Error estimates for model order reduction of Burgers' equation

Citation for published version (APA):
Abbasi, M. H., Iapichino, L., Besselink, B., Schilders, W., & van de Wouw, N. (2020). Error estimates for model order reduction of Burgers' equation. IFAC-PapersOnLine, 53(2), 5609-5616. https://doi.org/10.1016/j.ifacol.2020.12.1575

DOI:
10.1016/j.ifacol.2020.12.1575

Document status and date:
Published: 01/01/2020

Document Version:
Accepted manuscript including changes made at the peer-review stage

Please check the document version of this publication:
- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:
www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:
openaccess@tue.nl
providing details and we will investigate your claim.

Download date: 26. Oct. 2023
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 many newly developed model order reduction techniques and error estimates. Model order reduction of the parameterized Burgers’ equation is commonly done by the reduced basis method. In this method, an error estimate plays a crucial role in accelerating the offline phase and also 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 the 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, basis functions for approximating the solution are generated. This phase contains computations scaled 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 basis functions. The computations in this phase scale with the number of basis functions generated in the offline phase, which renders 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. [2019]). Second, it also certifies the accuracy of the solution that is obtained during the online phase. Therefore, developing a sharp error bound (or accurate estimate) is crucial within this approach.

To build an efficient yet accurate reduced-order model by the RB method and decompose 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 other 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 circumvent large computational times, the EIM/DEIM 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 (Drohmann et al. [2012], Benaceur et al. [2018]); e.g. the PODEI algorithm by Drohmann et al. [2012]. The inaccurate approximation
of the nonlinear functions also play a role 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, the Burgers’ equation. In general, hyperbolic systems are commonly solved by Finite-Volume (FV) techniques that lead to a 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. To mention a few works done in this regard, we refer to Haasdonk and Ohlberger [2008], Zhang et al. [2015], Abbasi et al. [2019]. These methods 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. [2019] utilize the norm of the state vector (estimates) are not valid and grow exponentially over time. The method introduced by Abbasi et al. [2019] (which also works well if applied to systems with local nonlinearities) circumvents this issue by using the \( \ell_2 \)-norm of the system with respect to its inputs and outputs, as similarly done in the balanced truncation method by Lordejani et al. [2018], Bessink 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 by Abbasi et al. [2019] from systems with local nonlinearities to systems with distributed nonlinearities. However, the error estimate of Abbasi et al. [2019] 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 \( \ell_2 \)-gain notion, an empirical error estimate is also introduced in this paper. This estimate is based on the snapshots generated in the off-line 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 the Burgers’ equation, which is sometimes referred to as the scalar version of the Navier-Stokes equations (Orlandi [2000]). This equation is defined as

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

where \( u := u(t, x; \mu) \) is the conservative variable of the system and \( f(u) = u^2/2 \) is the flux function associated with the 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 the multi-query analysis within the parameter domain \( \mathcal{D} \in \mathbb{R}^R \) with \( R > 0 \) the number of varying parameters. We assume that the initial condition and the boundary condition are represented by the varying parameters. For the initial condition, we assume \( u(0, x; \mu) = \mu_1 \), which is constant over the spatial domain. For the boundary condition at \( x = 0 \), we assume

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

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

In the CFD community, many highly nonlinear FV schemes are developed to solve this equation accurately (Kurganov and Tadmor [2000]). These solvers introduce more nonlinearities in the discretized system dynamics and render the system more difficult to reduce. To avoid further nonlinearities (such as the “max” operator), a simple linear scheme, namely the Lax-Friedrichs scheme (see Lax [1954], Friedrichs [1954]), is used to discretize the system dynamics of (1). Let \( \Delta t \) and \( \Delta x \) refer to the temporal and spatial discretization intervals over time and space, respectively. The spatial discretization consists of cells \( (x_{i-1/2}, x_{i+1/2}), i = 1, \ldots, N \), with the length of \( \Delta x \) centered at \( x_i = x_{i-1/2} + \Delta x/2 \) and \( N \) number of spatial grid cells. Time discretization is performed using a forward Euler integration method. The Lax-Friedrichs scheme approximates (1) with

\[
U_{i+1}^n = U_i^n - \frac{\Delta t}{\Delta x} \left( F(U_i^n, U_{i+1}^n) - F(U_{i-1}^n, U_i^n) \right),
\]

where

\[
F(U_i^n, U_{i+1}^n) = \frac{f(U_{i+1}^n) + f(U_i^n)}{2} - \frac{\Delta t}{2\Delta x} (U_{i+1}^n - U_i^n).
\]

Finally, Section 6 concludes the paper.
Here, \( y \in \mathbb{R}^w \) is the output of interest of the system and \( w \) is the dimension of the output (for instance, \( y \) can be the value of the conservative variable at the right-end of the spatial domain with \( w = 1 \)) and \( C_y \in \mathbb{R}^{w \times N} \) 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 (6), which is the topic of the next section. Before going to the reduction part, an assumption necessary for the computations of the error estimate based on the \( \ell_2 \)-gain notion is mentioned below.

