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 role 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 first 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 POPEI algorithm by Drohmann et al. [2012]. The inaccurate approximation of the nonlinear functions also plays a role...
in the final error induced by reduction, which has to be taken into account when building error estimates. To generate both collated 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] from systems with local nonlinearities circumvents this issue by using the \( \ell_2 \)-norm (which also works well if applied to systems with local nonlinearities) acting as the input into the system defined according to

\[
\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}(f(u)) = 0, \quad t \in [0, T], \quad x \in [0, L],
\]

where \( u := u(t, x; \mu) \) 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, \( \mu \in \mathcal{D} \) is a vector of parameters used in (1) that varies in a multi-query analysis within the parameter domain \( \mathcal{D} \subseteq \mathbb{R}^R \), 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

\[
u(0, x; \mu) = \mu_1, \quad \text{constant over the spatial domain.}
\]

For the boundary condition at \( x = 0 \), we assume

\[
u(0, 0; \mu) = \begin{cases} \mu_1, & t = 0, \\ \mu_2, & t > 0. \end{cases}
\]

Therefore, in this study, we have \( \mu = [\mu_1, \mu_2] \).

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

\[
U^{n+1} = L_{in} U^n + BU_0^n - \frac{\Delta t}{4\Delta x} L_{nl}(U^n)^2 + \frac{\Delta t}{2\Delta x} B(U_0^n)^2,
\]

where \( U^n := [U^n_0, \ldots, U^n_N]^T \in \mathbb{R}^N \) is the vector containing \( U^n_0 \), the average of the conservative variable \( u \) over the i-th grid cell at the time instant \( t^n := n\Delta t, n = 0, \ldots, N_t \) with \( N_t \) number of time steps. Here, \( \Delta t \) and \( \Delta x \) refer to the temporal and spatial discretization intervals over time and space, respectively. The spatial discretization consists of cells \( \{x_{i-1/2}, x_{i+1/2}, i = 1, \ldots, N_t\} \) with the length of \( \Delta x \) centered at \( x_i = x_{i-1/2} + \Delta x/2 \) and \( N_t \) spatial grid cells. Furthermore, \( L_{in}, L_{nl} \in \mathbb{R}^{N \times N} \) are the operators acting on the linear and nonlinear part of the system that emerge after applying the full-discretization. Also, \( U_0^n \in \mathbb{R}^r \) is the value of the conservative variable at the boundary \( x = 0 \) acting as the input into the system defined according to (2) and \( B \in \mathbb{R}^{N \times N} \) 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 \( \Sigma \) depicted in the left side of Figure 1, which comprises a linear subsystem \( \Sigma_{lin} \) and nonlinear subsystem \( \Sigma_{nl} \) given by

\[
\Sigma_{lin} : \begin{cases}
U^{n+1}_i = L_{lin} U^n_i + BU_0^n - \frac{\Delta t}{4\Delta x} L_{nl} U^n_i \\
\frac{\partial}{\partial x}(U^n_i), \quad g = C_p U^n_i, \\
z_n = U^n_i
\end{cases}
\]

\[
\Sigma_{nl} : \begin{cases}
U_{nl}^{n+1} = g(z_n) = (z_n)^2.
\end{cases}
\]

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 \( C_p \in \mathbb{R}^{w \times N} \) is the corresponding output matrix. This full-order model has large dimension (i.e., \( N \) 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 \( \mu \in \mathcal{D} \), i.e., \( \Sigma_{lin} \) in (4) is internally asymptotically stable.
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

\[
\hat{U}^n(\mu) = U_0^n(\mu) 1 + \Phi a^n,
\]

where \(\hat{U}^n \in \mathbb{R}^N\) is the solution of the reduced-order model, and \(1 \in \mathbb{R}^N\) is a vector of ones that enables the RB solution \(\hat{U}^n(\mu)\) to satisfy the boundary condition (2) at all time instants. Then, the RB functions \(\Phi \in \mathbb{R}^{N \times N}\), 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 \(\Phi\) vanishing at the location of the specified boundary, we modify the snapshots during the greedy algorithm for a selected parameter \(\mu^*\) and then apply the Proper Orthogonal Decomposition (POD) (Hesthaven et al. [2016]) on these modified snapshots (see Algorithm 1), defined as

\[
\begin{align*}
\hat{U}^{n,*}(\mu^*) &= \hat{U}^n(\mu^*) - U_0^n(\mu^*) 1, \\
\hat{U}^{*}(\mu^*) &= \{\hat{U}^{n,*}(\mu^*)\}, \quad \forall n = \{0, \cdots, N_1\}.
\end{align*}
\]

