that provides an accurate prediction of the output y (to be specified by the user of the reduction approach) of the full-order nonlinear model. To derive a bound for the output of a system, the concept of the ℓ_2-gain of the system is exploited.

Definition 2 (*[35]*)**.** The stable system Σ_{lin} as in (5) has a bounded ℓ_2-gain of less than γ from input w to output y if

$$\sum_{n=0}^{\infty} \|y^n\|^2 \leq \gamma^2 \sum_{n=0}^{\infty} \|w^n\|^2, \tag{10}$$

for all trajectories of Σ_{lin} with zero initial condition and where $\|\cdot\|$ is the Euclidean norm. We will use the shorthand notation $\|y\|_{\ell_2} \leq \gamma \|w\|_{\ell_2}$, where $\|y\|_{\ell_2} = \sqrt{\sum_{n=0}^{\infty} \|y^n\|^2}$.

Definition 2 implies that for a bounded input with respect to the ℓ_2-norm, the output is also bounded with respect to the same norm. In other words, γ in the inequality (10) can be interpreted as a bound on the ratio between the energies of output and input. This kind of inequality can be defined for any input–output pair of the system. If the system is LTI and asymptotically stable, as we assume to be the case for Σ_{lin} in (5) for all $\mu \in \mathcal{D}$ (Assumption 2), it can be proved that the ℓ_2-gain of a linear system equates to the $\mathcal{H}_\infty$-norm of the system [36] for which computationally efficient tools are available [37,38]. As the final system before applying the Galerkin projection should be cast into a linear model with respect to the states, the concept of ℓ_2-gain is also applicable in the reduced basis context. Here, we aim to exploit this notion to derive an error bound for substituting the full-order nonlinear model (5) with the reduced-order nonlinear model (9). For developing such an error bound, the dynamics that govern this error is required, which is the topic of Section 4.1. Then, we introduce an expensive yet accurate error bound in Section 4.2.2. As we are interested in easy-to-compute and cheap error *estimates*, a fast error estimate is introduced based on the error bound in Section 4.3.

4.1. Error dynamics

Recall that $\hat{U}^n$ is the result computed by the reduced basis method from (9). As this variable satisfies the reduced-order model, and not exactly the full-order model (5), by substituting $\hat{U}^n$ into (5), a residual denoted by $\mathcal{R}^n \in \mathbb{R}^{N_\delta}$, at each time step, appears in the formulation of the full-order dynamics

$$\Sigma_{nl}: \begin{cases} \Sigma_{lin}: \begin{cases} \hat{U}^{n+1} = A(\mu)\hat{U}^n + B(\mu)\hat{w}^n + \mathcal{R}^n, \\ \hat{y}^n = C_y \hat{U}^n, \\ \hat{z}^n = C_z \hat{U}^n, \end{cases} \\ \Sigma_{BC}: \begin{cases} \hat{V}^{n+1} = G(\hat{V}^n, \hat{z}^n, u_c^n; \mu), \\ \hat{w}^n = \mathcal{G}(\hat{V}^n, u_c^n; \mu), \\ \hat{U}_{BC}^n = C_{BC}\begin{bmatrix} (\hat{V}^n)^T & (\hat{w}^n)^T \end{bmatrix}^T. \end{cases} \end{cases} \tag{11}$$

It should be noted that the dynamics of the boundary conditions Σ_{BC} are fully preserved and no residual appears in their dynamics. By defining the reduction error by $e^n = U^n - \hat{U}^n$, the error dynamics can be obtained by using (11) and (5):

$$\Sigma_{nl}^e: \begin{cases} \Sigma_{lin}^e: \begin{cases} e^{n+1} = A(\mu)e^n + B(\mu)e_w^n - \mathcal{R}^n, \\ e_y^n = C_y e^n, \\ e_z^n = C_z e^n, \end{cases} \\ \Sigma_{BC}^e: \begin{cases} e_V^{n+1} = G(V^n, z^n, u_c^n; \mu) - G(\hat{V}^n, \hat{z}^n, u_c^n; \mu), \\ e_w^n = \mathcal{G}(V^n, u_c^n; \mu) - \mathcal{G}(\hat{V}^n, u_c^n; \mu), \\ e_{U_{BC}}^n = C_{BC}\begin{bmatrix} (e_V^n)^T & (e_w^n)^T \end{bmatrix}^T, \end{cases} \end{cases} \tag{12}$$

where other output errors are denoted as $e_{(\cdot)} = (\cdot) - (\hat{\cdot})$ and $V^n = e_V^n + \hat{V}^n$. Thus, the error dynamics generally contain two inputs. One input is e_w^n, which affects the error dynamics if the boundary values are not estimated perfectly.

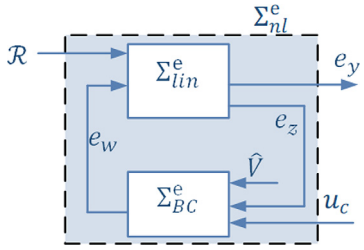

Fig. 3. The schematic representation of the error dynamics (12) consisting of a feedback interconnection of the linear and nonlinear part.

Although the boundary dynamics are exactly preserved, due to the fact that boundary values are determined together with an approximation of the linear model, the boundary values are not exact either, which gives an error e^n_w. The other input of the error dynamics is the residual $\mathcal{R}^n$, which contributes to the error due to the inaccuracy of $\hat{U}^n$ in approximating U^n. Fig. 3 explains the interconnection within (12). Next, bounds on the output error of the interconnected system (12) are introduced.

4.2. ℓ_2 and ℓ_∞ error bounds

As the error dynamics have the dimension of the full-order model, it cannot be used for efficiently determining the error and, therefore, bounds on the output of the error dynamics should be set. For systems with state-dependent boundary conditions, we propose the ℓ_2 error bound and for systems with boundary conditions explicitly stated over time, we propose the ℓ_∞ error bound as below.

4.2.1. ℓ_2 error bound

To impose a bound on the output e_y as in (12) by the ℓ_2-gain theory, we note that Assumption 2 implies the existence of gains $\gamma^{e_y e_w}$ and $\gamma^{e_y \mathcal{R}}$ such that

$$\left\|e_y\right\|_{\ell_2} \leq \gamma^{e_y e_w} \|e_w\|_{\ell_2} + \gamma^{e_y \mathcal{R}} \|\mathcal{R}_s\|_{\ell_2}, \tag{13}$$

where γ^{yu} is the $\mathcal{H}_\infty$-norm (ℓ_2-gain) of the system from input u to output y and $\mathcal{R}_s$ is the two-norm of the residual over the spatial domain at each time instant, i.e., $\mathcal{R}^n_s = \|\mathcal{R}^n\|$.

One ingredient for the calculation of the bound is the ℓ_2-norm of the error induced at the boundaries, $\|e_w\|_{\ell_2}$. This norm can be bounded by the Lipschitz constant of the nonlinear function governing the boundary. For instance in (12), we have:

$$\begin{aligned} e^n_w &= \mathcal{G}(V^n, u^n_c) - \mathcal{G}(\hat{V}^n, u^n_c), \\ \left\|e^n_w\right\| &\leq L_{\mathcal{G}} \left\|e^n_V\right\| \Longrightarrow \|e_w\|_{\ell_2} \leq L_{\mathcal{G}} \|e_V\|_{\ell_2}, \end{aligned} \tag{14}$$

where $L_{\mathcal{G}}$ is the global Lipschitz constant of the nonlinear function $\mathcal{G}(V^n, \cdot)$ with respect to its first argument.

Remark 8. For some cases, the function $\mathcal{G}(V^n, \cdot)$ might not be globally Lipschitz. Then, a local Lipschitz constant can be employed. In this study, we approximate such local Lipschitz constant by

$$L_{\mathcal{G}} \approx \max_n \left(\left| \frac{\partial \mathcal{G}}{\partial V} |_{V=\hat{V}^n} \right| \right), \tag{15}$$

which becomes more accurate if the solutions of the full-order and reduced-order systems are closer. The computation of $\|e_V\|_{\ell_2}$ in (14) is dependent on its governing dynamics G, which will be explained in one of the test cases in Appendix. The assumption of reduced and full solution being close to each other becomes more realistic when the number of basis vectors increases. For a low number of basis vectors, this assumption is not accurate and (13) (combined with (15)) cannot be a true bound on the error. When the number of basis vectors is low, a possible alternative is to restrict the solution domain. Then, during the offline phase, the local Lipschitz constant over this restricted solution manifold can be approximated based on (14) and the available actual error (see Remark 11). In this approach, potentially larger discrepancies between the solutions of the full-order and reduced-order systems are always allowed at the expense of higher $L_{\mathcal{G}}$.

Finally, the last ingredient for computing the error bound is $\|\mathcal{R}^n\|$ and consequently $\|\mathcal{R}_s\|_{\ell_2}$. This computation should also scale with N in the online phase to make the error bound computationally feasible. To analyze this, rearranging the first equation in (11) and dropping the parameter dependency yields:

$$\mathcal{R}^n = \hat{U}^{n+1} - (A\hat{U}^n + B\hat{w}^n). \tag{16}$$

By using the ansatz $\hat{U}^n = \Phi a^n + U_B^n$ and multiplying the above equation with its transpose, we obtain

$$\begin{aligned}
\|\mathcal{R}^n\|^2 = \mathcal{R}^{n\,T}\mathcal{R}^n = {} & a^{n+1\,T}\Phi^T\Phi a^{n+1} + a^{n+1\,T}\boxed{\Phi^T U_B^{n+1}} + \boxed{U_B^{n+1\,T}\Phi}\, a^{n+1\,T} + \boxed{U_B^{n+1\,T}U_B^{n+1}} \\
& - \left(a^{n+1\,T}\Phi^T A\Phi a^n + a^{n+1\,T}\boxed{\Phi^T A U_B^n} + a^{n+1\,T}\Phi^T B\hat{w}^n\right) - \left(\boxed{U_B^{n+1\,T}A\Phi}\, a^n + \boxed{U_B^{n+1\,T}AU_B^n} + \boxed{U_B^{n+1\,T}B}\,\hat{w}^n\right) \\
& - a^{n\,T}\Phi^T A^T\Phi a^{n+1} - a^{n\,T}\boxed{\Phi^T A^T U_B^{n+1}} + \left(a^{n\,T}\Phi^T A^T A\Phi a^n + a^{n\,T}\boxed{\Phi^T A^T AU_B^n} + a^{n\,T}\Phi^T A^T B\hat{w}^n\right) \\
& - \boxed{U_B^{n\,T}A^T\Phi}\, a^{n+1} - \boxed{U_B^{n\,T}A^T U_B^{n+1}} + \left(\boxed{U_B^{n\,T}A^T A\Phi}\, a^n + \boxed{U_B^{n\,T}A^T AU_B^n} + \boxed{U_B^{n\,T}A^T B}\,\hat{w}^n\right) \\
& - (\hat{w}^n)^T B^T\Phi a^{n+1} - (\hat{w}^n)^T\boxed{B^T U_B^{n+1}} + \left((\hat{w}^n)^T B^T A\Phi a^n + (\hat{w}^n)^T\boxed{B^T AU_B^n} + (\hat{w}^n)^T B^T B\hat{w}^n\right).
\end{aligned} \tag{17}$$