Assumption 1. The system matrix \( L_{lin} \) is Schur for all \( \mu \in \mathcal{D} \), i.e., \( \Sigma_{lin} \) in (6) is internally asymptotically stable.

3 Model reduction approach

In this section, the requirements for the model reduction approach are explained. The RB method for the reduction of the linear subsystem is illustrated while (D)EIM is used to reduce the computational effort associated to the nonlinear part of the dynamics. Finally, the combination of both methods for the hyper-reduction of the nonlinear system is explained.

3.1 Reduced basis method

A powerful method for dimension reduction of a parameter-dependent dynamical system is the RB method. The RB method constructs a reduced-order model on the basis of solutions of the full-order model for specific members of the parameter domain.

As discussed by Abbasi et al. [2019], handling time-varying boundary conditions within the RB method is vital as the (time-varying) control inputs commonly act at the boundaries. Therefore, the method introduced by Abbasi et al. [2019] is followed here to incorporate the boundary conditions. Specifically for our case study, in this section. Before going to the reduction part, an assumption necessary for the computations of the error estimate based on the \( \ell_2 \)-gain notion is mentioned below.

3.2 Empirical interpolation method

As the modified snapshot vanishes at the boundary for all time instants, “POD(\( \hat{U}^n(\mu^*) \))” obtained from Algorithm 1 gives a basis function which also vanishes at the boundary. For more details, we refer to Abbasi et al. [2019].

After applying EIM, the nonlinear function in (6) is approximated by a linear interpolation as follows:

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

where \( q_{nl} \in \mathbb{R}^{N_{nl}} \) is the matrix of collateral basis functions and \( \theta_{nl} \in \mathbb{R}^{N_{nl}} \) are the unknown coefficients of the collateral basis functions, to be calculated online. The collateral basis functions \( \theta_{nl} \) are obtained during the online 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 locations of all interpolation points concatenated together are denoted by \( X_m \in \mathbb{R}^{N_{nl}} \) where \( N_{nl} \) is the number of interpolation points. The effect of the nonlinear function is then fed back into the linear system via the feedback interconnection as shown in Figure 2.

The schematic representation of the 1: Perform a Singular Value Decomposition on the snapshots, \( U(n) \mu \mathbb{N}^{N_{nl}} \times 1 \), number of basis vectors \( n_{\text{POD}} \)

Output: \( \phi \in \mathbb{R}^{N_{nl} \times n_{\text{POD}}}, \) the first \( n_{\text{POD}} \) vectors of the left singular vectors \( U_{EVD}. \)

**Algorithm 1** POD algorithm, POD(\( U/n_{\text{POD}} \))

Input: Snapshots \( U(\mu) \in \mathbb{R}^{N_{nl} \times N_1} \), number of basis vectors \( n_{\text{POD}} \)

1. Perform a Singular Value Decomposition on the snapshots, \( U = U_{EVD} SV \)
2. \( \phi = U_{EVD}(1 \ldots n_{\text{POD}}) \) is the first \( n_{\text{POD}} \) vectors of the left singular vectors \( U_{EVD} \).
i-th iteration (the selection procedure of such points is introduced later in Algorithm 2). Specifically, let $P = [e_{x_1}, \ldots, e_{x_m}] \in \mathbb{R}^{N \times N_m}$ where $e_i$ is the i-th column of the identity matrix (of dimension $N \times N$). For the points $X_m$, we have
\[(PTU)^m = PTq_m^\theta n_m, \tag{10}\]

stating that the interpolation is exact at $X_m$ if $PTq_m$ is non-singular ($\theta n_m$ can then be computed from (10)).

