Error estimates for model order reduction of Burgers' equation [⋆]

M.H. Abbasi [*] **L. Iapichino** [*] **B. Besselink** [**] **W. Schilders** [*]
N. van de Wouw [***,****]

[*] *Department of Mathematics and Computer Science, Eindhoven University of Technology, The Netherlands (e-mail: m.h.abbasi@tue.nl).*
[**] *Bernoulli Institute for Mathematics, Computer Science and Artificial Intelligence, University of Groningen, The Netherlands*
[***] *Department of Mechanical Engineering, Eindhoven University of Technology, The Netherlands*
[****] *Department of Civil, Environmental and Geo-Engineering, University of Minnesota, U.S.A.*

Abstract: Burgers' equation is a nonlinear scalar partial differential equation, commonly used as a testbed for model order reduction techniques and error estimates. Model order reduction of the parameterized Burgers' equation is commonly done by using the reduced basis method. In this method, an error estimate plays a crucial rule in both accelerating the offline phase and quantifying the error induced after reduction in the online phase. In this study, we introduce two new estimates for this reduction error. The first error estimate is based on a Lur'e-type model formulation of the system obtained after the full-discretization of Burgers' equation. The second error estimate is built upon snapshots generated in the offline phase of the reduced basis method. The second error estimate is applicable to a wider range of systems compared to the first error estimate. Results reveal that when conditions for the error estimates are satisfied, the error estimates are accurate and work efficiently in terms of computational effort.

Keywords: Error estimate, Reduced basis method, Model order reduction, Nonlinear systems, Burgers' equation.

1 Introduction

Model order reduction of high-fidelity models is a necessary tool for enabling real-time simulation and controller design. These high-fidelity models are often the result of the discretization of Partial Differential Equations (PDEs) governing the physical phenomena. One way to reduce these models is the Reduced Basis (RB) method (Haasdonk and Ohlberger [2008]), consisting of decomposed offline and online phases. In the offline phase of the RB method, RB functions for approximating the solution are generated. This phase contains computations whose complexity scale with the degrees of freedom of the original system, thus it is computationally expensive. In the online phase, the solution is approximated by a linear combination of the RB functions. The computations in this phase scale with the number of RB functions generated in the offline phase, which renders obtaining the solution of the reduced model computationally efficient. However, replacing a model with its reduced version leads to an error between the solution of the full-order model and the reduced one. To ensure the accuracy of the reduced solution, an error bound or estimate should be provided. In the RB context, the benefits of having such an error bound or estimate are twofold. First, an error bound (or estimate) in the RB technique can be used to accelerate the offline phase during the greedy algorithm (Abbasi et al. [2020]). Second, it certifies the accuracy of the solution that is obtained during the online phase. Therefore, developing a sharp error bound (or an accurate error estimate) is crucial within this approach.

To build an efficient yet accurate reduced-order model by the RB method and decompose the offline and online phases, nonlinear problems are hyper-reduced by using the Empirical Interpolation Method (EIM) (Barrault et al. [2004]) or its discrete counterpart, the Discrete Empirical Interpolation Method (DEIM) (Chaturantabut and Sorensen [2010]), combined afterwards with the RB method itself. EIM and DEIM require additional basis functions (called collateral basis functions) to approximate the nonlinear functions and these collateral basis functions are usually generated in the offline phase before the generation of the RB functions, which makes the offline phase even more expensive. To reduce the computation time, the collateral basis functions can be generated in parallel to the RB functions. To synchronize the RB function generation and the collateral basis function generation, various algorithms have been introduced; e.g. the PODEI algorithm by Drohmann et al. [2012]. The inaccurate approximation of the nonlinear functions also plays a role

[⋆] 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.

in the final error induced by reduction, which has to be taken into account when building error estimates. To generate both collateral basis functions and RB functions, the solution snapshots of the full-order system of equations should be available.

In this paper, we focus on a hyperbolic PDE, Burgers' equation. Hyperbolic systems are commonly solved by Finite-Volume (FV) techniques that lead to state-space models of high order. The work on error bounds (or estimates) in the RB community for hyperbolic systems is still in the evolutionary stage, see Haasdonk and Ohlberger [2008], Zhang et al. [2015], Abbasi et al. [2020] for some works. Methods introduced in these works are typically tailored to linear systems and not efficient if applied to nonlinear systems. Moreover, most of these techniques (except the method by Abbasi et al. [2020]) utilize the norm of the state matrix of the discretized system. If the state matrix has a large norm (larger than one), these error bounds (estimates) are not valid and grow exponentially over time. The method introduced by Abbasi et al. [2020] (which also works well if applied to systems with local nonlinearities) circumvents this issue by using the ℓ_2-norm of the system with respect to its inputs and outputs, as similarly done in the balanced truncation method by Naderi Lordejani et al. [2018], Besselink et al. [2012, 2013]. In general, theoretical error estimates for nonlinear systems are lacking in the RB literature. In this paper, we aim to extend the methodology introduced in Abbasi et al. [2020] from systems with local nonlinearities to systems with distributed nonlinearities. However, the error estimate of Abbasi et al. [2020] cannot be efficiently used when strong nonlinearities (nonlinearities with high Lipschitz constant) are present in the system.

Therefore, in addition to the error estimate based on the ℓ_2-gain notion, an empirical error estimate is also introduced in this paper. This estimate is based on the snapshots generated in the offline phase of the RB method. This estimate does not suffer from restrictions of the previous error estimate. Most importantly, it does not require the residual calculation and it is tailored in a way that its computation is efficient, similar to the computation of the reduced-order solution.

The structure of this paper is as follows. In Section 2, Burgers' equation together with its discretization, which leads to the full-order model, is introduced. In Section 3, the model-order reduction approach used to obtain the reduced-order model is elaborated. In Section 4, the two error estimates for the nonlinear reduced-order model are discussed. In Section 5, numerical results are presented. Finally, Section 6 concludes the paper.

2 Burgers' equation

One of the simplest and yet fundamental nonlinear equations describing a conservative system is Burgers' equation, which is sometimes referred to as the scalar version of the Navier-Stokes equations (Orlandi [2000]). This equation is defined as

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left(f(u)\right) = 0,\ t \in [0,T],\ x \in [0,L], \tag{1}$$

where $u := u(t,x;\boldsymbol{\mu})$ is the conservative variable and $f(u) = u^2/2$ is the flux function associated with Burgers' equation. Here, t represents time and T is the time horizon of the simulation. In addition, x denotes the spatial coordinate and L is the length of the spatial domain. Finally, $\boldsymbol{\mu} \in \mathcal{D}$ is a vector of parameters used in (1) that varies in a multi-query analysis within the parameter domain $\mathcal{D} \in \mathbb{R}^R$, with R the number of varying parameters. We assume that the initial condition and boundary condition are represented by these varying parameters. For the initial condition, we assume $u(0,x;\boldsymbol{\mu}) = \mu_1$, which is constant over the spatial domain. For the boundary condition at $x = 0$, we assume

$$u(t,0;\boldsymbol{\mu}) = \begin{cases} \mu_1, & t = 0, \\ \mu_2, & t > 0. \end{cases} \tag{2}$$

Therefore, in this study, we have $\boldsymbol{\mu} = [\mu_1, \mu_2]$.