Most operators in the above equation such as $\Phi^T A^T A\Phi \in \mathbb{R}^{N\times N}$ can be computed offline and used in the online phase for fast computation of the norm of the residual. The efficient computation of the terms in the dashed boxes will be explained in Sections 5.1.1 and 5.1.2 as we define U_B for each test case separately. For more information, refer to [11]. As the two-norm of the residual over the spatial domain, $\mathcal{R}_s$, can be computed cheaply in the reduced basis context, its ℓ_2-norm over time is also cheaply computable.

4.2.2. ℓ_∞ error bound

For a single PDE with time-dependent (not state-dependent) boundary conditions, we have the value of w^n for all time instants. Therefore, the inclusion of Σ_{BC} is not required and we have $e_w^n = 0$ for all n. In this case, the error dynamic reduces to

$$\Sigma_{lin}^e : \begin{cases} e^{n+1} = A(\mu)e^n - \mathcal{R}^n, \\ e_y^n = C_y e^n. \end{cases} \tag{18}$$

Based on the error dynamics (18), the ℓ_2 error bound (13) for these types of systems is as follows:

$$\left\|e_y\right\|_{\ell_2} \leq \gamma^{e_y\mathcal{R}}\|\mathcal{R}_s\|_{\ell_2}, \tag{19}$$

and the error bound can be compared with an error bound constructed by using the notion of input-to-state-stability (ISS) [35], which leads to a bound point-wise in (discretized) time. By solving (18) recursively over time for a given parameter μ, the error at each time instant is given by

$$e^{n+1} = A(\mu)^{\{n+1\}}e^0 - \sum_{i=0}^{n} A(\mu)^{\{n-i\}}\mathcal{R}^i, \tag{20}$$

where the notation $A^{\{n\}}$ is used to indicate the nth power of the matrix A, i.e.,

$$A^{\{n\}} = \underbrace{A \times A \times \cdots A}_{n \text{ times}}. \tag{21}$$

The use of (20) leads to a bound on the norm of the error as

$$\left\|e^{n+1}\right\| \leq \kappa_{ISS} := \left\|A(\mu)^{\{n+1\}}\right\|_2 \left\|e^0\right\| + \sum_{i=0}^{n} \left\|A(\mu)^{\{n-i-1\}}\right\|_2 \mathcal{R}_s^i, \tag{22}$$

where $\|A\|_2$ is the two-norm of matrix A. The ISS-like bound (22), although expensive to compute due to calculation of $A(\mu)^{\{i\}}$ and the corresponding norms, is the sharpest possible error bound with the available information. The

norm of the matrix should be calculated at each time step and this renders the usage of this bound computationally unattractive. However, this bound is a good reference measure for the performance of other error bounds. Instead, the error bound in (13) requires the norm calculation only once, i.e., much less computation than the bound (22). However, there are also two drawbacks of the error bound (13). First, calculation of $\mathcal{H}_\infty$-norms, even though only needed once, is expensive, and second, this error bound may be conservative. For resolving these two issues, an error estimate based on the error bound (13) is developed, as presented in the next section.

4.3. Error estimate for the reduced-order model

To resolve the problem of the expensive calculation of the $\mathcal{H}_\infty$-norm in the online phase, as needed in the ℓ_2 error bound (13), all required $\mathcal{H}_\infty-$norms, $\gamma^{e_y e_w}$ and $\gamma^{e_y \mathcal{R}}$, are computed and stored offline for all parameters in the discrete parameter domain $\mathcal{D}_h \subset \mathcal{D}$. These norms correspond to the ℓ_2-gain from different inputs to the output of Σ_{lin} that are present in (13) for all $\mu \in \mathcal{D}_h$. Here, as only the norms corresponding to the members of $\mathcal{D}_h$ are saved, memory demands are not increased. Then, in the online phase, for any new parameter that does not lie in $\mathcal{D}_h$, the closest element in $\mathcal{D}_h$ (in the Euclidean sense) is found and the norm of that parameter in $\mathcal{D}_h$ is used instead of the norm of the original parameter used in the online phase. If the distance from many parameter sets are the same, the one with the higher $\mathcal{H}_\infty$-norm is chosen.

Remark 9. Since the $\mathcal{H}_\infty$-norms are not computed exactly, but rather computed approximately based on the discrete parameter domain, the expression (13) is not mathematically a bound and would be an approximation of the error bound. Henceforth, wherever we mention an error bound, we mean an approximation of the error bound.

For resolving the problem of conservativeness, a similar approach to the one in [24] but in a different norm realization is used. Specifically, the bound (13) is adapted to the estimate

$$\left\|e_y\right\|_{\ell_2} \leq \kappa^{e_y} := \bar{\rho}\gamma^{e_y e_w}\|e_w\|_{\ell_2} + \bar{\rho}\gamma^{e_y \mathcal{R}}\|\mathcal{R}_s\|_{\ell_2}, \tag{23}$$

where $\bar{\rho}$ is the so-called transition factor that provides a measure of the average conservativeness of the bound (13). In particular, $\bar{\rho}$ can be computed after defining

$$\bar{\rho}_i^f = \max_n \frac{\sqrt{\sum_{j=1}^n \left\|e_y^j\left(\mu_i^*\right)\right\|^2}}{\gamma^{e_y e_w}\sqrt{\sum_{j=1}^n \left\|e_w^i\left(\mu_i^*\right)\right\|^2} + \gamma^{e_y \mathcal{R}}\sqrt{\sum_{j=1}^n \left(\mathcal{R}_s^j\left(\mu_i^*\right)\right)^2}}, \tag{24}$$

where e_y is the actual error computed in the offline phase for a parameter set μ_i^* as in (12). The parameter μ_i^* is the parameter selected during the greedy algorithm in the ith stage of the greedy algorithm in offline phase, $i = 1, \ldots, N$. The denominator of (24) is motivated by (13).

There are two alternatives for calculating $\bar{\rho}$. The first one is that we choose the last entry of $\bar{\rho}^f$ as the transition factor,

$$\bar{\rho} = \bar{\rho}_N^f. \tag{25}$$

The second one is

$$\bar{\rho} = \max(\bar{\rho}_i^f), \tag{26}$$

By embedding these transition factors into error estimates, the offline and online algorithms in the reduced basis method are illustrated in Algorithms 2 and 3. Other definitions of $\bar{\rho}^f$ can be straightforwardly tested, such as computing (24) only at the last time instant, which is a good option when the total error after the simulation horizon is interesting for the user. In this study, the options in (25) and (26) are tested. Obviously, using (26) leads to more conservative error estimates than (25).

In the next section, the effectiveness of the proposed error bound and estimate is tested in numerical case studies.

Remark 10. To enable the computation of the reduced solution and the corresponding error estimate, the following items should be stored in the offline phase and used in the online phase:

Algorithm 2: OFFLINE: Basis vector generation using POD-greedy

Input: $\mathcal{D}_h$, μ_0, ϵ
Output: Φ, $\bar{\rho}$, $\gamma^{e_y e_w}$, $\gamma^{e_y \mathcal{R}}$ $\forall \mu \in \mathcal{D}_h$

, Compute the required $\mathcal{H}_\infty$-norms that appear in (13) for all $\mu \in \mathcal{D}_h$,
Initialization: $i = 1$, $\mu_1^* = \mu_0$, $\Phi = [\ ]$, $\bar{\rho} = [\ ]$, $\kappa^{e_y}(\mu^*) = 2\epsilon$
while $\kappa^{e_y}(\mu^*) > \epsilon$ **do**
  Compute the truth solution by the parameter set μ_i^* as in (5), $S_{FV}\left(\mu_i^*\right) = \{U^n\left(\mu_i^*\right)\}_{n=0}^{N_t}$,
  $\bar{U}\left(\mu_i^*\right) = S_{FV}\left(\mu_i^*\right) - \Phi\Phi^T S_{FV}\left(\mu_i^*\right)$,
  $\varphi_i = \mathrm{POD}(\bar{U}\left(\mu_i^*\right), 1)$ as in Algorithm 1,
  Enrich reduced basis space, $\Phi = [\Phi, \varphi_i]$, where φ_i is the first POD mode of $\bar{U}\left(\mu_i^*\right)$,
  For all members (or the last entry) of μ^*, solve (9) to compute $S_{RB}\left(\mu_i^*\right) = \{\hat{U}^n\left(\mu_i^*\right)\}_{n=1}^{N_t}$,
  Set $e_y\left(\mu_i^*\right) = C_y\left(S_{FV}\left(\mu_i^*\right) - S_{RB}\left(\mu_i^*\right)\right)$ and compute the denominator of (24) at each time instant,
  Compute $\bar{\rho}^f j$ as in (24) for $j = 1, \cdots, i$,
  Run the greedy algorithm for all $\mu \in \mathcal{D}_h$ and find $\mu_{i+1}^* = \arg\max_{\mu \in \mathcal{D}_h} \kappa^{e_y}(\mu)$, calculated by Eq. (23) (the error indicator here, κ^{e_y}, can be replaced by the actual error or any other error indicator),
  $i = i + 1$,
Compute the projected matrices of the reduced-order model.

