



**HAL**  
open science

# Accurate and robust predictions for model order reduction via an adaptive, hybrid FOM/ROM approach

Sébastien Riffaud

► **To cite this version:**

Sébastien Riffaud. Accurate and robust predictions for model order reduction via an adaptive, hybrid FOM/ROM approach. *Journal of Computational Physics*, In press. hal-04361506v2

**HAL Id: hal-04361506**

**<https://inria.hal.science/hal-04361506v2>**

Submitted on 21 Oct 2024 (v2), last revised 16 Dec 2024 (v3)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

## Graphical Abstract

**Accurate and robust predictions for model order reduction via an adaptive, hybrid FOM/ROM approach**

Sébastien Riffaud

## Highlights

### **Accurate and robust predictions for model order reduction via an adaptive, hybrid FOM/ROM approach**

Sébastien Riffaud

- An approach based on the principal directions of the reduced-order residual and reduced-order Jacobian matrix is proposed to improve the accuracy and computational efficiency of the ECSW method.
- A hybrid strategy that alternates between the FOM and ROM is employed to accelerate numerical simulations while maintaining accurate approximations.
- An error indicator based on a reduced-order approximation of the error is developed to determine when the ROM is not sufficiently accurate and the FOM needs to be employed.
- An adaptive-extended version of the hybrid approach is proposed to update the ROM with new solution snapshots.

# Accurate and robust predictions for model order reduction via an adaptive, hybrid FOM/ROM approach

Sébastien Riffaud<sup>a</sup>

<sup>a</sup>*Sorbonne Université, Inria & CNRS, UMR 7598, Laboratoire Jacques-Louis Lions, 75012 Paris, France*

---

## Abstract

In this paper, we introduce a hybrid approach that alternates between a high-fidelity model and a reduced-order model to speedup numerical simulations while maintaining accurate approximations. In particular, an error indicator based on a reduced-order approximation of the error is developed to determine when the reduced-order model is not sufficiently accurate and the high-fidelity model needs to be solved. Then, we propose an adaptive-extended version of the hybrid approach to update the reduced-order model with new solution snapshots. In particular, an interesting application concerns the acceleration of the offline stage of ROMs. The performance of the proposed method is finally evaluated on parametrized, time-dependent, nonlinear problems governed by the 1D Burgers' equation and 2D compressible Euler equations. The results demonstrate the accuracy and computational efficiency of the adaptive, hybrid approach with respect to the high-fidelity model.

*Keywords:* Reduced-order model, Proper orthogonal decomposition, Hyper-reduction, Error estimation, Adaptive model reduction

---

## 1. Introduction

Many applications in science and engineering require efficient numerical simulations, either due to runtime constraints in the case of large-scale computational models or due to the large number of simulations to run in the case of many-query problems. Projection-based reduced-order models (ROMs) [1, 2, 3, 4, 5, 6, 7, 8, 9] have been developed to dramatically reduce the computational cost associated with numerical solutions of parametric problems governed by partial differential equations (PDEs). They typically

consist of an offline stage in which training high-fidelity solutions are collected to define a low-dimensional approximation space. A popular dimensionality reduction technique for extracting the reduced basis that spans this approximation space is the proper orthogonal decomposition (POD) [10, 2]. Then, the low-dimensionality of the reduced basis is exploited during an online stage to enable fast or real-time predictions of new out-of-sample solutions.

In order to construct accurate ROMs over a wide range of input parameters, the standard approach consists in collecting a large number of high-fidelity solution snapshots at different training time-parameter instances. However, there are two major difficulties associated with this offline procedure. First, the number of high-fidelity simulations for sampling the solution manifold is limited in practice due to the expensive computational cost of the high-fidelity model (or high-dimensional model (HDM)). Second, if the training solution snapshots are too different from the predicted solution, the ROM may fail to deliver accurate approximations.

Several approaches have been proposed in the literature to address the aforementioned issues. In multifidelity methods [11, 12], the solutions of multilevel hierarchical models are combined to accelerate the estimation of statistics. The number of solution evaluations between the high- and low-fidelity models is balanced with respect to error and cost using an optimal model management. Alternatively, a hybrid HDM/ROM approach is proposed in [13, 14, 15]. The main idea is to switch between the HDM and ROM on the fly to achieve a prescribed level of accuracy. An *a posteriori* error estimation is employed to determine when the ROM is not sufficiently accurate and the HDM needs to be solved. Similarly, Zucatti et al. [16] introduced an adaptive, training-free ROM for convection-dominated problems based on hybrid snapshots. The latter are generated by combining space-local HDM and ROM solutions. An error strategy is employed to identify regions where the ROM is inaccurate and the HDM needs to be solved.