After approximating the nonlinearities in (6) with linear interpolation of the collateral basis functions, we can apply 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 (7) and the EIM approximation (9) in (6), applying a Galerkin projection of the resulting nonlinearities on the resulting subspace spanned by $\Phi$. We obtain the following reduced-order model approximation:

\[\begin{align*}
\Sigma_{lin} : & \quad q_{n+1} = L_{lin}q_n + (B + L_{BC})U_0^n - \frac{\Delta t}{\Delta x} L_{nl}^\theta n_l,
\sigma^g = U_0^n(C_q 1) + (C_q \Phi)a^n, \\
\Sigma_{nl} : & \quad U_m^n = g(z_n) = (z_n)^2, \tag{11}
\theta n_l = (PTq_m)^{-1}U_{nl}^n,
\end{align*}\]

where $L_{lin} = \Phi^T L_{lin} \Phi \in \mathbb{R}^{N \times N}$, $L_{nl} = \Phi^T L_{nl} q_{nl} \in \mathbb{R}^{N \times N_m}$, $B = \Phi^T B \in \mathbb{R}^{N}$, $L_{BC} = \Phi^T L_{lin} 1 \in \mathbb{R}^{N}$, and $\Phi_1 = \Phi^T 1 \in \mathbb{R}^N$. Moreover, $C_q 1, C_q \Phi \in \mathbb{R}^{N \times N}$, $PT \Phi \in \mathbb{R}^{N_m \times N}$. Finally, $z_n$ is the value of the reduced solution at the pre-selected points $X_m$. Recall that $N_r \ll N$ and $N_m \ll N$ are, respectively, the dimension of the basis functions and the collateral basis functions. None of the computations in (11) scales with the actual degrees of freedom $N$ and therefore the model is in a reduced form. It should be noted that the boundary-related terms in (11) (such as $U_0^n(PT1))$ are due to the ansatz used in (7) and the segregation of boundary condition from the solution. In this work, to synchronize the generation of the basis functions $\Phi$ 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 different types of error estimates. In the first one, we build the error dynamics and propose an estimate based on the $\ell_2$-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 $\ell_2$-gain notion

As shown in (11), the interconnection of the RB method and EIM can be represented as a Lur’e-type system as shown in Figure 2. 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 define the error estimate, the error dynamic is defined. To this end, inserting the RB solution $\hat{U}^n$ obtained from (11) and (7) into the full-order model (6) results in the residual $R^n$ as

$$R^n = U^n - \left( L_{lin} \hat{U}^n + BU_0^n - \frac{\Delta t}{\Delta x} L_{nl}^\theta n_l (\hat{U}^n)^2 \right) + \frac{\Delta t}{2\Delta x} L_{BC}(U_0^n)^2,$$

Then, subtracting (12) from the full-order model (5) leads to the error dynamics of the system given by

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

with $e := U - \hat{U}$. By denoting $(U^n)^2 - (\hat{U}^n)^2$ as $e_0$ and rewriting the dynamics in the feedback interconnected form, we obtain the error system $\Sigma_{lin}$ with its linear and nonlinear subsystems given as follows:

$$\Sigma_{lin} : \begin{cases} e^{n+1} = L_{lin} e^n + \frac{\Delta t}{\Delta x} L_{nl} e_0^n - R^n, \\
e_0^n = C_q 1 e_0^n, \\
e_0^n = e_0^n, \\
e_0^n = f(U, e_x) = g(e_x + U) - g(U) \end{cases}$$

This feedback interconnection is depicted in Figure 3. Notably, the relation in $\Sigma_{lin}$ 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 the online phase, however, we do not have access to the values for $(U^n)^2$ since the actual solution is not known. Therefore, an estimation of the output should be defined as we cannot simulate this error dynamics in a computationally efficient manner.

Following the same idea introduced by Abbasi et al. [2019] for linear systems and assuming $\Sigma_{lin}$ is asymptotically
stable (Assumption 1), an error bound on the $\ell_2$-norm of the error signal can be computed as follows:

$$
\|e_y\|_{\ell_2} \leq \gamma^{eR}\|R\|_{\ell_2} + \gamma^{e_u}\|e_y\|_{\ell_2},
$$

(15)

where $\|R\|_{\ell_2} := \sqrt{\sum_{n=0}^{\infty} \|R_n\|_2^2}$ (similarly for $\|e_y\|_{\ell_2}$ and $\|e_y\|_{\ell_2}$). Also, $\gamma^{e_u}$ represents the $\ell_2$-norm of the system from input $x$ to the output $y$. 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]), which is computed as described by Abbasi et al. [2019]. Apart from the gains, to compute this error bound, both $\|R\|_2$ and $\|e_y\|_2$ should be computed in a computationally efficient manner, which is explained next.