Algorithm 3: ONLINE: Reduced basis solution by Galerkin projection

Input: μ^o for online simulation, the required $\mathcal{H}_\infty$-norms for all $\mu \in \mathcal{D}_h$, the reduced basis space Φ, $\bar{\rho}$, projected matrices of the reduced-order model.
Output: $\hat{U}(\mu^o)$, κ^{e_y}

Find $\bar{\mu}^o = \arg\min_{\mu \in \mathcal{D}_h} \|\mu - \mu^o\|$,
Combine the projected matrices considering affine property of the system (5) to obtain the low-order dynamics in (9),
for $n = 1 : N_t$, **do**
  Compute the RB generalized coordinates, $a^n(\mu^o)$ based on the reduced-order model (9),
  Compute the RB solution $\hat{U}^n(\mu^o)$ via (7),
  Compute the norm of the residual $\mathcal{R}_s^n$ as in (17),
  Compute the error estimator via Eq. (23) by using $\gamma^{yu}(\bar{\mu}^o)$ and $\mathcal{R}_s^n$,
  $n = n + 1$,

- the parameter-independent operators obtained after the Galerkin projection, which are of the dimension of the reduced space,
- the ℓ_2-gains from different inputs to different outputs for all $\mu \in \mathcal{D}_h$.

The gains in the second item are scalar constants and require only little memory to be stored.

5. Numerical case studies

In this section, first the PDE models are introduced and the associated adaptations are discussed. Then, the reduced basis solution with the classical and the proposed ansatz (Eqs. (6) and (7)) are compared. Finally, the error bounds and estimates are applied on the discretized systems and the results are compared.

5.1. Case studies

As mentioned earlier, two well-known examples of models of the form (1) are an advection equation with a source term and the linearized isothermal Euler equations, which are discussed below.

5.1.1. Advection equation with a source term

We consider the following advection equation:

$$\begin{cases} \dfrac{\partial u}{\partial t} + c\dfrac{\partial u}{\partial x} = u, \\ u(0, x; \mu) = \mu_1, \\ u(t, 0; \mu) = \mu_2(t), \end{cases} \qquad t \in [0, T],\ x \in [0, L], \tag{27}$$

where $u = u(t, x; \mu) \in \mathbb{R}$ is the scalar conservative variable (e.g. heat or pressure), c is the transport velocity of the conservative variable ($f(u) = cu$, $J = c$, as in (1)) and $\mu_2(0) = \mu_1$. To be able to numerically solve this equation, an initial condition $\mu_1 \in \mathbb{R}$ (initial condition is constant over the spatial domain) and a time-varying boundary condition $\mu_2(t) \in \mathbb{R}$ are defined. In this test case, $x_\partial = 0$ as in (1) and we consider the varying parameters $\mu := (\mu_1, \mu_2, c, L) \in \mathcal{D} \subset \mathbb{R}^4$. The other parameters in the system are fixed.

Within this test case, the boundary condition is just a function of time, and no dynamics occur at the boundary. Therefore, the full-order model (5) simplifies to:

$$\Sigma_{lin} : \begin{cases} U^{n+1} = A(\mu)U^n + B(\mu)w^n, \\ y^n = C_y U^n. \end{cases} \tag{28}$$

To take into account the time-varying boundary conditions, the vector U_B in (7) should be defined. For this test case, there is only one boundary condition in the left side of the spatial domain at $x_\partial = 0$, thus:

$$U_B(U_{BC}^n) = U_B(U^n|_{x_\partial=0}) = \mu_2(t)\mathbf{1}, \tag{29}$$

where $\mathbf{1}$ is a vector of ones of dimension N_δ. By doing so, the vector U_B encodes the boundary condition and therefore $\hat{U}$ as in (7) correctly simulates the effects of the boundary. Hence, no dynamics occur at the boundary conditions as the boundary values are specified explicitly over time. Therefore, Σ_{BC} is omitted from (9) and this system reduces to

$$\hat{\Sigma}_{lin} : \begin{cases} a^{n+1} = \hat{A}_a a^n + \hat{B}\hat{w}^n + \hat{A}^*_{BC}\mu_2^n - \hat{B}^*_{BC}\mu_2^{n+1}, \\ \hat{U}^n = \mathbf{1}\mu_2^n + \Phi a^n, \\ \hat{y}^n = C_y \hat{U}^n, \end{cases} \tag{30}$$

where $\hat{A}^*_{BC} = \Phi^T A\mathbf{1}$ and $\hat{B}^*_{BC} = \Phi^T\mathbf{1}$ are both in $\mathbb{R}^N$. Now, the dynamics in (30) is completely independent of N_δ and offline-online decomposition is achieved thanks to the linearity and affine property of the full-order model (28) (Assumption 2). The error dynamics then is of the form (18) and the ℓ_2 error bound reads as (19). Therefore, in Algorithm 2 for this test case, we only need to calculate $\gamma^{e_y\mathcal{R}}$ for all parameters within $\mathcal{D}_h$ (recall that the error dynamics (18) is parameter dependent). In addition, by the definition of $U_B^n = \mathbf{1}\mu_2^n$, the terms in the dashed box in the residual calculation (17) can be computed cheaply. For instance,

$$U_B^{n\,T} A^T A\,\Phi = \mu_2^n \mathbf{1}^T A^T A\,\Phi \in \mathbb{R}^N,$$

is computed without being dependent on the size of the full-order system N_δ. Therefore, the online residual calculation does not depend on the dimension of the full-order model.

This test case can assess the quality of the ℓ_2 error bound by comparison against the ISS-like bound (22). After promising performance of the approximate error bound (19) on this test case, the more general approximate error bound (13) is applied to the linearized isothermal Euler equations where the effect of the local nonlinearities plays a crucial role.

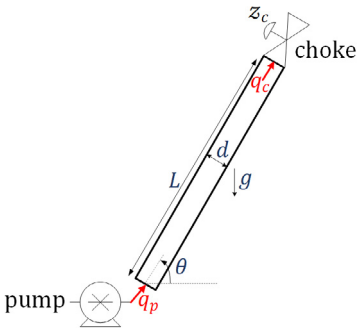

Fig. 4. Pipe configuration for the Euler equations.

5.1.2. Isothermal Euler equations

To mathematically describe a single-phase flow inside a pipe, the isothermal Euler equations encompassing mass and momentum conservation laws are used [18]. This system of PDEs is described as

$$\begin{cases} \dfrac{\partial \rho}{\partial t} + \dfrac{\partial}{\partial x}(\rho v) = 0, \\ \dfrac{\partial}{\partial t}(\rho v) + c_l^2 \dfrac{\partial \rho}{\partial x} = \rho g \sin\theta - 32 \dfrac{\zeta}{d^2} \dfrac{\rho v}{\rho_0}, \end{cases} \tag{31}$$

where $\rho = \rho(t, x; \mu)$ and $v = v(t, x; \mu)$ are the density and velocity of the fluid, respectively. Also, c_l is the speed of sound in the liquid medium, ρ_0 is the reference density around which the equation of state of the fluid is linearized, g is the gravitational acceleration, θ is the inclination of the pipe with respect to the horizontal direction, ζ is the viscosity of the fluid and d is the diameter of the pipe.

To compare it with (1), denote $u = [\rho \quad \rho v]^T$, $f = [\rho v \quad \rho c_l^2]^T$ and

$$S = \begin{bmatrix} 0 & \rho g \sin\theta - 32 \dfrac{\zeta}{d^2} \dfrac{\rho v}{\rho_0} \end{bmatrix}^T, \quad J = \begin{bmatrix} 0 & 1 \\ c_l^2 & 0 \end{bmatrix}. \tag{32}$$

Henceforth, wherever the Euler equations are mentioned, we refer to system (31) unless otherwise noted.

Boundary conditions are selected to enable the simulation towards industrial applications such as MPD [26] and pipeline networks [18]. In this regard, for the left boundary conditions (at $x_\partial = 0$), a liquid pump is assigned as

$$q_p = A_p v_p, \tag{33}$$

where q_p is the volumetric flow rate of the pump, A_p is the discharge area of the pump, and v_p is the liquid velocity at the pump. For the right boundary condition (at $x_\partial = L$), a choke manifold is located which is governed by the equation

$$q_c = K_c z_c \sqrt{\frac{2}{\rho_c}(p_c - p_{atm})}, \tag{34}$$

where q_c, K_c, z_c, ρ_c, p_c and p_{atm} are the volumetric flow rate through the choke, the choke constant, the choke opening, the density at the choke inlet at the pipe side, the pressure at the choke inlet at the pipe side and the atmospheric pressure, respectively. This test case can be visualized as in Fig. 4 with $x_\partial = \{0, L\}$.

However, to numerically solve the coupled system (31), the values for both ρ and ρv at each boundary are required while there is only one static relation at each boundary. For instance, at the left boundary condition, by assigning q_p, we know v at the boundary and ρ is unknown. In order to resolve this issue and render the dynamics at the boundary consistent with the internal dynamics, the characteristic-based boundary equations as in [19,21] are unified with the static relations (33) and (34). These characteristic-based equations define G in (5).

In this test case, $\mu := (\rho_0, \theta, c_l, d, \nu, q, L) \in \mathcal{D} \subset \mathbb{R}^7$. Other remaining parameters, such as K_c, are fixed from one simulation to another. Finally, for numerical simulations, the initial condition of this system for any given parameter setting is the corresponding steady-state solution of the system. Then, by changing the control inputs u_c (q_p and z_c in this test case), the dynamics of the system are excited.

To incorporate boundary conditions for the linear isothermal Euler equation (31), we take

$$U_B^n(\hat{U}_{BC}^n) = \left(1 - \frac{X}{L}\right)\hat{U}_{BC}^n|_{x_\partial=0} + \frac{X}{L}\hat{U}_{BC}^n|_{x_\partial=L}, \tag{35}$$

where $\hat{U}_{BC}^n|_{x_\partial=0}$ and $\hat{U}_{BC}^n|_{x_\partial=L}$, respectively, are the left and the right boundary values of the conservative variables (corresponding to U_{BC}^n in (5)) computed by the solution of the reduced-order model and $X \in \mathbb{R}^{N_\delta}$ is a vector containing the location of the grid cells. The logic behind choosing a linear function for incorporating the boundary conditions inside the domain originates from the steady-state solution of (31). In steady state, momentum is constant along the spatial domain, and due to the high value of the speed of sound, density varies almost linearly with respect to the spatial domain (this can be seen numerically in Section 5.2). In this way, the reduced basis approximation can capture the steady-state solution with a better accuracy compared to the situation where the classical ansatz (6) is used. Similar to the advection equation, by the definition of U_B^n, the residual (17) can be calculated cheaply. To include the effect of U_B^n in the reduced-order model and the detailed analysis for the error bound and estimate, refer to Appendix.