Discretizing (1) with the Lax-Friedrichs scheme (see Lax [1954], Friedrichs [1954]) leads to

$$\boldsymbol{U}^{n+1} = L_{lin}\boldsymbol{U}^n + BU_0^n - \frac{\Delta t}{4\Delta x}L_{nl}(\boldsymbol{U}^n)^2 + \frac{\Delta t}{2\Delta x}B(U_0^n)^2, \tag{3}$$

where $\boldsymbol{U}^n := [U_1^n, \cdots, U_{\mathcal{N}}^n]^T \in \mathbb{R}^{\mathcal{N}}$ is the vector containing U_i^n, the average of the conservative variable u over the i-th grid cell at the time instant $t^n := n\Delta t, n = \{0, \cdots, N_t\}$ with N_t number of time steps. Here, Δt and Δx refer to the temporal and spatial discretization intervals over time and space, respectively. The spatial discretization consists of cells $(x_{i-1/2}, x_{i+1/2}), i = 1, \cdots, \mathcal{N}$, with the length of Δx centered at $x_i = x_{i-1/2} + \Delta x/2$ and $\mathcal{N}$ spatial grid cells. Furthermore, $L_{lin}, L_{nl} \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}}$ are the operators acting on the linear and nonlinear part of the system that emerge after applying the full-discretization. Also, $U_0^n \in \mathbb{R}$ is the value of the conservative variable at the boundary $x = 0$ acting as the input into the system defined according to (2) and $B \in \mathbb{R}^{\mathcal{N}}$ is the input matrix corresponding to the boundary input. Moreover, the square operator $(\cdot)^2$ in (3) is interpreted element-wise. The nonlinearity associated with this equation is $g(U) = (U)^2$, where $g(\cdot)$ is a nonlinear operator. Then, system (3) is equivalent to the system Σ depicted in the left side of Figure 1, which comprises a linear subsystem Σ_{lin} and nonlinear subsystem Σ_{nl} given by

$$\begin{aligned} \Sigma_{lin} : &\begin{cases} \boldsymbol{U}^{n+1} = L_{lin}\boldsymbol{U}^n + BU_0^n - \dfrac{\Delta t}{4\Delta x}L_{nl}\boldsymbol{U}_{nl}^n \\ \qquad\quad + \dfrac{\Delta t}{2\Delta x}B(U_0^n)^2, \\ y^n = C_y\boldsymbol{U}^n, \\ z^n = \boldsymbol{U}^n, \end{cases} \\ \Sigma_{nl} : &\ \boldsymbol{U}_{nl}^n = g(z^n) = (z^n)^2. \end{aligned} \tag{4}$$

Here, $y \in \mathbb{R}^w$ is the output of interest of the system (for instance, y can be the value of the conservative variable at the right-end of the spatial domain with $w = 1$) and $C_y \in \mathbb{R}^{w\times\mathcal{N}}$ is the corresponding output matrix. This full-order model has large dimension (i.e., $\mathcal{N}$ is large). Therefore, real-time simulations cannot be achieved unless powerful computational resources are at the disposal. Moreover, control design for such a complex system is generally infeasible. Hence, model order reduction should be applied to (4), which is the topic of the next section. The following assumption will be used throughout the paper.

Assumption 1. The system matrix L_{lin} is Schur for all $\boldsymbol{\mu} \in \mathcal{D}$, i.e., Σ_{lin} in (4) is internally asymptotically stable.

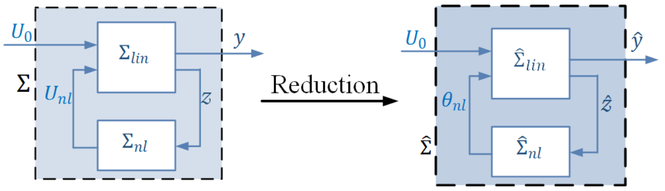

Fig. 1. Left: The schematic representation of the the interconnection between the linear dynamics and the static nonlinearity. Right: The schematic representation of the interconnection between the reduced linear subsystem (obtained by the RB method) and the reduced nonlinear function (number of nonlinear equations is reduced by (D)EIM).

3 Model reduction

This section subsequently discusses the RB method, (D)EIM, and their combination, leading to a method for hyper-reduction of the nonlinear system (4).

3.1 Reduced basis method

A powerful method for dimension reduction of a parameter-dependent dynamical system is the RB method. In the RB method, the system of equations is projected into a low dimensional space spanned by the solutions of the full-order model for specific members of the parameter domain.

As discussed in Abbasi et al. [2020], handling time-varying boundary conditions within the RB method is vital as the (time-varying) control inputs commonly act at the boundaries. Tailoring the method in Abbasi et al. [2020] to our case study, we introduce the RB ansatz

$$\hat{\boldsymbol{U}}^n(\boldsymbol{\mu}) = U_0^n(\boldsymbol{\mu})\mathbf{1} + \Phi a^n, \tag{5}$$

where $\hat{\boldsymbol{U}}^n \in \mathbb{R}^{\mathcal{N}}$ is the solution of the reduced-order model, and $\mathbf{1} \in \mathbb{R}^{\mathcal{N}}$ is a vector of ones that enables the RB solution $\hat{\boldsymbol{U}}^n(\boldsymbol{\mu})$ to satisfy the boundary condition (2) at all time instants. Then, the RB functions $\Phi \in \mathbb{R}^{\mathcal{N}\times N}$ should vanish at the location of the specified boundary condition (i.e., $\Phi|_{x=0}= 0,$), where N is the number of RB functions. Here, $a^n \in \mathbb{R}^N$ is the modal coordinate associated with the RB functions, which is the state of the reduced-order model. To generate the RB functions Φ vanishing at the location of the specified boundary, we modify the snapshots during the greedy algorithm for a selected parameter $\boldsymbol{\mu}^*$ and then apply the Proper Orthogonal Decomposition (POD) (Hesthaven et al. [2016]) on these modified snapshots (see Algorithm 1), defined as

$$\begin{aligned} \hat{\boldsymbol{U}}^{n,*}(\boldsymbol{\mu}^*) &= \hat{\boldsymbol{U}}^n(\boldsymbol{\mu}^*) - U_0^n(\boldsymbol{\mu}^*)\mathbf{1}, \\ \hat{\boldsymbol{U}}^*(\boldsymbol{\mu}^*) &= \{\hat{\boldsymbol{U}}^{n,*}(\boldsymbol{\mu}^*)\}, \quad \forall\, n = \{0, \cdots, N_t\}. \end{aligned} \tag{6}$$

Finally, "POD($\hat{\boldsymbol{U}}^*(\boldsymbol{\mu}^*), 1$)" obtained from Algorithm 1 yields an RB function. For more details, we refer to Abbasi et al. [2020].