In this work, we focus on ROMs based on POD in the offline stage and on the least-squares Petrov-Galerkin (LSPG) projection [17, 18] equipped with the energy-conserving sampling and weighting (ECSW) hyper-reduction method [19, 20] in the online stage. In particular, we propose an approach [21] based on the principal directions of the reduced-order residual and reduced-order Jacobian matrix to improve the performance of the ECSW method. With this framework, we then present a hybrid HDM/ROM model to accelerate numerical simulations while maintaining accurate approximations. The main contribution concerns the introduction of an error indicator based

on a reduced-order approximation of the error. Compared to the hybrid model presented in [15], the error indicator proposed here can be employed for Galerkin or Petrov-Galerkin projection-based ROMs equipped with any hyper-reduction method. The computational efficiency of the resulting hybrid model depends on the accuracy of the underlying ROM to predict out-of-sample solutions. If the HDM is used in a large part of the computations, the hybrid model will be accurate but computationally expensive to solve. Conversely, if the ROM is sufficient to approximate the solution most of the time, this approach enables a significant reduction in the computational cost of numerical simulations. An interesting application of the hybrid model concerns the acceleration of the offline stage of ROMs. The main idea is to generate solution snapshots using the HDM only when the ROM is not sufficiently accurate. Moreover, instead of storing all solution snapshots, we propose to progressively enrich the ROM with the new solution snapshots in order to mitigate computational memory requirements. In this way, we also expect the ROM to become more robust for predicting out-of-sample solutions, which in turn allows to further accelerate the offline procedure.

The remainder of the paper is organized as follows. In Section 2, we introduce the HDM and ROM used in this work. Then, Section 3 describes the hybrid HDM/ROM approach and the underlying error indicator. In Section 4, we present an application of the adaptive, hybrid approach for accelerating the offline stage of ROMs. Section 5 demonstrates the performance of the proposed methods for parametrized, time-dependent, nonlinear problems. Finally, Section 6 draws some conclusions and perspectives.

## 2. Projection-based reduced-order model

### 2.1. High-dimensional model

Let the parameter domain  $\mathcal{D}$  be a closed and bounded subset of the Euclidean space  $\mathbb{R}^{N_p}$ . In this work, we consider parametrized, time-dependent, nonlinear, semi-discrete problems of the form

$$\begin{cases} \frac{d\mathbf{u}}{dt}(t; \boldsymbol{\theta}) = \mathbf{f}(\mathbf{u}(t; \boldsymbol{\theta}), t; \boldsymbol{\theta}) \\ \mathbf{u}(0; \boldsymbol{\theta}) = \mathbf{u}_0(\boldsymbol{\theta}), \end{cases}$$

where  $t \geq 0$  denotes time,  $\boldsymbol{\theta} \in \mathcal{D}$  denotes the input parameters,  $\mathbf{u} : \mathbb{R}_+ \times \mathcal{D} \rightarrow \mathbb{R}^{N_{\text{dof}}}$  denotes the solution, and  $\mathbf{f} : \mathbb{R}^{N_{\text{dof}}} \times \mathbb{R}_+ \times \mathcal{D} \rightarrow \mathbb{R}^{N_{\text{dof}}}$  denotes the nonlinear function resulting from the semi-discretization of the PDE.

This semi-discrete problem is then integrated in time by an implicit scheme, leading to the nonlinear system of equations

$$\mathbf{r}_k(\mathbf{u}_k(\boldsymbol{\theta}); \boldsymbol{\theta}) = \mathbf{0} \quad \forall k \in \{1, \dots, N_t\}, \quad (1)$$

which is solved for  $\mathbf{u}_k(\boldsymbol{\theta}) \in \mathbb{R}^{N_{\text{dof}}}$  by the Newton-Raphson method. Here,  $\mathbf{u}_k(\boldsymbol{\theta})$  denotes the time-discrete counterpart of  $\mathbf{u}(t_k; \boldsymbol{\theta})$ ,  $t_k \geq 0$  denotes the  $k$ -th time-instance, and  $\mathbf{r}_k : \mathbb{R}^{N_{\text{dof}}} \times \mathcal{D} \rightarrow \mathbb{R}^{N_{\text{dof}}}$  denotes the discrete approximation of the residual  $\mathbf{r}(\mathbf{u}(t_k; \boldsymbol{\theta}), t_k; \boldsymbol{\theta}) = \frac{d\mathbf{u}}{dt}(t_k; \boldsymbol{\theta}) - \mathbf{f}(\mathbf{u}(t_k; \boldsymbol{\theta}), t_k; \boldsymbol{\theta})$ . In the following, this computational model will be referred to as the HDM.