5.1.3. Parameters for the numerical experiments

The numerical results are divided into two parts. In the first part, the effect of changing the ansatz is tested by comparing the use of (6) and (7) for both the advection and the Euler equations. It is shown that the proposed ansatz (7) with the interconnection shown in Fig. 2 is capable of incorporating the time-varying and nonlinear boundary conditions in the advection and Euler equations while the reduced solution with the ansatz (6) generates discontinuity at the boundary locations. The simulation speedups in the online phase compared to the full-order solution are also reported without considering computations of the residuals, the error bound and estimate; it means that in Algorithm 3, lines 1, 6 and 7 are not included in the cpu time. In the second part, the performance of the error bound and estimate is analyzed. The error bound is compared with the error bound introduced in [11] (denoted by HO in the figures). For the advection equation, it is numerically confirmed, by comparing the results with the ISS-like bound (22), that the proposed error bound is one of the sharpest possible error bound on systems reaching steady state with constant inputs. This is due to the fact that the $\mathcal{H}_\infty$-norms of Σ_{lin} occur at zero frequency. Moreover, the error estimate of [24] (denoted by Dual in the figures) is compared with the proposed error estimate. In the following, the simulation parameters for both test cases are introduced.

Advection equation with the source term

The simulation parameters for the advection equation (27) along with the properties of the parameter domain are listed in Table 1, where the minimum and maximum value for each parameter are specified. The discrete parameter domain $\mathcal{D}_h$ is composed of 81 equidistant members in the parameter domain $\mathcal{D}$ (each parameter contains three equally distributed values in its region. In other words, $\mathcal{D}_h$ is a grid of 81 equidistant points in $\mathcal{D}$.). In the last row of Table 1, the set of parameters selected for the online simulation μ^o is reported, which does not lie in the discrete parameter domain. The number of spatial grid cells are $N_\delta = 500$, the time horizon T is 10 s and time steps are changed in a way such that at each simulation $c\Delta t/\Delta x = 0.9$ (this is the so-called CFL number determining the stability of the full-order model [21]). For the boundary conditions, we take

$$\mu_2(t) = \begin{cases} \mu_1, & 0 \le t \le 2, \\ \mu_{BC}, & 2 < t \le 4, \\ 0, & 4 < t \le 6, \\ \dfrac{\mu_{BC}}{2}, & 6 < t \le 10. \end{cases} \tag{36}$$

Linearized isothermal Euler equations

For the Euler equations (31), the test case is shown in Fig. 4. The first two rows of Table 2 indicate the range of variation of the varying parameters and the third row shows the values used in the online phase, μ^o. The discrete parameter domain $\mathcal{D}_h$ contains 128 members equally distributed in the parameter domain $\mathcal{D}$ (each parameter contains two equally distributed values in its region). Fixed parameters for the Euler equations are listed in Table 3.

The full-order model contains 500 spatial grid cells, $N_\delta = 500$, with the time horizon $T = 25$ s and time steps chosen such that at each simulation the CFL number is 0.99, i.e., $c_l\Delta t/\Delta x = 0.99$. The the boundary conditions

Table 1
Variation range of the varying parameters in the advection equation.

Parameter	c [m/s]	L [m]	μ_{BC}	μ_1
Minimum	50	100	1	0
Maximum	100	200	5	2
Online μ^o	85	125	4	1.5

Table 2
Variation range of the varying parameters in the Euler equations.

Parameter	ρ_0 [kg/m^3]	θ	c_l [m/s]	d [m]	ζ [Pa s]	q [m^3/s]	L [m]
Minimum	800	60	1000	0.2	0.004	0.01	800
Maximum	1200	90	1500	0.5	0.04	0.05	1200
Online μ^o	1000	72	1250	0.35	0.02	0.03	1000

Table 3
Fixed parameters for the Euler equations.

Parameter	g [m/s^2]	p_{atm} [Pa]	K_c
Value	9.81	10^5	0.00285

are taken as

$$q_p(t) = \begin{cases} q & 0 \le t \le 3 \\ 0.2q & 3 < t \le 25 \end{cases}, \quad z_c(t) = \begin{cases} 1 & 0 \le t \le 1 \\ 0.1 & 1 < t \le 25 \end{cases}. \tag{37}$$

Finally, it should be mentioned that in the simulation related to the Euler equations, the same number of basis vectors for density ρ and momentum ρv is used.

5.2. Illustration of the proposed ansatz formulation

In this section, the effect of the ansatz modification on the correct implementation of the boundary conditions is investigated.

Advection equation with the source term

The result of the reduced basis method with and without changing the ansatz at different time instants are compared in Figs. 5 and 6. These figures compare the results of (27) solved by the finite-volume method (denoted by FV), the reduced basis method with the classical ansatz (6) (denoted by RB) and the reduced basis method of (27) with the proposed ansatz (7) (denoted by RB-ansatz) for a varying number of basis vectors, namely 10 and 30 basis vectors, respectively. The space of basis vectors in this section is not enriched with the initial conditions. As is apparent from the figures, the boundary condition (at $x_\partial = 0$) is not implemented correctly even by using 10 basis vectors with ansatz (6). This can be visually confirmed in the figures by the mismatch at the boundary at $x_\partial = 0$. This mismatch is due to the fact that the number of degrees of freedom to simultaneously satisfy the boundary conditions and the internal domain is not sufficient. In other words, in this case, the reduced basis compromises between the accuracy of the solution in the internal domain and the correct implementation of the boundaries. However, for the reduced basis with the ansatz (7), the boundary conditions are always implemented correctly, regardless of the number of basis vectors.

Moreover, the initial condition is captured more accurately with the ansatz (7). This means that the initial energy of the system is also captured better. It should be noted that the drawback in capturing the initial condition in the classical ansatz (6) can be resolved by enriching the space of the basis vectors with the initial condition.

To compare the speedups in the online phase and the full-order solution, see Table 4. As mentioned previously, computations of the error bound (estimate) are not included in the speedup calculation. In the online phase, the state for the entire domain is calculated to compare the solutions in the entire domain, which scales with the degrees of freedom of the original model and decreases the speedups. The other reason for the moderate speedups is the

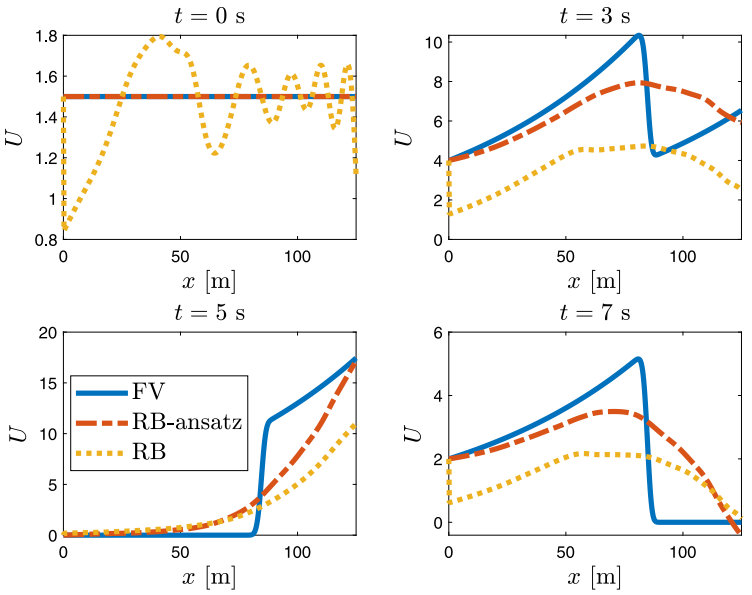

Fig. 5. Incorporation of the boundaries by using 10 basis vectors for the advection equation; comparison of the finite-volume (FV) solution and the reduced basis method with (7) (RB-ansatz) and (6) (RB).

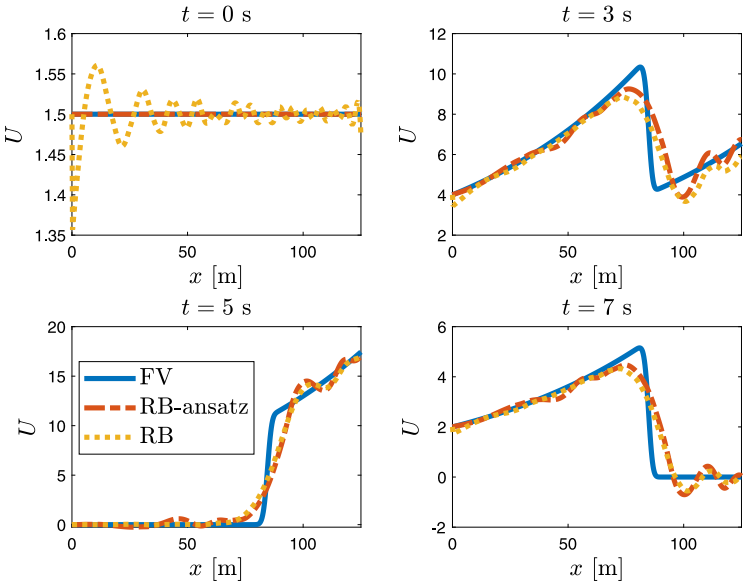

Fig. 6. Incorporation of the boundaries by using 30 basis vectors for the advection equation; comparison of the finite-volume (FV) solution and the reduced basis method with (7) (RB-ansatz) and (6) (RB).

Table 4
Speedup factors for the reduced basis method for the advection equation.

N	1	10	20	30	40	50
Speedup	16.8	8.2	7.9	6.2	5.5	5.3

1D type of problems considered in this paper, where the expected speedups are modest. The hyperbolicity of the advection equation can also be noted as another reason for moderate speedups.