3.2 Empirical interpolation method

To handle the nonlinearities in (4), EIM is applied as in Barrault et al. [2004]. By using this method, a nonlinear function is replaced by a linear interpolation of collateral basis functions (basis functions generated by the EIM/DEIM), which are obtained during the offline phase. In the online phase, the coefficients for the linear interpolation of the collateral basis functions are chosen such that this interpolation becomes exact at some preselected points, the so-called interpolation points, along the spatial domain. The effect of the nonlinear function is then fed back into the linear system via the feedback interconnection as shown in the right side of Figure 1.

After applying EIM, the nonlinear function in (4) is approximated by a linear interpolation

$$(\boldsymbol{U}^n)^2 \approx q_{nl}\theta_{nl}^n, \tag{7}$$

where $q_{nl} \in \mathbb{R}^{\mathcal{N}\times M}$ is the matrix of collateral basis functions and $\theta_{nl} \in \mathbb{R}^M$ are the unknown coefficients of the collateral basis functions, to be calculated online. The collateral basis functions q_{nl} are obtained by applying POD (Algorithm 1) on the snapshots of the nonlinearities $g(z_n)$ for specific members of the parameter domain during the offline phase. The coefficients θ_{nl} in (7) are obtained during the online phase such that the interpolation is exact at M pre-selected points $\boldsymbol{X_m} = \{x_1, \cdots x_M\}$ where $x_i \in \mathbb{R}$ is the grid-cell number of the interpolation point selected at the i-th iteration (the selection procedure of such points is introduced later in Algorithm 2). Specifically, let $P = [e_{x_1}, \cdots e_{x_M}] \in \mathbb{R}^{\mathcal{N}\times M}$ where e_i is the i-th column of the identity matrix (of dimension $\mathcal{N}\times\mathcal{N}$). For the points $\boldsymbol{X_m}$, we have

$$(P^T\boldsymbol{U}^n)^2 = P^T q_{nl}\theta_{nl}^n, \tag{8}$$

stating that the interpolation is exact at X_m if $P^T q_{nl}$ is non-singular (θ_{nl}^n can then be computed from (8)). After approximating the nonlinearities in (4) with linear interpolation of the collateral basis functions, we can apply a Galerkin projection (Haasdonk and Ohlberger [2008]) to the system of equations, as explained in the next section.

3.3 RB-EIM combination

After applying EIM to the nonlinear parts of the dynamics, all the operators involved in the full-order model become linear and therefore the system can be efficiently projected onto a lower-dimensional subspace spanned by Φ. Substituting the ansatz (5) and the EIM approximation (7) in (4), applying a Galerkin projection on the resulting system and taking into account the orthogonality of the basis functions Φ, we obtain the reduced-order model

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

```
Input: Snapshots $\boldsymbol{U}(\boldsymbol{\mu}) \in \mathbb{R}^{\mathcal{N}\times N_t}$, number of basis vectors $n_{\text{POD}}$
Output: $\phi \in \mathbb{R}^{\mathcal{N}\times n_{\text{POD}}}$
Perform a Singular Value Decomposition on the snapshots, $\boldsymbol{U} = U_{\text{SVD}}SV$
$\phi = U_{\text{SVD}}(:, 1 : n_{\text{POD}})$ is the first $n_{\text{POD}}$ vectors of the left singular vectors $U_{\text{SVD}}$.
```

Algorithm 2 PODEI-Greedy algorithm

```
Input: $\mathcal{D}_h$(discretized version of $\mathcal{D}$), $N, \boldsymbol{\mu}^1 \in \mathcal{D}_h$
Output: $\Phi, q_{nl}, \boldsymbol{X_m}, B_{nl}$
Set $\Phi = \{\}, q_{nl} = \{\}, \boldsymbol{X_m} = \{\}, B_{nl} = 1,$
for k=1 to N-1 do
    Solve (4) for $\boldsymbol{\mu}^k$ to obtain $\boldsymbol{U} = [\boldsymbol{U}^0, \cdots, \boldsymbol{U}^{N_t}]$ and $\boldsymbol{U}_{nl} = [\boldsymbol{U}_{nl}^0, \cdots, \boldsymbol{U}_{nl}^{N_t}]$
    Generate $\boldsymbol{U}^* = \boldsymbol{U} - [U_0^0, \cdots, U_0^{N_t}]\mathbf{1}$ and $\bar{\boldsymbol{U}}_{nl} = \boldsymbol{U}_{nl} - q_{nl} B_{nl}^{-1} \boldsymbol{U}_{nl}(\boldsymbol{X_m}, :)$
    Set $\bar{\boldsymbol{U}} = \boldsymbol{U}^* - \Phi\Phi^T \boldsymbol{U}^*$
    $\Phi \leftarrow \text{orth}\left\{\Phi \cup \text{POD}(\bar{\boldsymbol{U}}, 1)\right\}$ and $q_{POD} = \text{POD}(\bar{\boldsymbol{U}}_{nl}, 1)$
    $\sigma_M = (q_{nl}(\boldsymbol{X_m}, :))^{-1} q_{POD}(\boldsymbol{X_m})$
    $r_d = q_{POD} - q_{nl}\sigma_M$
    $\boldsymbol{X_m} \leftarrow \{\boldsymbol{X_m} \cup (r_d)\},$
    $q_{nl} \leftarrow \{q_{nl} \cup \frac{r_d}{\max r_d}\}, B_{nl} = q_{nl}(\boldsymbol{X_m}, :)$
    Based on $\Phi$ and $q_{nl}$, perform the error estimates to find the worst approximated solution and find $\boldsymbol{\mu}^{k+1}$ and $e(\boldsymbol{\mu}^{k+1})$
    if $e(\boldsymbol{\mu}^{k+1}) > e(\boldsymbol{\mu}^k)$ then
        $q_{nl} = q_{nl}(:, 1 : \text{end} - 1)$, $\boldsymbol{X_m} = \boldsymbol{X_m}(1 : \text{end} - 1)$, $B_{nl} = q_{nl}(\boldsymbol{X_m}, :)$
    end if
end for
```