## 2.2. Reduced-order model

In the ROM, the high-dimensional solution  $\mathbf{u}_k(\boldsymbol{\theta})$  is approximated on a low-dimensional subspace to reduce the number of degrees of freedom:

$$\tilde{\mathbf{u}}_k(\boldsymbol{\theta}) = \mathbf{u}_{\text{off}}(\boldsymbol{\theta}) + \mathbf{V}\mathbf{y}_k(\boldsymbol{\theta}),$$

where the offset  $\mathbf{u}_{\text{off}}(\boldsymbol{\theta}) \in \mathbb{R}^{N_{\text{dof}}}$  is set to the initial condition  $\mathbf{u}_0(\boldsymbol{\theta})$ , the reduced basis  $\mathbf{V} \in \mathbb{R}^{N_{\text{dof}} \times N_v}$  is constructed by POD, the reduced coordinates  $\mathbf{y}_k(\boldsymbol{\theta}) \in \mathbb{R}^{N_v}$  are determined by the LSPG projection equipped with the ECSW hyper-reduction method, and  $N_v \ll N_{\text{dof}}$ .

### 2.2.1. Proper orthogonal decomposition

During the offline stage, high-dimensional solution snapshots are collected at different training time-parameter instances  $\{(t_1^{\text{train}}, \boldsymbol{\theta}_1^{\text{train}}), \dots, (t_{N_s}^{\text{train}}, \boldsymbol{\theta}_{N_s}^{\text{train}})\}$ . Let the resulting snapshot database be  $\mathbf{S}_u = [\mathbf{s}_{u_1}, \dots, \mathbf{s}_{u_{N_s}}] \in \mathbb{R}^{N_{\text{dof}} \times N_s}$ , where  $\mathbf{s}_{u_k} \in \mathbb{R}^{N_{\text{dof}}}$  is the discrete approximation of  $\mathbf{u}(t_k^{\text{train}}, \boldsymbol{\theta}_k^{\text{train}})$ . The reduced basis  $\mathbf{V}$  is computed by POD to minimize the projection error of the snapshots onto this basis:

$$\begin{cases} \text{minimize} & \|\mathbf{S}_u - \mathbf{S}_{u_{\text{off}}} - \mathbf{V}\mathbf{V}^T(\mathbf{S}_u - \mathbf{S}_{u_{\text{off}}})\|_F \\ \text{subject to} & \mathbf{V}^T\mathbf{V} = \mathbf{I}, \end{cases}$$

where  $\mathbf{S}_{u_{\text{off}}} = [\mathbf{u}_{\text{off}}(\boldsymbol{\theta}_1^{\text{train}}), \dots, \mathbf{u}_{\text{off}}(\boldsymbol{\theta}_{N_s}^{\text{train}})] \in \mathbb{R}^{N_{\text{dof}} \times N_s}$ , and  $\|\cdot\|_F$  stands for the Frobenius norm. Notably, the solution to this minimization problem is given by the Eckart-Young theorem [22]. Let  $\hat{\mathbf{S}}_u = \mathbf{\Phi}\mathbf{\Sigma}\mathbf{\Psi}^T$  be the truncated singular value decomposition (SVD) of  $\mathbf{S}_u - \mathbf{S}_{u_{\text{off}}}$ , where  $\mathbf{\Phi} \in \mathbb{R}^{N_{\text{dof}} \times N_v}$  is an orthonormal matrix,  $\mathbf{\Sigma} = \text{diag}(\sigma_1, \dots, \sigma_{N_v}) \in \mathbb{R}^{N_v \times N_v}$  is a diagonal matrix with non-negative real number on the diagonal, and  $\mathbf{\Psi} \in \mathbb{R}^{N_s \times N_v}$  is an

orthogonal matrix. The reduced basis corresponds to the  $N_v$  leading left singular vectors (i.e.,  $\mathbf{V} = \mathbf{\Phi}$ ), and the square of the projection error is equal to the sum of the neglected singular values squared:

$$\|\mathbf{S}_u - \mathbf{S}_{u_{\text{off}}} - \mathbf{V}\mathbf{V}^T(\mathbf{S}_u - \mathbf{S}_{u_{\text{off}}})\|_F^2 = \sum_{n=N_v+1}^{\min(N_{\text{dof}}, N_s)} \sigma_n^2.$$