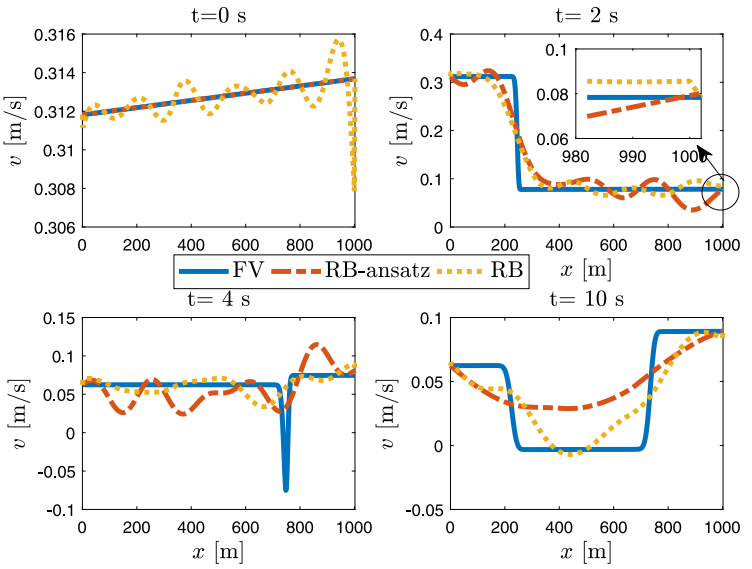

Fig. 7. Incorporation of the boundaries by using 10 basis vectors for the Euler equations; comparison of the finite-volume (FV) solution and the reduced basis method with (7) (RB-ansatz) and (6) (RB).

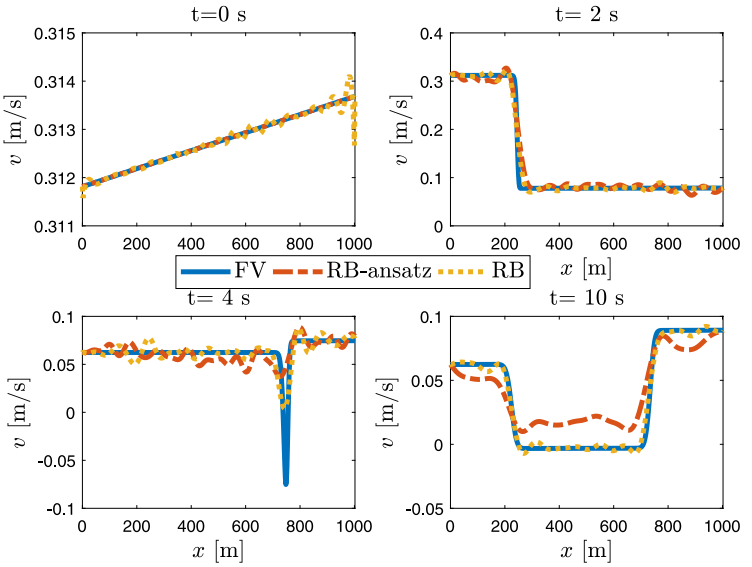

Fig. 8. Incorporation of the boundaries by using 30 basis vectors for the Euler equations; comparison of the finite-volume (FV) solution and the reduced basis method with (7) (RB-ansatz) and (6) (RB).

Linearized isothermal Euler equations

Analogously, the result of the reduced basis method with the ansatz (6) and (7) at different time instants are compared in Figs. 7 and 8 for the linearized isothermal Euler equations. The velocity of the liquid v is depicted in these figures when using 10 and 30 basis vectors.

Similar to the advection equation case study, the boundary condition treatment is not perfect with the classical ansatz (6) as mismatches at the two boundaries (at $x_\partial = 0$ and $x_\partial = L$) are present. Similar to the result for the advection equation, initial conditions, initial mass and initial energy of the system are captured better with the ansatz (7) compared to the ansatz (6).

To compare the speedups, refer to Table 5.

Table 5
Speedup factors for the reduced basis method for the Euler equations.

N	1	10	20	30	40	50
Speedup	18.8	11.4	11.2	8.3	7.7	7.5

The relatively modest speedup factors can be attributed to the fact that, at each time step, a nonlinear function should be solved to compute the boundary conditions both for the full-order and the reduced-order model, which dominates the cpu time of both simulations. Similar to the advection equation, the full state is reconstructed in the online phase to compare the solutions in the entire domain, whose computation scales with the degrees of freedom of the original model and reduces the speedup. Hyperlic nature of the Euler equations is another reason of moderate speedup values.

All in all, the ansatz (7) captures the effect of the boundary conditions better than the classical one (6), leading to a correct incorporation of the control inputs at the boundaries.

5.3. Performance of the error bound and estimate

In this section, the proposed error bound and estimate are compared with the actual error and also existing error bounds [11] and estimates [24] in the ℓ_2-norm. First, the results are presented for the advection equation with source term and, second, for the linearized isothermal Euler equations. The variables that are shown in the figures of this section are computed as

$$\mathcal{E}^n_{act} = \frac{1}{\sqrt{n}} \sqrt{\sum_{i=1}^{n} \left\| e^i_y \right\|^2}, \tag{38a}$$

$$\mathcal{E}^n_{\mathcal{H}} = \frac{1}{\sqrt{n}} \left(\gamma^{e_y e_w} \sqrt{\sum_{i=1}^{n} \left\| e^i_w \right\|^2} + \gamma^{e_y \mathcal{R}} \sqrt{\sum_{i=1}^{i=n} (\mathcal{R}^i_s)^2} \right), \tag{38b}$$

$$\mathcal{E}^n_{ISS} = \frac{1}{\sqrt{n}} \sqrt{\sum_{i=1}^{n} (\kappa^i_{ISS})^2}, \tag{38c}$$

where $\mathcal{E}_{act}$, $\mathcal{E}_{\mathcal{H}}$ and $\mathcal{E}_{ISS}$ are the indicators used for the actual error, the proposed error bound and ISS-like error bound. For the error bound and estimate introduced in [11,24] (HO and Dual in the following figures, respectively), a similar approach to (38c) is followed to compute the variables shown in the figures of this section; instead of κ_{ISS}, the bound in those papers are used.

Advection equation with source term

In order to make a fair comparison between the proposed error bound in (19), which is a bound on the output error e_y, and the ISS-like bound in (22), which is a bound on the state error e, all states are considered as the output, i.e., $C_y = I_{N_\delta \times N_\delta}$ where I is the identity matrix. To have a zero error at the initial time and justify using the ℓ_2-gain notion (10), the space of basis vectors is enriched by the modified initial conditions, which is computed similarly to (8) with $t = 0$. The error bound comparisons are depicted in Fig. 9. Obviously, the ISS-like bound is always the sharpest bound. On the one hand, the computation of the ISS-like bound is expensive due to the computation of the norms in (22) at each time step. On the other hand, the bound (19) is cheaper to compute as it calculates the $\mathcal{H}_\infty$-norms only once and also it is closer to the ISS-like bound compared to the HO bound after the initial transient. The proposed bound, unlike the HO bound, remains bounded for increasing time. Therefore, the proposed bound in this paper is one of the sharpest possible error bounds for steady-state certification. The jumps visible in Figs. 9 and 10 in the error bounds and estimates are due to the fact that at these time instants the boundary values change drastically and therefore the residual changes significantly.

The comparison of the error *estimates* is depicted in Fig. 10, revealing that the estimate (23) is very close to the actual error by selecting either option regarding $\bar{\rho}$. In this figure, the option of $\bar{\rho}$ as in (25) is shown by '$\mathcal{H}_\infty$-last'

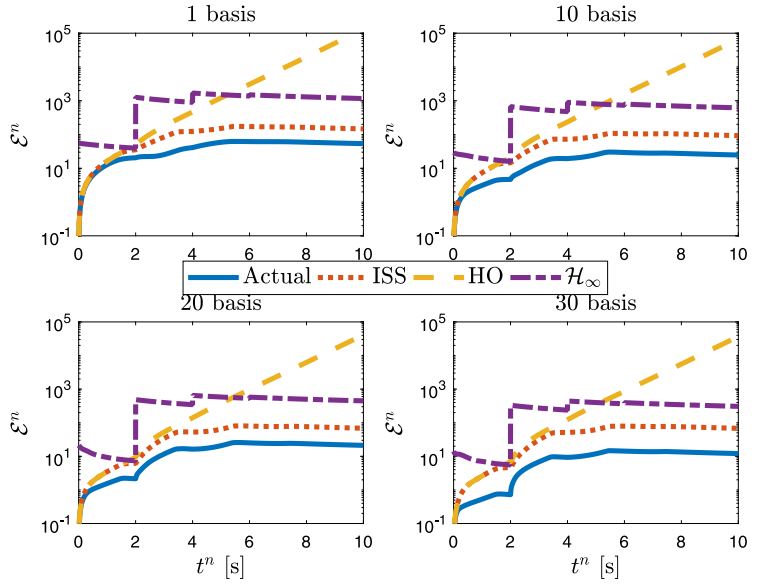

Fig. 9. Comparison of the error bounds, $\mathcal{E}^n_{act}$ ('Actual'), $\mathcal{E}_{\mathcal{H}}$ ('$\mathcal{H}_\infty$'), $\mathcal{E}_{ISS}$ ('ISS') and 'HO', for the online parameter set; advection equation.