$$\hat{\Sigma}_{lin} : \begin{cases} a^{n+1} = \hat{L}_{lin} a^n + (\hat{B} + \hat{L}_{BC}) U_0^n - \dfrac{\Delta t}{4\Delta x} \hat{L}_{nl} \theta_{nl}^n \\ \qquad + \dfrac{\Delta t}{2\Delta x} \hat{B} (U_0^n)^2 - \Phi_{\mathbf{1}} U_0^{n+1}, \\ \hat{y}^n = U_0^n C_y \mathbf{1} + C_y \Phi a^n, \\ z_m^n = U_0^n P^T \mathbf{1} + P^T \Phi a^n, \end{cases}$$
$$\hat{\Sigma}_{nl} : \begin{cases} \boldsymbol{U}_{nl}^n = g(z_m^n) = (z_m^n)^2, \\ \theta_{nl}^n = (P^T q_{nl})^{-1} \boldsymbol{U}_{nl}^n, \end{cases} \tag{9}$$

where $\hat{y}$ is an approximation of y and $\hat{L}_{lin} = \Phi^T L_{lin} \Phi$, $\hat{L}_{nl} = \Phi^T L_{lin} q_{nl}$, $\hat{B} = \Phi^T B$, $\hat{L}_{BC} = \Phi^T L_{lin} \mathbf{1}$, and $\Phi_{\mathbf{1}} = \Phi^T \mathbf{1}$. Finally, z_m^n is the value of the reduced solution at the pre-selected points $\boldsymbol{X_m}$. None of the computations in (9) scales with the actual degrees of freedom $\mathcal{N}$ and therefore the model is in a reduced form. It should be noted that the boundary-related terms in (9) (such as $U_0^n P^T \mathbf{1}$) are due to the ansatz used in (5) and the segregation of the boundary condition from the solution. In this work, to synchronize the generation of the RB functions Φ and the collateral basis functions q_{nl}, the PODEI algorithm in Drohmann et al. [2012] is used, which is mentioned in Algorithm 2 together with the greedy algorithm and the selection of the interpolation points. Now, the reduced-order model is available and the error estimates can be introduced.

4 Error estimates

In this section, we introduce two types of error estimates. In the first one, we build the error dynamics and propose an estimate based on the ℓ_2-gain notion. In the second one, we use the solutions of the full-order model generated in the offline phase to obtain an empirical error estimate.

4.1 Error estimate based on the ℓ_2-gain notion

As shown in (9), the interconnection of the RB method and EIM can be represented as a Lur'e-type system as shown in the right side of Figure 1. The error estimate introduced here relies on the notion of small-gain condition of the error dynamics (Besselink et al. [2012]), to be introduced here. If this condition is not satisfied, the error estimate presented here cannot be used. To enable cheap computation of the residual, the following assumption is used.

Assumption 2. (Drohmann et al. [2012]). We assume the exactness of the EIM approximation for a certain number of collateral RB functions; i.e., there exists a positive integer $M^* > M$ with the set of enriched collateral basis functions by q_{nl}^* and the corresponding coefficients by ${\theta_{nl}^*}^n$, such that

$$(\hat{\boldsymbol{U}}^n(\mu))^2 = q_{nl}^* {\theta_{nl}^*}^n(\mu) \quad \forall n = 1, \cdots, N_t, \quad \text{and} \quad \mu \in \mathcal{D}. \tag{10}$$

Statement 1. Let $\boldsymbol{U}^n$ be obtained from (4) and $\hat{\boldsymbol{U}}^n$ be obtained from (9) and (5) with $n = 1, \cdots, N_t$ under the same initial condition and the same boundary input U_0^n. We define the residual $\mathcal{R}^n$ by inserting the RB solution $\hat{\boldsymbol{U}}^n$ into (4) as follows:

$$\begin{aligned} \mathcal{R}^n =& \hat{\boldsymbol{U}}^{n+1} - \Big(L_{lin} \hat{\boldsymbol{U}}^n + B U_0^n - \frac{\Delta t}{4\Delta x} L_{nl} g(\hat{\boldsymbol{U}}^n) \\ & + \frac{\Delta t}{2\Delta x} B (U_0^n)^2 \Big). \end{aligned} \tag{11}$$

We assume the Lipschitz continuity L_g for the nonlinear function $\boldsymbol{e}_g^n = g(\boldsymbol{e}^n)$, i.e., $\left\| \boldsymbol{e_g^n} \right\| \leq L_g \left\| \boldsymbol{e^n} \right\|$. An estimate of the error bound of $\|\boldsymbol{e}_y\|_{\ell_2}$ with $\boldsymbol{e}_y := y - \hat{y}$ is given by

$$\begin{aligned} \|\boldsymbol{e_y}\|_{\ell_2} &\leq \kappa(\boldsymbol{\mu}) \|\mathcal{R}\|_{\ell_2} \\ \text{with} \quad \kappa(\boldsymbol{\mu}) &:= \gamma^{\boldsymbol{e_y}\mathcal{R}} + \frac{\gamma^{\boldsymbol{e_y e_g}} L_g \gamma^{\boldsymbol{e}\mathcal{R}}}{1 - L_g \gamma^{\boldsymbol{e e_g}}}, \end{aligned} \tag{12}$$

with $\|(\cdot)\|_{\ell_2} := \sqrt{\sum\limits_{n=0}^{\infty} \|(\cdot)\|^2}$ and γ^{yu} denoting the ℓ_2-norm of the system from input u to the output y.

Derivation: To define the error estimate, the error dynamics is defined by subtracting (11) from the full-order model (3)

$$\boldsymbol{e}^{n+1} = L_{lin} \boldsymbol{e}^n - \frac{\Delta t}{4\Delta x} L_{nl} ((\boldsymbol{U}^n)^2 - (\hat{\boldsymbol{U}}^n)^2) - \mathcal{R}^n, \tag{13}$$

with $\boldsymbol{e} := \boldsymbol{U} - \hat{\boldsymbol{U}}$. By denoting $(\boldsymbol{U}^n)^2 - (\hat{\boldsymbol{U}}^n)^2$ as $\boldsymbol{e}_g^n$ and rewriting the dynamics in the feedback interconnected form, we obtain the error system Σ^e with its linear and nonlinear subsystems given as follows:

$$\begin{aligned} \Sigma_{lin}^e &: \begin{cases} \boldsymbol{e}^{n+1} = L_{lin} \boldsymbol{e}^n + \dfrac{\Delta t}{4\Delta x} L_{nl} \boldsymbol{e}_g^n - \mathcal{R}^n, \\ \boldsymbol{e}_y^n = C_y \boldsymbol{e}^n, \\ \boldsymbol{e}_z^n = \boldsymbol{e}^n, \end{cases} \\ \Sigma_{nl}^e &: \boldsymbol{e}_g^n = f(\hat{\boldsymbol{U}}, \boldsymbol{e}_z) = g(\boldsymbol{e}_z + \hat{\boldsymbol{U}}) - g(\hat{\boldsymbol{U}}). \end{aligned} \tag{14}$$

This feedback interconnection is depicted in Figure 2. Notably, the relation in Σ_{nl}^e holds regardless of using EIM as we have already lifted the solution to the full-order space. The effect of inaccurate approximation of the nonlinearities plays a role in the residual calculation, which is explained later in this section.