Finally, “POD(\(\hat{U}(\mu^*), 1\)”, 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 pre-selected 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

\[
(\hat{U}^n)^2 \approx q_{nl} \theta_{nl},
\]

where \(q_{nl} \in \mathbb{R}^{N \times M}\) 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 \(\theta_{nl}\) in (7) are obtained during the online phase such that the interpolation is exact at \(M\) pre-selected points \(X_m = \{x_1, \cdots, x_M\}\) 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 \(P = [e_{x_1}, \cdots, e_{x_M}] \in \mathbb{R}^{N \times M}\) where \(e_i\) is the \(i\)-th column of the identity matrix (of dimension \(N \times N\)). For the points \(X_m\), we have

\[
(P^T \hat{U}^n)^2 = P^T q_{nl} \theta_{nl}^T,
\]

stating that the interpolation is exact at \(X_m\) if \(P^T q_{nl}\) is non-singular (\(\theta_{nl}\) 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 \(\Phi\). 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 \(\Phi\), we obtain the reduced-order model

\begin{algorithm}
\caption{POD algorithm, POD\((U, n_{POD})\)}
\begin{algorithmic}
\State **Input:** Snapshots \(U(\mu) \in \mathbb{R}^{N \times n_{POD}}\)\n\State **Output:** \(\phi \in \mathbb{R}^{N \times n_{POD}}\), number of basis vectors \(n_{POD}\)
\begin{enumerate}
\item Perform a Singular Value Decomposition on the snapshots, \(U = U_{SVD}SV\)
\item \(\phi = U_{SVD}(1 : n_{POD})\) is the first \(n_{POD}\) vectors of the left singular vectors \(U_{SVD}\).
\end{enumerate}
\end{algorithmic}
\end{algorithm}
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 $\ell_2$-gain notion. For the second one, we use the solutions of the full-order model generated in the offline phase to obtain an empirical error estimate.

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 $\ell_2$-gain notion. For the second one, we use the solutions of the full-order model generated in the offline phase to obtain an empirical error estimate.

Algorithm 2 PODEI-Greedy algorithm

Input: $D_h(d$iscretized version of $D), N, \mu, \mu^k \in D_h$.

Output: $\Phi, q_{nl}, X_m, B_{nl}$.

1: Set $\Phi = \{\}, q_{nl} = \{\}, X_m = \{\}, B_{nl} = 1$.

2: for $k = 1$ to $N - 1$ do

3: Solve (4) for $\mu^k$ to obtain $U = [U^0, \ldots, U^{N_t}]$ and $U_{nl} = [U_{nl}^0, \ldots, U_{nl}^{N_t}]$.

4: Generate $U^* = U - [U^0, \ldots, U_0^{N_t}]$ and $U_{nl} = U_{nl} - q_{nl}B_{nl}^{-1}U_{nl}(X_m, :)$.

5: Set $U = U^* - \Phi \Phi^T U^*.$

6: $\Phi \leftarrow \text{orth} [\Phi \cup \text{POD}(U, 1)]$, $q_{POD} = \text{POD}(U_{nl}, 1)$.

7: $\sigma_M = (q_{nl}(X_m, :))^{-1}q_{POD}(X_m)$.

8: $r_{q} = q_{POD} - q_{nl}\sigma_M,$ $X_m \leftarrow \{X_m \cup (r_{q})\}.$

9: $\Phi$ and $q_{nl}$ perform the error estimates to find the worst approximated solution and find $\mu^{k+1}$ and $e(\mu^{k+1}).$

12: if $e(\mu^{k+1}) > e(\mu^k)$ then

13: $q_{nl} = q_{nl}(1 : \text{end - 1}), X_m = X_m(1 : \text{end - 1}), B_{nl} = q_{nl}(X_m, :)$.