In particular, the reduced basis dimension  $N_v$  is chosen here as the smallest integer such that the relative projection error is less than a user-defined tolerance  $\varepsilon_{\text{pod}} \in [0, 1]$ :

$$\|\mathbf{S}_u - \mathbf{S}_{u_{\text{off}}} - \mathbf{V}\mathbf{V}^T(\mathbf{S}_u - \mathbf{S}_{u_{\text{off}}})\|_F \leq \varepsilon_{\text{pod}} \|\mathbf{S}_u\|_F.$$

Furthermore, the approximation of the different variables is decorrelated since they can have different orders of magnitude. For example, in Section 5.2, the variables  $\{\rho, \rho u, \rho v, E\}$  are approximated on  $\{\mathbf{V}_\rho, \mathbf{V}_{\rho u}, \mathbf{V}_{\rho v}, \mathbf{V}_E\}$  respectively, and the reduced basis is defined as

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_\rho & & & \\ & \mathbf{V}_{\rho u} & & \\ & & \mathbf{V}_{\rho v} & \\ & & & \mathbf{V}_E \end{bmatrix}.$$

### 2.2.2. Least-squares Petrov-Galerkin method

During the online stage, the reduced coordinates  $\mathbf{y}_k(\boldsymbol{\theta})$  are determined by the residual-minimizing LSPG method:

$$\begin{cases} \text{minimize}_{\mathbf{y}_k(\boldsymbol{\theta}) \in \mathbb{R}^{N_v}} \|\mathbf{r}_k(\tilde{\mathbf{u}}_k(\boldsymbol{\theta}); \boldsymbol{\theta})\|_2^2 & \forall k \in \{1, \dots, N_t\} \\ \mathbf{y}_0(\boldsymbol{\theta}) = \mathbf{V}^T(\mathbf{u}_0(\boldsymbol{\theta}) - \mathbf{u}_{\text{off}}(\boldsymbol{\theta})). \end{cases} \quad (2)$$

Notably, the first optimality condition leads to the Petrov-Galerkin projection:

$$\begin{cases} (\mathbf{J}_k(\tilde{\mathbf{u}}_k(\boldsymbol{\theta}); \boldsymbol{\theta})\mathbf{V})^T \mathbf{r}_k(\tilde{\mathbf{u}}_k(\boldsymbol{\theta}); \boldsymbol{\theta}) = \mathbf{0} & \forall k \in \{1, \dots, N_t\} \\ \mathbf{y}_0(\boldsymbol{\theta}) = \mathbf{V}^T(\mathbf{u}_0(\boldsymbol{\theta}) - \mathbf{u}_{\text{off}}(\boldsymbol{\theta})), \end{cases}$$

where  $\mathbf{J}_k(\mathbf{u}_k(\boldsymbol{\theta}); \boldsymbol{\theta}) \in \mathbb{R}^{N_{\text{dof}} \times N_{\text{dof}}}$  denotes the Jacobian of  $\mathbf{r}_k(\mathbf{u}_k(\boldsymbol{\theta}); \boldsymbol{\theta})$  with respect to  $\mathbf{u}_k$ .



### 2.2.3. Energy-conserving sampling and weighting method

In this work, the nonlinear least-squares problem (2) is solved by the Gauss-Newton algorithm, which involves the projected reduced-order quantities

$$\bar{\mathbf{J}}_k(\boldsymbol{\theta})^T \bar{\mathbf{J}}_k(\boldsymbol{\theta}) \quad \text{and} \quad \bar{\mathbf{J}}_k(\boldsymbol{\theta})^T \bar{\mathbf{r}}_k(\boldsymbol{\theta}),$$