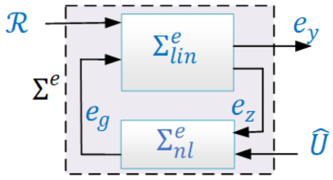

Fig. 2. The schematic representation of the feedback interconnection for the error dynamics.

In the online phase, however, we do not have access to the values for $(\boldsymbol{U}^n)^2$ since the actual solution is not known. Therefore, an estimation of the output should be defined as we cannot simulate these error dynamics in a computationally efficient manner.

Following the idea introduced by Abbasi et al. [2020] for linear systems and assuming Σ_{lin} is asymptotically stable (Assumption 1), an error bound on the ℓ_2-norm of the error signal can be computed as follows:

$$\|\boldsymbol{e_y}\|_{\ell_2} \le \gamma^{\boldsymbol{e_y}\mathcal{R}} \|\mathcal{R}\|_{\ell_2} + \gamma^{\boldsymbol{e_y}\boldsymbol{e_g}} \|\boldsymbol{e_g}\|_{\ell_2}. \tag{15}$$

This ℓ_2-norm is equal to the $\mathcal{H}_\infty$-norm of the linear system (14) with respect to the same input and output (Khalil [2001]). Apart from the gains, in order to compute this error bound, both $\|\mathcal{R}^n\|$ and $\|\boldsymbol{e_g}^n\|$ should be computed in a computationally efficient manner.

To compute the norm of the residual, we decompose the residual into a linear and a nonlinear part as below:

$$\mathcal{R}^n = \mathcal{R}^n_{lin} + \mathcal{R}^n_{nl}, \tag{16}$$

where

$$\begin{aligned} \mathcal{R}^n_{lin} &= \hat{\boldsymbol{U}}^{n+1} - \Big(L_{lin}\hat{\boldsymbol{U}}^n + BU_0^n - \frac{\Delta t}{4\Delta x} L_{nl} q_{nl} \theta^n_{nl} \\ &\qquad + \frac{\Delta t}{2\Delta x} B (U_0^n)^2 \Big), \qquad\qquad (17) \\ \mathcal{R}^n_{nl} &= -\frac{\Delta t}{4\Delta x} L_{nl} (q_{nl}\theta^n_{nl} - (\hat{\boldsymbol{U}}^n)^2). \end{aligned}$$

In computing the two-norm of the residual $\mathcal{R}^n$, it is necessary to compute $\mathcal{R}^n_{nl}$, which is time-consuming due to the presence of the nonlinear term $(\hat{\boldsymbol{U}}^n)^2$. To avoid this computational issue, following the idea presented by Drohmann et al. [2012], this term is calculated empirically by using Assumption 2. This assumption requires the reduced-order problem to be solved once more with an enriched set of collateral basis functions. Employing this assumption in the equation governing $\mathcal{R}^n_{nl}$ leads to

$$\mathcal{R}^n_{nl} = -\frac{\Delta t}{4\Delta x} L_{nl} (q_{nl}\theta^n_{nl} - q^*_{nl} {\theta^*_{nl}}^n). \tag{18}$$

The other required quantity for calculating the error estimate via (15) is $\|\boldsymbol{e_g}\|_{\ell_2}$. As $\boldsymbol{e^n_g} := \boldsymbol{e^n}$ represents the error in approximating the nonlinear function, we have

$$\left\|\boldsymbol{e^n_g}\right\| \le L_g \left\|\boldsymbol{e^n}\right\|, \tag{19}$$

where L_g is an approximation of the local Lipschitz constant of the nonlinear operator g. The inequality (19) implies

$$\|\boldsymbol{e_g}\|_{\ell_2} \le L_g \|\boldsymbol{e}\|_{\ell_2}. \tag{20}$$

Similar to (15), we have

$$\|\boldsymbol{e}\|_{\ell_2} \le \gamma^{\boldsymbol{e}\mathcal{R}} \|\mathcal{R}\|_{\ell_2} + \gamma^{\boldsymbol{e}\boldsymbol{e_g}} \|\boldsymbol{e_g}\|_{\ell_2}. \tag{21}$$

Combining (20) and (21), and assuming that the small-gain condition $L_g\gamma^{\boldsymbol{e}\boldsymbol{e_g}} < 1$ holds, leads to

$$\|\boldsymbol{e}\|_{\ell_2} \le \frac{\gamma^{\boldsymbol{e}\mathcal{R}}}{1 - L_g\gamma^{\boldsymbol{e}\boldsymbol{e_g}}} \|\mathcal{R}\|_{\ell_2}. \tag{22}$$

Finally, the use of this result in (15) gives (12). ∎

Remark 1. Exact satisfaction of Assumption 2 requires $M^* = \mathcal{N}$, which renders the error estimate expensive. In the results presented in this paper, we set $M^* = M + 1$. Therefore, $\|\mathcal{R}^n\|$ is computed cheaply and the ℓ_2-norm can be calculated.

Remark 2. To compute ${\mathcal{R}^n}^T\mathcal{R}^n$, some operators such as $\Phi^T L_{lin} L_{nl} q_{nl} \in \mathbb{R}^{N\times M}$ should be pre-computed during the offline phase and stored for usage during the online phase. Now, the two-norms of $\mathcal{R}^n$ can be computed with computations that scale at most with the dimension of q^*_{nl} or Φ, which is still much lower than the number of actual degrees of freedom of the high-fidelity scheme. For the details of residual calculation, we refer to Abbasi et al. [2020]. Using Remark 1 instead of $M^* = \mathcal{N}$ renders the bound (12) to be an error estimate, not an actual error bound.

Remark 3. As the nonlinear operator for Burgers' equation $g(U) = (U)^2$ is not globally Lipschitz, we have to restrict the solution domain to be able to define a finite L_g. Note that the inequality (19) holds only locally as the value of L_g depends on the magnitude of $\boldsymbol{U}$, which restricts the range of $\boldsymbol{U}$ in the simulations. Assuming $\boldsymbol{e^n_z} := \boldsymbol{e^n}$ to be small and estimating the Lipschitz constant by the derivative of the nonlinear function $L_g = 2\max\limits_{i,n} U^n_i$ reveals that

$$\max_{i,n} U^n_i < \frac{1}{2\gamma^{\boldsymbol{e}\boldsymbol{e_g}}}, \tag{23}$$

ensures that the small-gain condition in (14) is satisfied.

To enlarge and shift the applicability region, a loop transformation can be pursued as follows.

4.1.1 *Loop transformation*

The range of the applicability of the small-gain condition can be enlarged by using a so-called loop transformation (see Khalil [2001]). In this section, we aim to apply this transformation to the feedback interconnection in (14) induced by the EIM and RB methods.