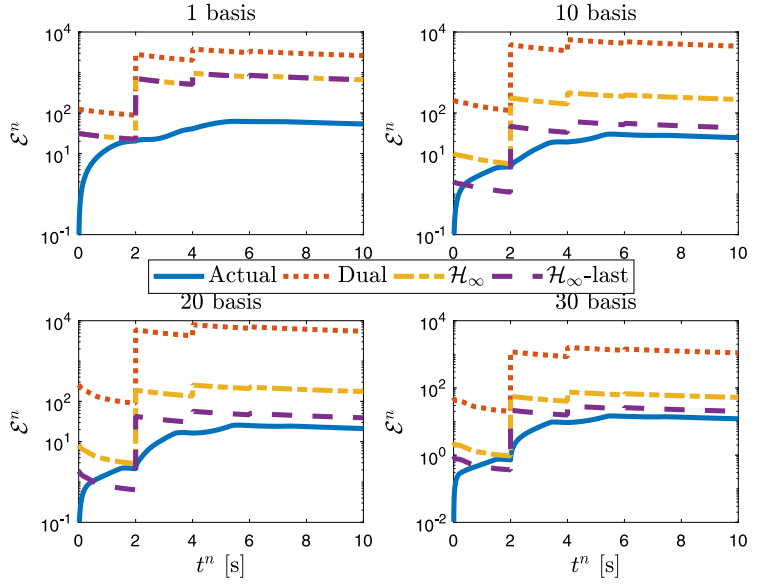

Fig. 10. Comparison of the error estimates, $\mathcal{E}^n_{act}$ ('Actual'), 'Dual', $\bar{\rho} \times \mathcal{E}_{\mathcal{H}}$ ($\bar{\rho}$ as in (26) denoted by '$\mathcal{H}_\infty$' and as in (25) denoted by '$\mathcal{H}_\infty$-last') for the online parameter set; advection equation.

and $\bar{\rho}$ as in (26) is shown by '$\mathcal{H}_\infty$'. Clearly, both error estimates proposed here outperform the one proposed in [24] in the ℓ_2-norm realization. Moreover, '$\mathcal{H}_\infty$-last' provides a sharper estimate than '$\mathcal{H}_\infty$' (as expected). However, it was observed that sometimes the estimate '$\mathcal{H}_\infty$-last' underestimates the actual error. Nevertheless, it is always close to the actual error within a narrow band. For $\bar{\rho}$ as in (26), the error estimate never underestimated the actual error. It should be noted that the dual error introduced in [24] may perform better in other norms.

The dependency of the actual error and the error bounds and estimates as a function of the number of basis vectors is shown in Fig. 11, where 'ℓ_2' is the bound computed by (38b) at $n = N_t$. This figure compares the maximum estimated error among the members in the discrete parameter domain during the greedy algorithm. The

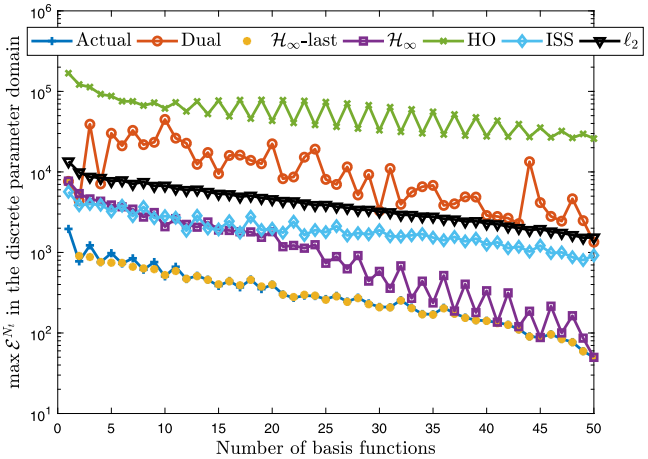

Fig. 11. Comparison of the maximum of error bounds and estimates computed for all members of the parameter domain during the greedy algorithm; advection equation.

Table 6
Percentage of the CPU time allocated to the computation of the error estimate during the online phase of the reduced-order model, the advection equation.

N	1	10	20	30	40	50
Ratio	44%	75%	74%	72%	71%	70%

Table 7
Percentage of the CPU time allocated to the computation of the error estimate during the online phase of the reduced-order model, the Euler equations.

N	1	10	20	30	40	50
Ratio	82%	71%	70%	70%	69%	69%

proposed error estimate with $\bar{\rho}$ as in (25) always approximates the maximum error in the parameter domain with high accuracy while the performance of the error estimate with $\bar{\rho}$ as in (26) and the error estimate in [24] are more conservative. Table 6 reports on the percentage of the online calculation devoted to the calculation of the error estimate when varying the number of basis vectors. This table numerically confirms that the computational cost required for the online error bound computation is slightly larger than the one for the computation of the reduced-order solution due to the residual calculation. However, as shown before, the computations related to the residual calculation are independent of the dimension of the full-order model N_δ.

Linearized isothermal Euler equations

For this test case, as we can only approximate the Lipschitz constant of the nonlinear functions, no error bound is introduced. The detailed analysis of the error estimation computation can be found in Appendix. Here, the pressure at the last grid cell, the grid cell before the choke manifold, is selected as the output. The comparison of the error estimates with the actual error is presented in Fig. 12. Moreover, Fig. 13 compares the maximum estimated error among the members of the discrete parameter domain during the greedy algorithm. Similar to the advection equation, the error indicator in Algorithm 2 is controlled by different approaches proposed and introduced in this paper. The proposed error estimate in this paper works effectively and always predicts the error with a good accuracy. Compared to the error estimate in [24], the estimation of the actual error has improved significantly in the ℓ_2-norm. Although this estimate is very effective, its applicability region is limited, which has been discussed at the end of Appendix of this paper. The percentage of the CPU time allocated to calculate the error estimate in the online phase by varying N is listed in Table 7. Similar to the advection equation, calculation of the residual takes a significant percentage of the CPU time. The high ratio at the very low number of basis vectors relates to the

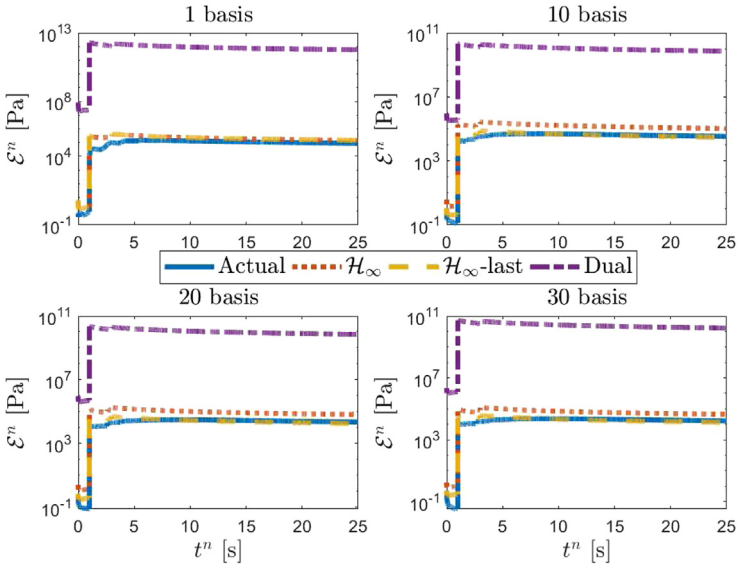

Fig. 12. Comparison of the error estimates, $\mathcal{E}^n_{act}$ ('Actual'), 'Dual', $\bar{\rho} \times \mathcal{E}_{\mathcal{H}}$ ($\bar{\rho}$ as in (26) denoted by '$\mathcal{H}_\infty$' and as in (25) denoted by '$\mathcal{H}_\infty$-last') for the online parameter set; Euler equations.

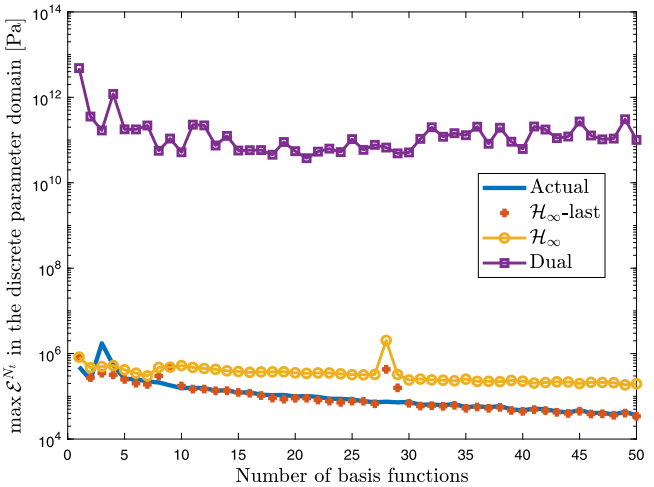

Fig. 13. Comparison of the maximum of error estimates computed for all members of the parameter domain during the greedy algorithm; Euler equations.

very fast solution of the reduced-order model while the computation of the Lipschitz constants do not scale with the number of basis vectors.

5.4. Discussion

To conclude, as apparent from the results presented in this section, the ansatz (7) is capable of incorporating the boundary conditions effectively, both for the advection and the Euler equations. In addition, the proposed error bound (19) is very close to the ISS-like bound (22) in the advection equation and significantly computationally more attractive than the ISS-like bound (22). Finally, the computationally efficient error estimate (23) is sharp and able to predict the actual error with a reasonable accuracy.

6. Conclusion and future work

In this paper, the reduced basis method is applied to systems with time-varying and nonlinear boundary conditions. By changing the ansatz of the reduced basis, these types of boundary conditions are handled properly while mismatches and discontinuities are observed at the boundaries with the classical ansatz.

Moreover, a new residual-based error bound and estimate are proposed. Given the accessible information in the reduced basis context (the spatial two-norm of the residual at each time instant), this error bound is sharp in steady-state for a special class of systems; systems for which inputs are constant. To mitigate the conservativeness of the error bound, an accurate error estimate is introduced. The error estimate outperforms the existing error estimates in ℓ_2-norm realization.

For future work, the application of the approach on real applications such as a managed pressure drilling systems will be investigated. The interconnection approach of handling local nonlinearities at the boundaries will be extended to local nonlinearities inside the domain. The extension of the error bound and estimate for nonlinear systems is also being investigated.

Acknowledgment

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.

Appendix. An error estimate for the linearized isothermal Euler equations

In Section 5.1.1, the procedure for the computation of the error bound and error estimate for the advection equation (27) was elaborated. This equation consists of a linear system with time-varying boundary conditions. In this section, the procedure for computing the error estimate for the Euler equations (31) is illustrated. This test case is also an internally linear system but with time-varying, nonlinearly state-dependent boundary conditions. The pump boundary located at $x_\partial = 0$ and the choke boundary at $x_\partial = L$ lead to nonlinear time-varying boundary conditions. Both of the boundary conditions are coupled with the states of the system. As the Lipschitz constant for the local nonlinearity at $x_\partial = L$ is not exactly known and has to be estimated, the error bound is not a true upper bound and is somehow an estimate of the upper bound. Due to this fact, we just propose an error estimate for this test case. The entire procedure for the error estimation computation is detailed below.