14: end if

15: end for

4.1 Error estimate based on the $\ell_2$-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. (Drohnmann 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 $\theta_{nl}^*$ and the corresponding coefficients by $\theta_{nl}^*$, such that $\Sigma_{nl}^g(\mu)^2 = \theta_{nl}^* \theta_{nl}^* (\mu) \forall n = 1, \ldots, N_t$ and $\mu \in D.$

Statement 1. Let $U^n_0$ be obtained from (4) and $U^n_r$ be obtained from (9) and (5) with $n = 1, \ldots, N_t$ under the same initial condition and the same boundary input $U_0^n.$ We define the residual $R^n$ by inserting the RB solution $U^n_r$ into (4) as follows:

$$R^n = U^n_r - \left( \dot{L}_{lin} \dot{U}^n_r + B U^n_r - \frac{\Delta t}{4\Delta x} L_{nl}g(U^n_r) \right) + \frac{\Delta t}{2\Delta x} B(U_0^n)^2. \tag{11}$$

We assume the Lipschitz continuity $L_g$ for the nonlinear function $e^n_\text{y} = g(e^n), i.e., \|e^n_\text{y}\| \leq L_g \|e^n\|$. An estimate of the error bound of $\|e_\text{y}\|_{\ell_2}$ with $e_\text{y} := y - \hat{y}$ is given by $\|e_\text{y}\|_{\ell_2} \leq \kappa(\mu)\|R\|_{\ell_2}$ with $\kappa(\mu) = \frac{\gamma e^n R + \gamma e^n L^2 g \gamma e^n R}{1 - Lg e^n R}$, where $\|R\|_{\ell_2} := \left( \sum_{n=0}^{\infty} \|R\|^2 \right)^{\frac{1}{2}}$ and $\gamma e^n$ denoting the $\ell_2$-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)

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

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

$\Sigma^e_{lin}$:

$$\begin{cases} e^{n+1} = L_{lin}e^n + \frac{\Delta t}{4\Delta x} L_{nl}e^n_\text{y} - R^n, \\ e^n_\text{y} = C_y e^n, \\ e^n = e^n, \end{cases}$$

$\Sigma^e_{nl}$:

$$\begin{cases} e^{n+1} = f(U, e^n) = g(e^n + \hat{U}) - g(U), \end{cases}$$

This feedback interconnection is depicted in Figure 2. Notably, the relation in $\Sigma^e_{nl}$ 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.
In this section, we introduce two types of error estimates. The error estimate introduced.

\[
\| e \|_{\ell_2} \leq \gamma e_{\mathcal{R}} \| \mathcal{R} \|_{\ell_2} + \gamma e_{\mathcal{E}} \| e_g \|_{\ell_2}.
\]

This \( \ell_2 \)-norm is equal to the \( \mathcal{H}_2 \)-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 \( \| e_g^n \| \) 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 follows:

\[
\mathcal{R}^n = \mathcal{R}^n_{\text{lin}} + \mathcal{R}^n_{\text{nl}},
\]

where

\[
\mathcal{R}^n_{\text{lin}} = \hat{U}^{n+1} - \left( L_{\text{lin}} \hat{U}^n + B U_0^n \right) - \frac{\Delta t}{4 \Delta x} L_{\text{nl}} q_{n_l} e_{\mathcal{E}}^n + \frac{\Delta t}{2 \Delta x} B(U_0^n)^2,
\]

\[
\mathcal{R}^n_{\text{nl}} = - \frac{\Delta t}{4 \Delta x} L_{\text{nl}}(q_{n_l} e_{\mathcal{E}}^n - \hat{U}^n)^2.
\]

In computing the two-norm of the residual \( \mathcal{R}^n \), it is necessary to compute \( \mathcal{R}^n_{\text{nl}} \), which is time-consuming due to the presence of the nonlinear term \( (\hat{U}^n)^2 \). To avoid this computational issue, following the idea presented by Drohmann et al. [2012], this term is calculated empirically using Remark 1 instead of \( \mathcal{R}^n_{\text{nl}} \).

\[
\mathcal{R}^n_{\text{nl}} = - \frac{\Delta t}{4 \Delta x} L_{\text{nl}}(q_{n_l} e_{\mathcal{E}}^n - \hat{U}^n)^2.
\]

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

\[
\| e_g^n \| \leq L_g \| e^n \|,
\]

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

\[
\| e_g \|_{\ell_2} \leq L_g \| e \|_{\ell_2}.
\]

Similar to (15), we have

\[
\| e \|_{\ell_2} \leq \gamma e_{\mathcal{R}} \| \mathcal{R} \|_{\ell_2} + \gamma e_{\mathcal{E}} \| e_g \|_{\ell_2}.
\]
dependence of the conditions should be chosen in a way that the satisfaction of the small-gain condition, for all members of the parameter set $\mu$, holds 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 state-space must be considered. We terminate the minimization problem as soon as the constraints are satisfied.

$$\min(\mu_1, \mu_2) \leq U^\prime(\mu^i) \leq \max(\mu_1, \mu_2).$$

(26)

The constraints in the minimization problem (25) ensure that for each parameter setting, first, the linear part of the error dynamics $\Sigma^{\ell}_{ini}$ is stable, and second, the interconnection of the linear subsystem $\Sigma^{lin}_{ini}$ and the nonlinear subsystem $\Sigma^{nl}_{ini}$ 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 state-space must be considered. We terminate the minimization problem as soon as the constraints are satisfied.