The loop transformation changes the interconnection in Figure 2 to Figure 3. The error dynamics after the loop transformation can be written as

$$\begin{aligned} \Sigma^{e,\epsilon}_{lin} &: \begin{cases} \boldsymbol{e}^{n+1} = (L_{lin} - \epsilon\dfrac{\Delta t}{4\Delta x} L_{nl})\boldsymbol{e}^n + \dfrac{\Delta t}{4\Delta x} L_{nl}\boldsymbol{e}^n_g - \mathcal{R}^n, \\ \boldsymbol{e}^n_y = C_y\boldsymbol{e}^n, \\ \boldsymbol{e}^n_z = \boldsymbol{e}^n, \end{cases} \\ \Sigma^{e,\epsilon}_{nl} &: \boldsymbol{e}^n_g = g(\boldsymbol{e}_z + \hat{\boldsymbol{U}}) - g(\hat{\boldsymbol{U}}) + \epsilon\boldsymbol{e}_z. \end{aligned} \tag{24}$$

It should be noted that Σ^e in (14) and (24) are exactly the same. The constant ϵ should be defined such that it minimizes the product $L_g\gamma^{ee_g}$ and therefore enlarges the applicability region while also reducing the conservatism in the small-gain condition and the estimate (12). For the parameterized system (24), the following minimization problem is solved to obtain ϵ,

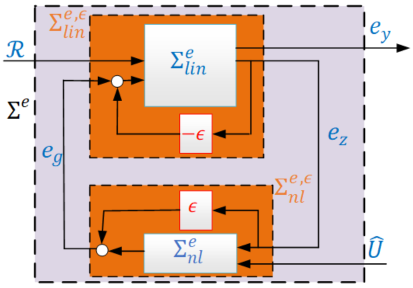

Fig. 3. The schematic representation of the feedback interconnection within the error dynamics after loop transformation.

$$\epsilon = \arg\min_{\epsilon}\left(\sum_i \left(\max\left(|2\boldsymbol{U}(\boldsymbol{\mu}^i)+\epsilon|\right)\times\gamma^{e_z e_g}(\boldsymbol{\mu}^i)\right)\right)$$

$$\text{s.t.}\quad \forall\boldsymbol{\mu}^i\in\mathcal{D}_h\quad\begin{cases}\rho(L_{lin}-\epsilon\dfrac{\Delta t}{4\Delta x}L_{nl})<1\\ \max(|2\boldsymbol{U}(\boldsymbol{\mu}^i)+\epsilon|)\times\gamma^{e_z e_g}(\boldsymbol{\mu}^i)<1\end{cases}\tag{25}$$

where $\rho(\cdot)$ is the spectral radius of a matrix and $\mathcal{D}_h$ is the discrete version of the varying parameter domain $\mathcal{D}$. For the test case under study, we have designed the experiments such that

$$\min(\mu_1,\mu_2)\leq\boldsymbol{U}^n(\boldsymbol{\mu}^i)\leq\max(\mu_1,\mu_2).\tag{26}$$

The constraints in the minimization problem (25) ensure that for each parameter setting, first, the linear part of the error dynamics $\Sigma^{e,\epsilon}_{lin}$ is stable, and second, the interconnection of the linear subsystem $\Sigma^{e,\epsilon}_{lin}$ and the nonlinear subsystem $\Sigma^{e,\epsilon}_{nl}$ is also stable. In order to render the computations tractable, we terminate the minimization problem as soon as the constraints are satisfied.

Due to the fact that the nonlinear part of the system is not globally Lipschitz, a restriction on the region of the solution still holds after determining ϵ. In other words, to satisfy the small-gain condition, for all members of the parameter domain, we require (based on the second constraint in (25))

$$-\frac{1}{2\gamma^{e_z e_g}(\boldsymbol{\mu})}-\frac{1}{2}\epsilon<u(x,t;\boldsymbol{\mu})<\frac{1}{2\gamma^{e_z e_g}(\boldsymbol{\mu})}-\frac{1}{2}\epsilon.\tag{27}$$

Therefore, the parameters, boundary conditions and initial conditions should be chosen in a way that the satisfaction of (27) would be possible. Based on the knowledge of the dependence of the ℓ_2-gains on ϵ and the variation of initial and boundary conditions, one can a priori have an insight whether this condition can be satisfied or not.

However, the error estimate (12), even with this loop transformation, can lead to conservative results. To alleviate the conservativeness, we tighten (sharpen) the error estimate as below.

4.1.2 Sharpening the error estimate To resolve the problems of expensive calculation of the ℓ_2-gain of the system and conservativeness of the error estimate, we follow Abbasi et al. [2020]. The main idea is that in the offline phase, the average of the conservatism of the error estimate is known and we can sharpen the error estimate in the online phase according to the experience in the offline phase. To do so, the error gain κ in (12) is multiplied by a reduction factor to obtain

$$\kappa^{e_y}(\boldsymbol{\mu})=\bar{\rho}\kappa(\boldsymbol{\mu}),\tag{28}$$

where $\kappa^{e_y}\|\mathcal{R}\|_{\ell_2}$ is an estimate of $\|\boldsymbol{e}_y\|_{l_2}$, which is calculated based on κ in (12). To define $\bar{\rho}$, we first introduce the variable $\bar{\rho}^f_i$ as a measure of the conservatism

$$\bar{\rho}^f_i=\frac{\left\|\boldsymbol{e_y}(\boldsymbol{\mu}^{*,i})\right\|_{\ell_2}}{\left(\gamma^{\boldsymbol{e_y}\mathcal{R}}+\frac{\gamma^{\boldsymbol{e_y e_g}}L_g\gamma^{\boldsymbol{e}\mathcal{R}}}{1-L_g\gamma^{\boldsymbol{ee_g}}}\right)\|\mathcal{R}(\boldsymbol{\mu}^{*,i})\|_{\ell_2}},\tag{29}$$

where $\boldsymbol{e_y}$ is the actual error computed in the offline phase for a parameter set $\boldsymbol{\mu}^{*,i}$ selected at the i-th stage of the greedy algorithm in the offline phase. The denominator of (29) is motivated by (12). Then, $\bar{\rho}$ is defined as

$$\bar{\rho}=\max_i(\bar{\rho}^f_i).\tag{30}$$

In Section 5, the performance of the error estimate is investigated numerically. For the detailed algorithm of this error estimate, we refer to Abbasi et al. [2020], where it is limited to systems without distributed nonlinearities.

4.2 Empirical error estimate

The underlying idea for the empirical error estimate is similar to the idea used for finding the contributed error from EIM (Drohmann et al. [2012]) and the idea presented by Hain et al. [2019].