For computing the norm of the residual, we decompose the residual into a linear and a nonlinear part as below,

$$
\mathcal{R}_n = \mathcal{R}_n^{lin} + \mathcal{R}_n^{nl},
$$

(16)

where

$$
\begin{align*}
\mathcal{R}_n^{lin} &= U^{n+1} - \left( L_{lin}U^n + BU^n_0 - \frac{\Delta t}{4\Delta x} L_{nl}q_{nl}\theta_{nl}^n \right), \\
\mathcal{R}_n^{nl} &= -\frac{\Delta t}{4\Delta x} L_{nl}\left(q_{nl}\theta_{nl}^n - (U^n)^2\right).
\end{align*}
$$

(17)

In computing the two-norm of 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 $(U^n)^2$. To avoid this computational issue, following the idea presented by Drohmann et al. [2012], this term is calculated empirically, as explained below.

**Assumption 2.** (Drohmann et al. [2012]). By adding a small number of interpolation points and collateral basis functions to the nonlinear approximation, the nonlinear function is approximated with a high accuracy and we assume that it is exact at all points of the discretized spatial domain.

This assumption requires the reduced-order problem to be solved once more with an enriched set of collateral basis functions. Denoting this set of enriched collateral basis functions by $\theta^{*\ell,\ell,n}$ and the corresponding coefficients by $q^{*\ell,\ell,n}$, we assume that the approximation is exact at all points of the computational domain as

$$
(U^n)^2 = q^{*\ell,\ell,n}.n.
$$

(18)

Replacing this assumption into the equation governing $\mathcal{R}_n^{nl}$ leads to

$$
\mathcal{R}_n^{nl} = -\frac{\Delta t}{4\Delta x} L_{nl}(q_{nl}\theta_{nl}^n - q^{*\ell,\ell,n} n).n.
$$

(19)

To compute $\mathcal{R}^n^{T}\mathcal{R}_n$, some operators such as $\Phi^TL_{lin}L_{nl}q_{nl} \in \mathbb{R}^{N_r \times N_m}$ 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 $q^{*\ell,\ell,n}$ or $\Phi$, which is still much lower than the number of actual degrees of freedom of the high-fidelity scheme. In the results presented in this paper, we assume that the dimension of $q^{*\ell,\ell,n}$ (which is $N_{m}1$) is equal to $N_{m}1+1$. Therefore, $\|\mathcal{R}_n\|$ is computed cheaply and the $\ell_2$-norm can be calculated. For the details of residual calculation, we refer to Abbasi et al. [2019]. The fact that Assumption 2 is never exactly satisfied renders the bound (15) to be an error estimate, not an actual error bound.

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

$$
\|e_y^n\|_{\ell_2} \leq \|e_y\|_{\ell_2},
$$

(20)

where $\|e_y\|_{\ell_2}$ is an approximation of the local Lipschitz constant of the nonlinear operator $g(U) = (U)^2$. As this function (for the Burgers’ equation) is not globally Lipschitz, we have to restrict the solution domain to be able to define a finite $L_g$, which is discussed later. The inequality (20) implies

$$
\|e_y\|_{\ell_2} \leq \|e_y\|_{\ell_2}.
$$

(21)

Similar to (15), we have

$$
\|e_y\|_{\ell_2} \leq \gamma^{eR}\|R\|_{\ell_2} + \gamma^{e_u}\|e_y\|_{\ell_2}.
$$

(22)

Combining (21) and (22), assuming that the small-gain condition $L_g\gamma^{e_R} < 1$ holds, leads to

$$
\|e_y\|_{\ell_2} \leq \frac{\gamma^{e_R} e_y^n}{1 - L_g\gamma^{e_R} \|R\|_{\ell_2}}.
$$

(23)

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

$$
\|e_y\|_{\ell_2} \leq \gamma^{e_R} e_y^n + \gamma^{e_u}\gamma^{e_R} L_g\gamma^{e_R} \frac{\kappa(\mu)}{1 - L_g\gamma^{e_R} \|R\|_{\ell_2}}.
$$

(24)