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 contribution 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 $\eta_{N,M} < 1$:

$$\|y - \hat{y}_{N',M'}\|_{l_2} \leq \eta_{N,N'}^{N',M'} \|y - \hat{y}_{N,M}\|_{l_2},$$

(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

$$\|y - \hat{y}_{N,M}\|_{l_2} \leq \zeta_{N,N'}^{N',M'} \frac{\gamma_{N,N'}^{N',M'}}{1 - \eta_{N,N'}},$$

(32)

with

$$\zeta_{N,N'}^{N',M'} = \|\hat{y}_{N',M'} - \hat{y}_{N,M}\|_{l_2}.$$  

(33)

**Derivation:** To increase the accuracy in the offline phase, based on the snapshots of the current selected parameter $\mu^{\ast,i}$ in the $i$-th iteration of the greedy algorithm, we enrich $\Phi$ 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

$$\kappa^{\prime} = \rho \kappa(\mu),$$

(28)

where $\kappa^{\prime}$ is an estimate of the error gain $\kappa$ in (12), which is calculated based on $\kappa$ in (12). To define $\rho$, we first introduce the variable $\rho^i$ as a measure of the conservatism

$$\rho^{i} = \frac{\|e_y(\mu^{i})\|_{l_2}}{\gamma e_y R + \gamma e_y R L e_y \eta_{N,M}^{N',M'} \|R(\mu^{i})\|_{l_2}},$$

(29)

In the offline phase, the error estimate is investigated numerically.
Algorithm 3 Empirical error estimate

Input: \( \eta_{nl}, \Phi, X_m \), parameters selected in the previous greedy iteration \( \mu^* \) and their corresponding full solutions

Output: \( N', M', \eta_{N',M'} \)

1: Set \( N' = N \) and \( M' = M \).
2: Based on the recently selected parameters, enrich \( \Phi(N' + 1) \) and \( \eta_{nl}, X_m(M' + 1) \)
3: Compute \( \eta_{N',M'} \)
4: Set \( \eta = \max (\eta_{N',M'}) \)
5: If \( \eta < 1 \) then
6: \( \eta_{N',M'} = \eta \)
7: Else
8: Go back to step 2
9: End if

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

<table>
<thead>
<tr>
<th>parameter</th>
<th>( L ) [m]</th>
<th>( \mu_1 )</th>
<th>( \mu_2 )</th>
<th>( \mu^* )</th>
</tr>
</thead>
<tbody>
<tr>
<td>minimum</td>
<td>100</td>
<td>4</td>
<td>6</td>
<td>100</td>
</tr>
<tr>
<td>maximum</td>
<td>110</td>
<td>5</td>
<td>7</td>
<td>110</td>
</tr>
<tr>
<td>Online</td>
<td></td>
<td></td>
<td></td>
<td>105</td>
</tr>
</tbody>
</table>

Let \( \eta_{N',M'} = \| \hat{y}_{N',M'} - \hat{y}_{N,M} \|_{\ell_2} \), (34)

Then, based on the following inequality

\[
\| y - \hat{y}_{N,M} \|_{\ell_2} \leq \| y - \hat{y}_{N',M'} \|_{\ell_2} + \| \hat{y}_{N',M'} - \hat{y}_{N,M} \|_{\ell_2},
\]

and taking into consideration from the offline phase that

\[
\| y - \hat{y}_{N',M'} \|_{\ell_2} \leq \eta_{N,M} \| y - \hat{y}_{N,M} \|_{\ell_2},
\]

we finally obtain

\[
\| y - \hat{y}_{N,M} \|_{\ell_2} \leq \eta_{N,M} \| \hat{y}_{N',M'} - \hat{y}_{N,M} \|_{\ell_2} \cdot \frac{1}{\eta_{N',M'}}.
\]

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

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

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

<table>
<thead>
<tr>
<th>( N = M )</th>
<th>1</th>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speedup</td>
<td>17.4</td>
<td>4.2</td>
<td>4</td>
<td>3.4</td>
<td>3</td>
</tr>
</tbody>
</table>
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, 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 ℓ₂-gain notion.

References