Statement 2. In the offline phase, we enrich RB functions from dimension N to dimension N' and the collateral basis functions from dimension M to dimension M' such that, based on the snapshots of previously selected parameters during the greedy algorithm, the following relation holds with $\eta^{N',M'}_{N,M}<1$:

$$\|y-\hat{y}_{N',M'}\|_{\ell_2}\leq\eta^{N',M'}_{N,M}\|y-\hat{y}_{N,M}\|_{\ell_2},\tag{31}$$

where y is the actual output computed from (4) and $\hat{y}_{N,M}$ is obtained from (9) with N RB functions and M collateral basis functions. An output error estimate can be defined as

$$\|y-\hat{y}_{N,M}\|_{\ell_2}\leq\frac{\zeta^{N',M'}_{N,M}}{1-\eta^{N',M'}_{N,M}},\tag{32}$$

with

$$\zeta^{N',M'}_{N,M}=\|\hat{y}_{N',M'}-\hat{y}_{N,M}\|_{\ell_2}.\tag{33}$$

Derivation: To increase the accuracy in the offline phase, based on the snapshots of the current selected parameter $\boldsymbol{\mu}^{*,i}$ in the i-th iteration of the greedy algorithm, we enrich Φ and q_{nl} step by step. During the greedy algorithm, we increase N' and M' until $\eta^{N',M'}_{N,M}$ in (31) becomes smaller than 1 for all parameters whose corresponding full-solution is available. Therefore, for any (N,M), we can find (N',M') such that $\eta^{N',M'}_{N,M}<1$. This condition bears similarities with the small-gain condition introduced in the first error estimate in this paper. Now, in the offline phase, corresponding to each (N,M), a pair of (N',M') and the value of $\eta^{N',M'}_{N,M}$ are known.

In the online phase, two reduced solutions with (N,M) and (N',M') basis functions should be solved. After obtaining these two computationally cheap solutions, we set

Algorithm 3 Empirical error estimate

```
Input: $q_{nl}, \Phi, \boldsymbol{X_m}$, parameters selected in the previous
       greedy iteration $\boldsymbol{\mu}^*$ and their corresponding full
       solutions
Output: $N', M', \eta_{N,M}^{N',M'}$
1: Set $N' = N$ and $M' = M$,
2: Based on the recently selected parameters, enrich
   $\Phi(N' = N' + 1)$ and $q_{nl}, \boldsymbol{X_m}(M' = M' + 1)$
3: compute $\eta^{N',M'} = \|y - \hat{y}_{N',M'}\|_{\ell_2}$ and $\eta_{N,M} =$
   $\|y - \hat{y}_{N,M}\|_{\ell_2}$ for all members of $\boldsymbol{\mu}^*$,
4: Set $\eta = \max\left(\frac{\eta^{N',M'}}{\eta_{N,M}}\right)$
5: if $\eta < 1$ then
6:     $\eta_{N,M}^{N',M'} = \eta$
7: else
       Go back to step 2
8: end if
```

Table 1. Test case parameter range for Burgers' equation.

parameter	L [m]	μ_1	μ_2
minimum	100	4	6
maximum	110	5	7
Online $\boldsymbol{\mu}^o$	105	4.5	6.5

$$\zeta_{N,M}^{N',M'} = \|\hat{y}_{N',M'} - \hat{y}_{N,M}\|_{\ell_2}. \tag{34}$$

Then, based on the following inequality

$$\|y - \hat{y}_{N,M}\|_{\ell_2} \leq \|y - \hat{y}_{N',M'}\|_{\ell_2} + \|\hat{y}_{N',M'} - \hat{y}_{N,M}\|_{\ell_2}, \tag{35}$$

and taking into consideration from the offline phase that $\|y - \hat{y}_{N',M'}\|_{\ell_2} \leq \eta_{N,M}^{N',M'} \|y - \hat{y}_{N,M}\|_{\ell_2}$, we finally obtain

$$\|y - \hat{y}_{N,M}\|_{\ell_2} \leq \frac{\zeta_{N,M}^{N',M'}}{1 - \eta_{N,M}^{N',M'}}. \tag{36}$$

The reason for having $\eta_{N,M}^{N',M'} < 1$ shows itself here to have finite and positive error estimate. ■

For the implementation of this error estimate, refer to Algorithm 3.

5 Numerical results

The simulation parameters in the online phase $\boldsymbol{\mu}^o$ for Burgers' equation along with the parameter domain are listed in Table 1, where the minimum and maximum value for each parameter are specified. The discrete parameter domain is composed of 8 equidistant members in the parameter domain. In the last row of Table 1, the set of parameters selected for the online simulation $\boldsymbol{\mu}^o$ is reported, which does not lie in the discrete parameter domain. This kind of parameter setting ensures that $4 \leq u(t, x; \boldsymbol{\mu}) \leq 7$ for all $(t, x) \in [0, T] \times [0, L]$. The number of spatial grid cells is $\mathcal{N} = 250$, the time horizon T is 50 s and time step is $\Delta t = 0.01$ s. The output is the value of the conservative variable at $x = L$.

The effect of using the actual error, the error estimates based on the ℓ_2-gain notion (with and without the reduction factor $\bar{\rho}$ in (30)) and the empirical error estimate in the greedy algorithm of PODEI algorithm (Algorithm 2) as in Drohmann et al. [2012] is shown in Figure 4. Clearly, the error estimates accurately approximate the maximum error in the parameter domain. The "estimation before $\bar{\rho}$" is the most conservative error estimate, which is due the high conservativeness in the ℓ_2-gain notion. In the eighth iteration, using the ℓ_2-based error estimate, the collateral basis function is inconsistent with the RB basis functions, which reduces the accuracy of the RB solution and increases the residual values. Discarding the collateral basis functions, as in PODEI algorithm, resolves the problem for the next iteration. In general, the accuracy of the RB solution increases by enriching the RB and collateral basis functions.

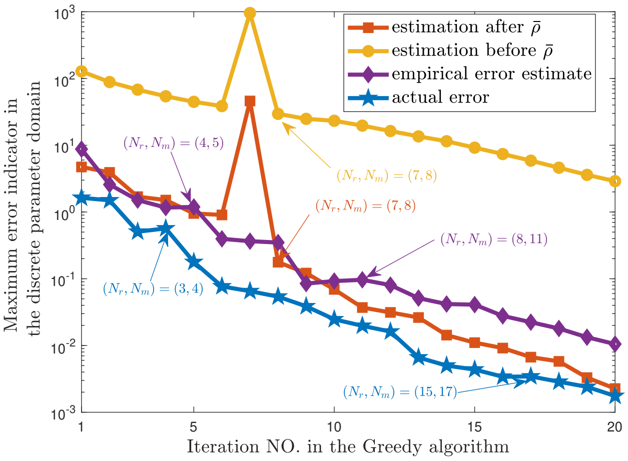

Fig. 4. Maximum error in the discrete parameter domain during the greedy algorithm.