Note that the inequality (20) holds only locally as the value of $L_g$ depends on the magnitude of $U$, which restricts the range of $U$ in the simulations. Assuming $e_y^n := e^n$ to be small and estimating the Lipschitz constant by the derivative of the nonlinear function $L_g = 2\max_{i,n} U^n_i$ reveals that

$$
\max_{i,n} U^n_i < \frac{1}{2\gamma^{e_R} \gamma^{e_u}}.
$$

(25)

ensures that the small-gain condition in (14) is satisfied. This restricts the applicability of this error estimate significantly. To enlarge and shift the applicability region, some alternatives can be pursued. As the simplest alternative, since the maximum of the conservative variables is dependent on $\gamma^{e_u}$, this value is dependent on the simulation parameters (due to the factor $\Delta t/4\Delta x$ in (14)). By decreasing the factor $\Delta t/4\Delta x$, the value of $\gamma^{e_u}$ decreases; however, this leads to inaccurate full-order solutions as the so-called associated CFL number (Leveque [2002]) decreases. Thus, this term cannot be set too small. Another method to alleviate this restriction on the applicability of the error estimate is through loop transformation, which is studied below.

**Loop transformation**

The range of the applicability of 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 3 to Figure 4. The error dynamics after the loop transformation can be written as

\[ \Sigma e : \begin{cases} e_{n+1} = (L_{in} - \frac{\Delta t}{4\Delta \Sigma} L_{nl})e^n + \frac{\Delta t}{4} L_{nl}e_g^n - R^n, \\ e^n = \Sigma e^n, \\ e_g^n = e^n. \end{cases} \]

(26)

It should be noted that \( \Sigma e \) in (14) and (26) are exactly the same. The constant \( \epsilon \) should be defined such that it minimizes the product \( L_{g}\epsilon \) and therefore enlarges the applicability region and also reduces the conservatism in the small-gain condition and the estimate (24). For the parameterized system (26), the following minimization problem is solved to obtain \( \epsilon \),

\[ \epsilon = \arg \min \left( \sum_i \left( \max_i |2U(\mu^i) + \epsilon g(z_{e,n}^i(\mu^i))| \right) \right) \]

\[ \text{s.t. } \forall \mu^i \in D_h \left\{ \rho(L_{in} - \frac{\Delta t}{4\Delta \Sigma} L_{nl}) < 1 \right\} \]

(27)

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

\[ \min(\mu_1, \mu_2) \leq U^\eta(\mu^i) \leq \max(\mu_1, \mu_2). \]

(28)

The constraints in the minimization problem (27) ensure that for each parameter setting, first, the linear part of the error dynamics \( \Sigma e \) is stable, and second, the interconnection of the linear subsystem \( \Sigma_{lin} \) and the nonlinear subsystem \( \Sigma_{nl} \) 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 \( \epsilon \). In other words, to satisfy the small-gain condition, for all members of the parameter domain, we require (based on the second constraint in (27))

\[ \frac{1}{2} \frac{1}{2} \epsilon < u(x; t; \mu) < \frac{1}{2} \frac{1}{2} \epsilon. \]

(29)

Therefore, the parameters, boundary conditions and initial conditions should be chosen in a way that the satisfaction of the above equation would be possible. Based on the knowledge of the dependence of the \( \epsilon \)-gains on \( \epsilon \) 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 (24), even with this loop transformation, can lead to conservative results. To alleviate the conservativeness, we tighten (sharpen) the error estimate as below.

### 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]. Assume that we have the reduced solution with two different levels of accuracy, one using \( N \) basis functions and \( M \) collateral basis functions, the other one with \( N' \) basis functions and \( M' \) collateral basis functions. Assume that we are interested in the error analysis for the case with \( N \) and \( M \) basis functions. To increase the accuracy in the offline phase, based on the snapshots of the current selected parameter \( \mu^i \) in the \( i \)-th iteration of the greedy algorithm, we enrich \( \Phi \) and \( q_{nl} \) step by step. In the offline phase, based on the snapshots of previously selected parameters during the greedy algorithm, we can assign the following relation:

\[ \|y - y_{N,M}^N\|_{\ell_2} \leq \eta_{N,M}^{N,M} \|y - y_{N,M}^N\|_{\ell_2}. \]

(33)

Here, \( y \) is the actual output computed from (6) and \( y_{N,M}^N \) is obtained from (11) with \( N \) basis functions and \( M \) collateral basis functions. In the offline phase, \( N' \) and \( M' \) are increased until \( \eta_{N,M}^{N,M} \) 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.
The effect of using the actual error, the error estimates \( x \) and time step is \( \Delta t \), the spatial grid cells are \( u \leq \Delta t \), which does not lie in the discrete parameter \( \mu \) of parameters selected for the online simulation domain is composed of 8 equidistant members in the parameter domain. The simulation parameters in the online phase \( \mu \) refer to Algorithm 3.