Following the same idea for advection equation, we can derive the error dynamics for the Euler equations. Pursuing the same steps explained in Section 4.1, the following error dynamics is obtained:

$$\Sigma_{nl}: \begin{cases} \Sigma^e_{lin}: \begin{cases} e^{n+1}_\rho = A_{11}e^n_\rho + A_{12}e^n_m + B_1 e^n_{V_1} + F_1 e^n_{W_2} - \mathcal{R}^n_\rho, \\ e^{n+1}_m = A_{21}e^n_\rho + A_{22}e^n_m + B_2 e^n_{V_1} + F_2 e^n_{W_2} - \mathcal{R}^n_m, \\ e^n_y = C_y[e^T_\rho \ e^T_m]^T, \\ e_z = [e^n_{\rho_1} \ e^n_{\rho_{N_\delta}} \ e^n_{m_1} \ e^n_{m_{N_\delta}}]^T = C_z[e^T_\rho \ e^T_m]^T, \end{cases} \\ \Sigma^e_{BC}: \begin{cases} x_\partial = 0: \begin{cases} e^{n+1}_{V_2} = \alpha_1 e^n_{V_2} + \alpha_2 e^n_{\rho_1} + \alpha_3 e^n_{m_1}, \\ e^n_{V_1} = Q^n e^n_{V_2}, \end{cases} \\ x_\partial = L: \begin{cases} e^{n+1}_{W_1} = \beta_1 e^n_{W_1} + \beta_2 e^n_{\rho_{N_\delta}} + \beta_3 e^n_{m_{N_\delta}}, \\ e^n_{W_2} = h(W^{n+1}_1, z^n_c) - h(\hat{W}^{n+1}_1, z^n_c), \end{cases} \\ e^n_{U_{BC}} = C_{BC}[(e^n_V)^T \ (e^n_w)^T]^T. \end{cases} \end{cases} \tag{A.1}$$

Here, ρ is a vector of averaged densities within the grid cells, and $m = \rho v$ is momentum. Next, V_i and W_i, $i \in \{1, 2\}$, are the expressions emerging from the finite-volume discretization necessary for solving the boundary conditions. It should be noted that $W^n_1 = e^n_{W_1} + \hat{W}^n_1$. Moreover, $e_{(\cdot)} = (\cdot) - (\hat{\cdot})$ is the difference between the full-order and reduced-order solution. The variable Q^n represents the time variation related to the pump boundary condition (33) and the function $h(\cdot, \cdot)$ encompasses the effect of the nonlinear choke boundary condition (34).

In addition, α_i and β_i, $i \in \{1, 2, 3\}$ are some constants and A_{ij}, B_i and F_i, $i, j \in \{1, 2\}$ are matrices generated as the result of the finite-volume discretization. $\mathcal{R}_{(\cdot)}$ is the residual attributed to the equation solving for the variable $(\cdot)$. The nonlinearities and time variation associated with the boundary conditions does not fit into the LTI representation

of the system. Then, by the interconnection in Fig. 3, the effects of the boundaries are included in the system as auxiliary inputs. To make a connection between (A.1) and (12), set

$$\begin{aligned} e &= \begin{bmatrix} e_\rho \\ e_m \end{bmatrix}, A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, B = \begin{bmatrix} B_1 & F_1 \\ B_2 & F_2 \end{bmatrix}, w = \begin{bmatrix} V_1 \\ W_2 \end{bmatrix}, \\ V &= \begin{bmatrix} V_2 \\ W_1 \end{bmatrix}, G = \begin{bmatrix} \alpha_1 V_2^n + \alpha_2 \rho_1^n + \alpha_3 m_1^n \\ \beta_1 W_1^n + \beta_2 \rho_{N_\delta}^n + \beta_3 m_{N_\delta}^n \end{bmatrix}, \mathcal{G} = \begin{bmatrix} Q^n V_2^n \\ h\left(W_1^n, z_c^n\right) \end{bmatrix}. \end{aligned} \tag{A.2}$$

Finally, C_z and C_{BC} in (A.1) are selected to generate the required auxiliary outputs and C_y is defined by the user to yield the output of interest. Here, the pressure at the choke is chosen as the output. A good indicator for the pressure is density, therefore $C_y = \left[\mathbf{0}_{N_\delta - 1},\ 1,\ \mathbf{0}_{N_\delta}\right]$, where $\mathbf{0}_{N_\delta} \in \mathbb{R}^{1 \times N_\delta}$ is a vector of zeros.

Considering the following state of vectors and rewriting the whole error dynamics (A.1), we have

$$\begin{cases} \bar{e} = \begin{bmatrix} e_{V_2} & e_\rho^T & e_m^T & e_{W_1} \end{bmatrix}^T, \\ \bar{e}^{n+1} = \bar{A}\bar{e}^n + B_{\mathcal{R}}\mathcal{R}^n + \bar{B}_{V_1} e_{V_1}^n + \bar{B}_{W_2} e_{W_2}^n, \\ \bar{e}_y^n = \bar{C}_y \bar{e}^n, \end{cases} \tag{A.3}$$

with

$$\bar{A} = \begin{bmatrix} \alpha_1 & \alpha_2 E_1 & \alpha_3 E_1 & 0 \\ \mathbf{0} & A_{11} & A_{12} & \mathbf{0} \\ \mathbf{0} & A_{21} & A_{22} & \mathbf{0} \\ 0 & \beta_2 E_N & \beta_3 E_N & \beta_1 \end{bmatrix}, B_{\mathcal{R}} = \begin{bmatrix} \mathbf{0} \\ -I_{2N_\delta \times 2N_\delta} \\ \mathbf{0} \end{bmatrix}, \bar{B}_{V_1} = \begin{bmatrix} 0 \\ B_1 \\ B_2 \\ 0 \end{bmatrix}, \bar{B}_{W_2} = \begin{bmatrix} 0 \\ F_1 \\ F_2 \\ 0 \end{bmatrix}, \mathcal{R} = \begin{bmatrix} \mathcal{R}_\rho \\ \mathcal{R}_m \end{bmatrix}, \tag{A.4}$$

where $\mathbf{0}$ is a zero vector with appropriate dimension and $E_1 = [1\ 0\ \cdots\ 0]$, $E_N = [0\ \cdots\ 0\ 1] \in \mathbb{R}^{N_\delta}$. The nonlinear part then includes only the static function $\mathcal{G}$ as in (A.2).

Relying on the same idea of (13) and decomposing the inputs of the linear system into e_{V_1} and e_{W_2}, the error bound for (A.3) is obtained as

$$\left\|\bar{e}_y\right\|_{\ell_2} \leq \gamma^{\bar{e}_y \mathcal{R}} \|\mathcal{R}_s\|_{\ell_2} + \gamma^{\bar{e}_y V_1} \left\|e_{V_1}\right\|_{\ell_2} + \gamma^{\bar{e}_y W_2} \left\|e_{W_2}\right\|_{\ell_2}. \tag{A.5}$$

If we can compute the right-hand side of (A.5), we can use (23) to compute the error estimate. To this end, we need the following relations (note that Q^n is a scalar),

$$\begin{aligned} \left\|e_{V_1}^n\right\| &\leq \left|Q^n\right| \left\|e_{V_2}^n\right\| \quad &\rightarrow \left\|e_{V_1}\right\|_{\ell_2} \leq \|Q\|_{\ell_\infty} \left\|e_{V_2}\right\|_{\ell_2}, \\ \left\|e_{W_2}^n\right\| &\leq \eta_{W_2 W_1} \left\|e_{W_1}^n\right\| \quad &\rightarrow \left\|e_{W_2}\right\|_{\ell_2} \leq \eta_{W_2 W_1} \left\|e_{W_1}\right\|_{\ell_2}, \end{aligned} \tag{A.6}$$

where $\eta_{W_2 W_1}$ is the local Lipschitz constant of the nonlinear function $h(\cdot, \cdot)$ with respect to its first argument, which is defined similar to (15). As mentioned earlier, due to this approximation of the Lipschitz constant, we do not calculate any error bound for the Euler equations and only calculate an error estimate.

Remark 11. An empirical alternative (which is not tested in this paper) for computing the Lipschitz constant can be provided due to the fact that we have the truth solution in the offline phase. In the offline phase, for the set of parameters μ^*, $\eta_{W_2 W_1}$ can be found. One can use the Lipschitz constant either for the last selected set of parameters or the maximum Lipschitz constant for all the previously selected set of parameters.

For the sake of notation, henceforth, we set $\eta := \left\|\eta_{W_2 W_1}\right\|_{\ell_\infty}$ and $Q := \|Q\|_{\ell_\infty}$. Thus, we have:

$$\begin{aligned} \left\|e_{W_1}\right\|_{\ell_2} &\leq \gamma^{W_1 \mathcal{R}} \|\mathcal{R}_s\|_{\ell_2} + \gamma^{W_1 W_2} \left\|e_{W_2}\right\|_{\ell_2} + \gamma^{W_1 V_1} \left\|e_{V_1}\right\|_{\ell_2}, \\ \left\|e_{W_2}\right\|_{\ell_2} &\leq \eta\gamma^{W_1 \mathcal{R}} \|\mathcal{R}_s\|_{\ell_2} + \eta\gamma^{W_1 W_2} \left\|e_{W_2}\right\|_{\ell_2} + \eta\gamma^{W_1 V_1} \left\|e_{V_1}\right\|_{\ell_2}. \end{aligned} \tag{A.7}$$

Then,

$$\left\|e_{W_2}\right\|_{\ell_2} \leq \frac{\eta}{1 - \eta\gamma^{W_1 W_2}} \left(\gamma^{W_1 \mathcal{R}} \|\mathcal{R}_s\|_{\ell_2} + \gamma^{W_1 V_1} \left\|e_{V_1}\right\|_{\ell_2}\right). \tag{A.8}$$

Similarly, for e_{V_1}, we have

$$\left\|e_{V_1}\right\|_{\ell_2} \leq \frac{Q}{1-Q\gamma^{V_2V_1}}\left(\gamma^{V_2\mathcal{R}}\left\|\mathcal{R}_s\right\|_{\ell_2}+\gamma^{V_2W_2}\left\|e_{W_2}\right\|_{\ell_2}\right). \tag{A.9}$$