where  $\bar{\mathbf{r}}_k(\boldsymbol{\theta}) = \mathbf{r}_k(\tilde{\mathbf{u}}_k(\boldsymbol{\theta}); \boldsymbol{\theta}) \in \mathbb{R}^{N_{\text{dof}}}$  and  $\bar{\mathbf{J}}_k(\boldsymbol{\theta}) = \mathbf{J}_k(\tilde{\mathbf{u}}_k(\boldsymbol{\theta}); \boldsymbol{\theta}) \mathbf{V} \in \mathbb{R}^{N_{\text{dof}} \times N_v}$ . Unfortunately, as the residual is nonlinear with respect to the solution, the evaluation of these quantities typically scales with the size of the HDM, unless hyper-reduction techniques are employed. For this reason, we consider the ECSW method, in which the standard scalar product is approximated by the weighted sum:

$$\mathbf{a}^T \mathbf{b} \approx \sum_{i=1}^{N_{\text{dof}}} \omega_i a_i b_i,$$

where  $\mathbf{a}, \mathbf{b} \in \mathbb{R}^{N_{\text{dof}}}$ , and only a few weights  $\omega_i \geq 0$  are nonzero (i.e.,  $\|\boldsymbol{\omega}\|_0 \ll N_{\text{dof}}$ , where  $\|\cdot\|_0$  stands for the  $\ell_0$  pseudo-norm). In this way, the reduced-order quantities  $\bar{\mathbf{r}}_k(\boldsymbol{\theta})$  and  $\bar{\mathbf{J}}_k(\boldsymbol{\theta})$  need to be evaluated only at a few points of the spatial domain, which are associated with nonzero weights, in order to speedup computations. This set of points will be referred to as the reduced mesh in the following, and  $N_p$  will denote the size of the reduced mesh.

The weights  $\boldsymbol{\omega} = [\omega_1, \dots, \omega_{N_{\text{dof}}}] \in \mathbb{R}^{N_{\text{dof}}}$  are determined during the offline stage to best approximate the exact scalar product. Given snapshots of the reduced-order residual  $\mathbf{r}_k(\tilde{\mathbf{u}}_k(\boldsymbol{\theta}); \boldsymbol{\theta})$  and reduced-order Jacobian matrix  $\mathbf{J}_k(\tilde{\mathbf{u}}_k(\boldsymbol{\theta}); \boldsymbol{\theta}) \mathbf{V}$  at different time-parameter instances  $(t_k, \boldsymbol{\theta}) \in \{(t_1^{\text{train}}, \boldsymbol{\theta}_1^{\text{train}}), \dots, (t_{N_s}^{\text{train}}, \boldsymbol{\theta}_{N_s}^{\text{train}})\}$ , the evaluation of the projected reduced-order quantities leads to an approximation problem of the form

$$\mathbf{C}\boldsymbol{\omega} \approx \mathbf{d}, \tag{3}$$

where  $\mathbf{C} \in \mathbb{R}^{N_c \times N_{\text{dof}}}$  and  $\mathbf{d} \in \mathbb{R}^{N_c}$  are defined in Appendix A. The weights  $\boldsymbol{\omega}$  are then solution to the non-negative least-squares (NNLS) problem

$$\min_{\boldsymbol{\omega} \geq 0} \|\mathbf{C}\boldsymbol{\omega} - \mathbf{d}\|_2^2, \tag{4}$$

which is solved by the Matlab implementation of the Lawson-Hanson algorithm [23]. This algorithm promotes sparsity in the solution and terminates when the stopping criterion  $\|\mathbf{C}\boldsymbol{\omega} - \mathbf{d}\|_2 \leq \varepsilon_{\text{ecsw}} \|\mathbf{d}\|_2$  is satisfied for a user-defined tolerance  $\varepsilon_{\text{ecsw}} \in [0, 1]$ .

Notably, we propose in Appendix A to construct  $\mathbf{C}$  using the principal directions of the reduced-order residual and reduced-order Jacobian matrix.

Specifically, instead of constructing  $\mathbf{C}$  directly from snapshots of the reduced-order quantities, we consider the POD basis  $\mathbf{U} \in \mathbb{R}^{N_{\text{dof}} \times N_u}$  of these snapshots, and we define  $\mathbf{C}^T = \mathbf{U}\mathbf{\Lambda}$ , where  $\mathbf{\Lambda} \in \mathbb{R}^{N_u \times N_u}$  is the diagonal matrix containing the singular values associated to  $\mathbf{U}$ . This modification allows to drastically reduce the size of the approximation problem (3) and, consequently, the runtime for solving the NLS problem (4). In addition, the approximation problem (3) is fully described by the reduced basis  $\mathbf{U}$  and the singular values  $\mathbf{\Lambda}$ , which will be a key point for updating the reduced mesh in Section 4.2.