In the online phase by using 20 RB functions and 20 collateral basis functions, the time-wise evolution of the solution is shown in Figure 5 at four different time instants in comparison with the FV solution. The speedup factor is reported in Table 2 (without including the error estimate computational time). The moderate speedup is due to the hyperbolic, nonlinear and 1D nature of the original problem. The effect of the number of RB functions in the induced error due to the reduction for the parameters used in the online phase is shown in Figure 6.

The results of this section verify that both error estimates perform successfully in estimating the maximum error during the greedy algorithm in the offline phase and also estimating the error for a new parameter setting during the online phase. However, the estimate based on the ℓ_2-gain notion suffers from restricted applicability to satisfy the small-gain condition. This becomes even more restricted in the case of stronger nonlinearities (nonlinearities with higher local Lipschitz constant). On the other hand, in the empirical error estimate, we only need to find (N', M') to be sufficiently large to satisfy the condition on $\eta_{N,M}^{N',M'} < 1$. Apart from this condition that should be resolved in the offline phase, there is no restriction on the applicability of the method in the online phase.

Table 2. Speedup factors for the reduced basis method for Burgers' equation.

$N = M$	1	5	10	15	20
Speedup	17.4	4.2	4	3.4	3

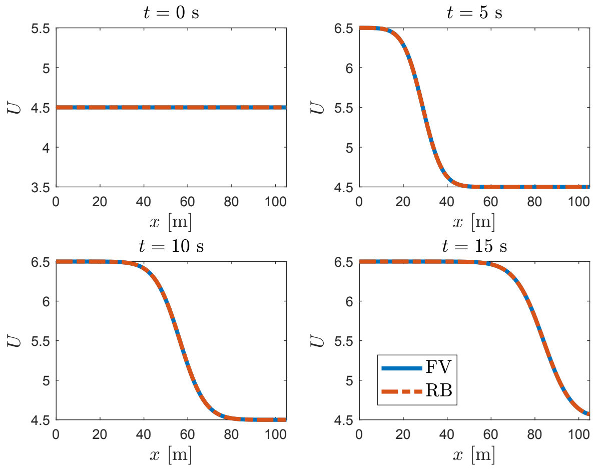

Fig. 5. Comparison of the full-order and low-order solutions over time using 20 RB functions and 20 collateral basis functions.

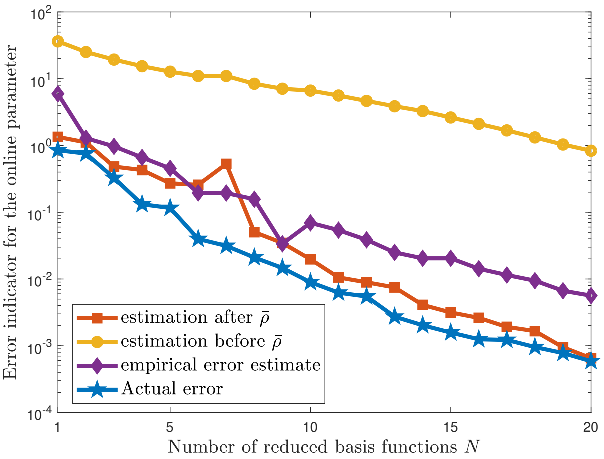

Fig. 6. Error evolution by increasing the number of basis functions.

6 Conclusion

In this paper, a new perspective on the interaction between EIM and RB methods is introduced. First, a new error estimate based on a Lur'e type formulation of nonlinear Burgers' equation is defined. This estimate is rigorous, accurate and effective, but has limited applicability due to satisfying a small-gain condition. Furthermore, it requires another reduced-order model to be solved to approximate the residual. To circumvent the small-gain condition issue, hinged on the snapshots generated in the offline phase, an empirical error estimate is introduced that does not suffer from the restrictions of the first error estimate. Both error estimates work efficiently in terms of computational effort and accuracy. The empirical error estimate is faster and also applicable on a wider range of problems than the error estimate proposed on the basis of ℓ_2-gain notion.

References

Abbasi, M.H., Iapichino, L., Besselink, B., Schilders, W., and van de Wouw, N. (2020). Error estimation in reduced basis method for systems with time-varying and nonlinear boundary conditions. *Computer Methods in Applied Mechanics and Engineering*, 360, 112688.

Barrault, M., Maday, Y., Nguyen, N.C., and Patera, A.T. (2004). An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathematique*, 339(9), 667–672.

Besselink, B., van de Wouw, N., and Nijmeijer, H. (2012). Model reduction for a class of convergent nonlinear systems. *IEEE Transactions on Automatic Control*, 57(4), 1071–1076.

Besselink, B., van de Wouw, N., and Nijmeijer, H. (2013). Model reduction for nonlinear systems with incremental gain or passivity properties. *Automatica*, 49, 861–872.

Chaturantabut, S. and Sorensen, D. (2010). Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing*, 32(5), 2737–2764.

Drohmann, M., Haasdonk, B., and Ohlberger, M. (2012). Reduced Basis Approximation for Nonlinear Parametrized Evolution Equations based on Empirical Operator Interpolation. *SIAM Journal on Scientific Computing*, 34(2), A937–A969.

Friedrichs, K.O. (1954). Symmetric hyperbolic linear differential equations. *Communications on Pure and Applied Mathematics*, 7(2), 345–392.

Haasdonk, B. and Ohlberger, M. (2008). Reduced basis method for finite volume approximations of parametrized linear evolution equations. *ESAIM Mathematical Modelling and Numerical Analysis*, 42(2), 277–302.

Hain, S., Ohlberger, M., Radic, M., and Urban, K. (2019). A hierarchical a posteriori error estimator for the reduced basis method. *Advances in Computational Mathematics*, 1–24.

Hesthaven, J.S., Rozza, G., and Stamm, B. (2016). *Certified reduced basis methods for parametrized partial differential equations*. SpringerBriefs in Mathematics. Springer International Publishing.

Khalil, H.K. (2001). *Nonlinear systems*. Pearson, Upper Saddle River, N.J, 3 edition edition.

Lax, P.D. (1954). Weak solutions of nonlinear hyperbolic equations and their numerical computation. *Communications on Pure and Applied Mathematics*, 7(1), 159–193.

Naderi Lordejani, S., Besselink, B., Abbasi, M.H., Kaasa, G.O., Schilders, W.H.A., and van de Wouw, N. (2018). Model order reduction for managed pressure drilling systems based on a model with local nonlinearities. *IFAC-PapersOnLine*, 51(8), 50–55.

Orlandi, P. (2000). The Burgers equation. In *Fluid Flow Phenomena: A Numerical Toolkit*, Fluid Mechanics and Its Applications, 40–50. Springer Netherlands, Dordrecht.

Zhang, Y., Feng, L., Li, S., and Benner, P. (2015). An efficient output error estimation for model order reduction of parametrized evolution equations. *SIAM Journal on Scientific Computing*, 37(6), B910–B936.