Finite and positive error estimate. For the implementation, the reason for having \( \eta \) is to be sufficiently large to satisfy the condition on 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 \((N', M')\) basis functions should be solved. After obtaining the number of basis functions in the induced error due to the reduction for the parameters used in the online phase is shown in Figure 7.

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 \( \ell_2 \)-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 \( y_{N,M} \). 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.

6 Conclusion

In this paper, a new perspective on the interaction between Empirical Interpolation Method and Reduced Basis method is introduced. First, a new error estimate based on the Lur’e type formulation of the nonlinear Burgers’ equation is defined. This estimate is rigorous, accurate and effective, but has limited applicability due to satisfying the small-gain condition. Furthermore, it requires

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

| Parameter | \( L \) [m] | \( \mu_1 \) | \( \mu_2 \) |
|-----------|-------------|-------------|-------------|
| Minimum   | 100         | 4           | 6           |
| Maximum   | 110         | 5           | 7           |
| Online \( \mu^o \) | 105 | 4.5 | 6.5 |

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

\[
\eta_{N,M}^{N',M'} = \| y_{N',M'} - y_{N,M} \|_{\ell_2}.
\]

Then, based on the following inequality

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

and taking into consideration from the offline phase that

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

we finally obtain

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

The reason for having \( \eta_{N,M}^{N',M'} < 1 \) shows itself here to have finite and positive error estimate. For the implementation, refer to Algorithm 3.

5 Numerical results

The simulation parameters in the online phase \( \mu^o \) for the Burgers’ equation along with the parameter domain are listed in Table 5, 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 5, the set of parameters selected for the online simulation \( \mu^o \) is reported, which does not lie in the discrete parameter domain. This kind of parameter setting ensures that 4 \( \leq u(t,x; \mu) \leq 7 \) for all \( (t,x) \in [0,T] \times [0,L] \). The number of spatial grid cells are \( N = 250 \), the time horizon \( T = 50 \) 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 \( \ell_2 \)-gain notion (with and without the reduction factor \( \rho \) in (32)) and the empirical error estimate in the greedy algorithm of PODEI algorithm (Algorithm 2) as in Drohmann et al. [2012] is shown in Figure 5. Clearly, the error estimates accurately approximate the maximum error in the parameter domain.

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 6 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 basis functions in the induced error due to the reduction for the parameters used in the online phase is shown in Figure 7.

![Fig. 5. Maximum error in the discrete parameter domain during the greedy algorithm.](image)

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

| \( N = M \) | 1 | 5 | 10 | 15 | 20 |
|-------------|---|---|----|----|----|
| Speedup     | 17.4 | 4.2 | 4 | 3.4 | 3 |

In this paper, a new perspective on the interaction between Empirical Interpolation Method and Reduced Basis method is introduced. First, a new error estimate based on the Lur’e type formulation of the nonlinear Burgers’ equation is defined. This estimate is rigorous, accurate and effective, but has limited applicability due to satisfying the small-gain condition. Furthermore, it requires
Fig. 6. Comparison of the full-order and low-order solutions over time using 20 RB functions and 20 collateral basis functions.

Fig. 7. Error evolution by increasing the number of basis functions.

References

Abbasi, M.H., Iapichino, L., Besselink, B., Schilders, W., and van de Wouw, N. (2019). Error estimation in reduced basis method for systems with time-varying and nonlinear boundary conditions. Computer Methods in Applied Mechanics and Engineering, in press.

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.

Benaceur, A., Ehrlicher, V., Ern, A., and Meunier, S. (2018). A Progressive Reduced Basis/Empirical Interpolation Method for Nonlinear Parabolic Problems. SIAM Journal on Scientific Computing, 40(5), A2930–A2955.

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.

Kurganov, A. and Tadmor, E. (2000). New High-Resolution Central Schemes for Nonlinear Conservation Laws and Convection-Diffusion Equations. Journal of Computational Physics, 160(1), 241–282.

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

Leveque, R. (2002). Finite Volume Methods for Hyperbolic Problems. Cambridge Texts in Applied Mathematics.

Lorège, S.N., 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.