**Remark 1.** *To compute snapshots of  $\mathbf{r}_k(\tilde{\mathbf{u}}_k(\boldsymbol{\theta}); \boldsymbol{\theta})$  and  $\mathbf{J}_k(\tilde{\mathbf{u}}_k(\boldsymbol{\theta}); \boldsymbol{\theta})\mathbf{V}$ , the reduced-order solution  $\tilde{\mathbf{u}}_k(\boldsymbol{\theta})$  is replaced in practice by the projection of the high-dimensional solution  $\mathbf{u}_{\text{off}}(\boldsymbol{\theta}) + \mathbf{V}\mathbf{V}^T(\mathbf{u}_k(\boldsymbol{\theta}) - \mathbf{u}_{\text{off}}(\boldsymbol{\theta}))$  in order to save the computation time associated with the evaluation of the ROM.*

### 3. Hybrid HDM/ROM model

Given the previous HDM and ROM, we now introduce the hybrid model, which aims at accelerating numerical simulations while maintaining accurate approximations. The main idea is to compute the solution using the HDM only when the ROM is not sufficiently accurate. Specifically, we want the relative space-time error to be smaller than a prescribed tolerance  $\varepsilon_{\text{err}} \in [0, 1]$ :

$$\sqrt{\sum_{k=1}^{N_t} \|\mathbf{u}_k(\boldsymbol{\theta}) - \tilde{\mathbf{u}}_k(\boldsymbol{\theta})\|_2^2} \leq \varepsilon_{\text{err}} \sqrt{\sum_{k=1}^{N_t} \|\mathbf{u}_k(\boldsymbol{\theta})\|_2^2}.$$

For this purpose, we impose at each time-step  $t_k$  that

$$\|\mathbf{u}_k(\boldsymbol{\theta}) - \tilde{\mathbf{u}}_k(\boldsymbol{\theta})\|_2 \leq \varepsilon_{\text{err}} \|\mathbf{u}_k(\boldsymbol{\theta})\|_2,$$

which is equivalent to bound the error  $\mathbf{e}_k(\boldsymbol{\theta}) = \mathbf{u}_k(\boldsymbol{\theta}) - \tilde{\mathbf{u}}_k(\boldsymbol{\theta})$  as follows

$$\|\mathbf{e}_k(\boldsymbol{\theta})\|_2 \leq \varepsilon_{\text{err}} \|\mathbf{e}_k(\boldsymbol{\theta}) + \tilde{\mathbf{u}}_k(\boldsymbol{\theta})\|_2. \quad (5)$$

However, computing the error  $\mathbf{e}_k(\boldsymbol{\theta})$  is computationally prohibitive as it requires the evaluation of the high-dimensional solution  $\mathbf{u}_k(\boldsymbol{\theta})$ . For this reason, we employ an error estimation  $\tilde{\mathbf{e}}_k(\boldsymbol{\theta}) \approx \mathbf{e}_k(\boldsymbol{\theta})$ . The latter is presented in Section 3.1, and the resulting hybrid model is described in Algorithm 1.

---

**Algorithm 1** Hybrid model
 

---

**Input:**  $\boldsymbol{\theta} \in \mathcal{D}$ , and  $\varepsilon_{\text{err}} \in [0, 1]$ .

**Output:**  $\{\hat{\mathbf{u}}_1(\boldsymbol{\theta}), \dots, \hat{\mathbf{u}}_{N_t}(\boldsymbol{\theta})\} \in \mathbb{R}^{N_{\text{dof}}}$ .

```

1: Initialize  $\hat{\mathbf{u}}_0(\boldsymbol{\theta}) = \mathbf{u}_0(\boldsymbol{\theta})$ ;
2: for  $k \in \{1, \dots, N_t\}$  do
3:   Compute  $\tilde{\mathbf{u}}_k(\boldsymbol{\theta})$  by solving equation (2);
4:   Compute  $\tilde{\mathbf{e}}_k(\boldsymbol{\theta})$  by solving equation (6);
5:   if  $\|\tilde{\mathbf{e}}_k(\boldsymbol{\theta})\|_2 \leq \varepsilon_{\text{err}} \|\tilde{\mathbf{e}}_k(\boldsymbol{\theta}) + \tilde{\mathbf{u}}_k(\boldsymbol{\theta})\|_2$  then
6:     Set  $\hat{\mathbf{u}}_k(\boldsymbol{\theta}) = \tilde{\mathbf{u}}_k(\boldsymbol{\theta})$ ;
7:   else
8:     Compute  $\mathbf{u}_k(\boldsymbol{\theta})$  by solving equation (1);
9:     Set  $\hat{\mathbf{u}}_k(\boldsymbol{\theta}) = \mathbf{u}_k(\boldsymbol{\theta})$ ;
10:  end
11: end