Finally, by inserting (A.8) into (A.9), we get

$$\left\|e_{V_1}\right\|_{\ell_2}\left(1-\sigma_1\sigma_2\gamma^{V_2W_2}\gamma^{W_1V_1}\right) \leq \sigma_1\left(\gamma^{V_2\mathcal{R}}+\sigma_2\gamma^{V_2W_2}\gamma^{W_1\mathcal{R}}\right)\left\|\mathcal{R}_s\right\|_{\ell_2}, \tag{A.10}$$

where

$$\begin{aligned}\sigma_1 &= \frac{Q}{1-Q\gamma^{V_2V_1}},\\ \sigma_2 &= \frac{\eta}{1-\eta\gamma^{W_1W_2}}.\end{aligned} \tag{A.11}$$

For using this bound, these norms in Algorithm 2 should be calculated: $\gamma^{\bar{e}_y\mathcal{R}}$, $\gamma^{\bar{e}_yV_1}$, $\gamma^{\bar{e}_yW_2}$, $\gamma^{W_1\mathcal{R}}$, $\gamma^{W_1W_2}$, $\gamma^{W_1V_1}$, $\gamma^{V_2\mathcal{R}}$, $\gamma^{V_2W_2}$. Due the special structure of the model and also parameter settings, in all simulations we observe that $\sigma_1 \approx 1$ and $\gamma^{V_2W_2}\gamma^{W_1V_1} \approx 1$. Then, the conservativeness and applicability of the error estimate are highly dependent on σ_2 and hence on η and, consequently, on the choke operating condition. By having the upper bound on $\left\|e_{V_1}\right\|_{\ell_2}$, the upper bounds on $\left\|e_{W_2}\right\|_{\ell_2}$ and then $\left\|\bar{e}_y\right\|_{\ell_2}$ are computable by subsequently using (A.9) and (A.8) into (A.5). It should be noted that for using the error estimate, the condition

$$1-\sigma_1\sigma_2\gamma^{V_2W_2}\gamma^{W_1V_1} > 0, \tag{A.12}$$

should hold, which depends on the system and boundary condition properties such as pump and choke characteristics; otherwise, the error estimate cannot be used (by applicability we mean that the condition (A.12) should be satisfied on the solution manifold). This condition originates from the small-gain condition on the interconnection of the system dynamics with the boundary dynamics [25], see Fig. 3.

To resolve this issue, we propose to use a hybrid error estimate; meaning that at those regions that the proposed error estimate does not exist due to violating the small-gain condition, we can switch to the error estimate presented in [24]. In the test cases of this paper, the condition (A.12) was always satisfied and this hybrid method has not been used. Other approaches will be investigated in future works.

References

[1] A. Antoulas, Approximation of large-scale dynamical systems, Society for Industrial and Applied Mathematics, 2005.
[2] J.S. Hesthaven, G. Rozza, B. Stamm, Certified reduced basis methods for parametrized partial differential equations, SpringerBriefs in Mathematics, Springer International Publishing, 2016.
[3] S. Gugercin, A.C. Antoulas, A survey of model reduction by balanced truncation and some new results, Internat. J. Control 77 (8) (2004) 748–766.
[4] S. Gugercin, A.C. Antoulas, C. Beattie, $\mathcal{H}_2$ model reduction for large-scale linear dynamical systems, SIAM J. Matrix Anal. Appl. 30 (2) (2008) 609–638.
[5] L. Iapichino, A. Quarteroni, G. Rozza, Reduced basis method and domain decomposition for elliptic problems in networks and complex parametrized geometries, Comput. Math. Appl. 71 (1) (2016) 408–430.
[6] J. Edmonds, Matroids and the greedy algorithm, Math. Program. 1 (1) (1971) 127–136.
[7] S. Lorenzi, A. Cammi, L. Luzzi, G. Rozza, POD-Galerkin method for finite volume approximation of Navier–Stokes and RANS equations, Comput. Methods Appl. Mech. Engrg. 311 (2016) 151–179.
[8] C. Daversin, C. Prud'homme, Simultaneous empirical interpolation and reduced basis method for non-linear problems, C. R. Math. 353 (12) (2015) 1105–1109.
[9] M.A. Grepl, Certified reduced basis methods for nonaffine linear time-varying and nonlinear parabolic partial differential equations, Math. Models Methods Appl. Sci. 22 (3) (2012) 1150015.
[10] M.A. Grepl, Model order reduction of parametrized nonlinear reaction–diffusion systems, Comput. Chem. Eng. 43 (2012) 33–44.
[11] B. Haasdonk, M. Ohlberger, Reduced basis method for finite volume approximations of parametrized linear evolution equations, ESAIM Math. Modelling Numer. Anal. 42 (2) (2008) 277–302.
[12] M. Ohlberger, S. Rave, Nonlinear reduced basis approximation of parameterized evolution equations via the method of freezing, C. R. Math. 351 (2013) 901–906.
[13] D. Rim, S. Moe, R. LeVeque, Transport reversal for model reduction of hyperbolic partial differential equations, SIAM / ASA J. Uncertain. Quantif. 6 (1) (2018) 118–150.
[14] J. Reiss, P. Schulze, J. Sesterhenn, V. Mehrmann, The shifted proper orthogonal decomposition: A mode decomposition for multiple transport phenomena, SIAM J. Sci. Comput. 40 (3) (2018) 1322—1344.

[15] A. Cosimo, A. Cardona, S. Idelsohn, Improving the k-compressibility of hyper reduced order models with moving sources: Applications to welding and phase change problems, Comput. Methods Appl. Mech. Engrg. 274 (2014) 237–263.
[16] M.D. Gunzburger, J.S. Peterson, J.N. Shadid, Reduced-order modeling of time-dependent PDEs with multiple parameters in the boundary data, Comput. Methods Appl. Mech. Engrg. 196 (4) (2007) 1030–1047.
[17] A. Cosimo, A. Cardona, S. Idelsohn, General treatment of essential boundary conditions in reduced order models for non-linear problems, Adv. Model. Simul. Eng. Sci. 3 (1) (2016) 3:7.
[18] H. Egger, T. Kugler, B. Liljegren-Sailer, N. Marheineke, V. Mehrmann, On structure-preserving model reduction for damped wave propagation in transport networks, SIAM J. Sci. Comput. 40 (1) (2017) 331—365.
[19] K.K. Fjelde, K.H. Karlsen, High-resolution hybrid primitive–conservative upwind schemes for the drift flux model, Comput. & Fluids 31 (3) (2002) 335–367.
[20] M.H. Abbasi, S. Naderi Lordejani, N. Velmurugan, C. Berg, L. Iapichino, W. Schilders, N. van de Wouw, A Godunov-type scheme for the drift flux model with variable cross section, J. Pet. Sci. Eng. 179 (2019) 796–813.
[21] S. Evje, K.K. Fjelde, Hybrid flux-splitting schemes for a two-phase flow model, J. Comput. Phys. 175 (2) (2002) 674–701.
[22] M. Billaud-Friess, A. Nouy, Dynamical model reduction method for solving parameter-dependent dynamical systems, SIAM J. Sci. Comput. 39 (4) (2016) 1766—1792.
[23] A. Janon, M. Nodet, C. Prieur, Certified reduced-basis solutions of viscous Burgers equation parametrized by initial and boundary values, ESIAM: Math. Modelling Numer. Anal. 47 (2) (2013) 317–348.
[24] Y. Zhang, L. Feng, S. Li, P. Benner, An efficient output error estimation for model order reduction of parametrized evolution equations, SIAM J. Sci. Comput. 37 (6) (2015) B910–B936.
[25] B. Besselink, N. van de Wouw, H. Nijmeijer, Model reduction for nonlinear systems with incremental gain or passivity properties, Automatica 49 (2013) 861–872.
[26] S.N. Lordejani, B. Besselink, M.H. Abbasi, G.O. Kaasa, W.H.A. Schilders, N. van de Wouw, Model order reduction for managed pressure drilling systems based on a model with local nonlinearities, IFAC-PapersOnLine 51 (8) (2018) 50–55.
[27] B. Besselink, N. van de Wouw, H. Nijmeijer, Model reduction for a class of convergent nonlinear systems, IEEE Trans. Automat. Control 57 (4) (2012) 1071–1076.
[28] A. Morin, G. Reigstad, Pipe networks: Coupling constants in a junction for the isentropic euler equations, in: 3rd Trondheim Gas Technology Conference, Energy Procedia, Trondheim, Norway, 2015, pp. 140–149.
[29] P. Colella, Multidimensional upwind methods for hyperbolic conservation laws, J. Comput. Phys. 87 (1) (1990) 171–200.
[30] S. Patankar, Numerical Heat Transfer and Fluid Flow, Taylor & Francis, 1980.
[31] R. Leveque, Finite Volume Methods for Hyperbolic Problems, Cambridge Texts in Applied Mathematics, 2002.
[32] A. Quarteroni, A. Manzoni, F. Negri, Reduced Basis Methods for Partial Differential Equations, in: La Matematica per il 3+2, Springer, 2016.
[33] P. Benner, T. Stykel, Model or Reduction of Differential-Algebraic Equations: A Survey, Surveys in Differential-Algebraic Equations IV, Springer, 2017.
[34] S. Chaturantabut, D. Sorensen, Nonlinear model reduction via discrete empirical interpolation, SIAM J. Sci. Comput. 32 (5) (2010) 2737–2764.
[35] H.K. Khalil, Nonlinear Systems, third ed., Pearson, Upper Saddle River, N.J, 2001.
[36] S. Skogestad, I. Postlethwaite, Multivariable Feedback Control: Analysis and Design, second ed., Wiley, 2005.
[37] N.A. Bruinsma, M. Steinbuch, A fast algorithm to compute the $\mathcal{H}_\infty$-norm of a transfer function matrix, Systems Control Lett. 14 (4) (1990) 287–293.
[38] M.N. Belur, C. Praagman, An efficient algorithm for computing the $\mathcal{H}_\infty$-norm, IEEE Trans. Automat. Control 56 (7) (2011) 1656–1660.