```

---

### 3.1. Error estimation

In this work, the error is estimated by a second ROM, which will be referred to as the error indicator in the following and which uses the same methods as the ROM presented in Section 2.2. Specifically, the error  $\mathbf{e}_k(\boldsymbol{\theta})$  is approximated as

$$\tilde{\mathbf{e}}_k(\boldsymbol{\theta}) = \mathbf{e}_{\text{off}}(\boldsymbol{\theta}) + \mathbf{W}\mathbf{z}_k(\boldsymbol{\theta}),$$

where the offset  $\mathbf{e}_{\text{off}}(\boldsymbol{\theta}) \in \mathbb{R}^{N_{\text{dof}}}$  is set to the initial error  $\mathbf{e}_0(\boldsymbol{\theta})$ , the reduced basis  $\mathbf{W} \in \mathbb{R}^{N_{\text{dof}} \times N_w}$  is constructed by POD, and the reduced coordinates  $\mathbf{z}_k(\boldsymbol{\theta}) \in \mathbb{R}^{N_w}$  are determined by the LSPG projection equipped with the ECSW hyper-reduction method. Note that the initial error  $\mathbf{e}_0(\boldsymbol{\theta})$  vanishes here since  $\mathbf{u}_{\text{off}}(\boldsymbol{\theta}) = \mathbf{u}_0(\boldsymbol{\theta})$ .

During the offline stage, solution snapshots of the ROM are collected to assemble the database  $\mathbf{S}_e = [\mathbf{s}_{e_1}, \dots, \mathbf{s}_{e_{N_s}}] \in \mathbb{R}^{N_{\text{dof}} \times N_s}$ , where  $\mathbf{s}_{e_k}$  is the snapshot of the error  $\mathbf{e}_k(\boldsymbol{\theta})$  at time-parameter instance  $(t_k, \boldsymbol{\theta}) = (t_k^{\text{train}}, \boldsymbol{\theta}_k^{\text{train}})$ . The reduced basis  $\mathbf{W}$  is then computed by POD from  $\mathbf{S}_e$  without offset (i.e.,  $\mathbf{S}_{e_{\text{off}}} = \mathbf{0}$ ).

The reduced coordinates  $\mathbf{z}_k(\boldsymbol{\theta})$  are determined during the online stage by the LSPG method:

$$\begin{cases} \underset{\mathbf{z}_k(\boldsymbol{\theta}) \in \mathbb{R}^{N_w}}{\text{minimize}} \|\mathbf{r}_k(\tilde{\mathbf{e}}_k(\boldsymbol{\theta}) + \tilde{\mathbf{u}}_k(\boldsymbol{\theta}); \boldsymbol{\theta})\|_2^2 & \forall k \in \{1, \dots, N_t\} \\ \mathbf{z}_0(\boldsymbol{\theta}) = \mathbf{W}^T(\mathbf{e}_0(\boldsymbol{\theta}) - \mathbf{e}_{\text{off}}(\boldsymbol{\theta})), \end{cases} \quad (6)$$

where the reduced-order solution  $\tilde{\mathbf{u}}_k(\boldsymbol{\theta})$  is known. This nonlinear least-squares problem is then solved by the Gauss-Newton algorithm, and the ECSW method is employed to reduce the computational cost associated with the evaluation the projected reduced-order quantities. Compared to the procedure described in Section 2.2.3, the only difference is that the approximation problem (3) is assembled using snapshots of the reduced-order residual  $\mathbf{r}_k(\tilde{\mathbf{e}}_k(\boldsymbol{\theta}) + \tilde{\mathbf{u}}_k(\boldsymbol{\theta}); \boldsymbol{\theta})$  and reduced-order Jacobian matrix  $\mathbf{J}_k(\tilde{\mathbf{e}}_k(\boldsymbol{\theta}) + \tilde{\mathbf{u}}_k(\boldsymbol{\theta}); \boldsymbol{\theta})\mathbf{W}$ .

**Remark 2.** *To compute snapshots of  $\mathbf{r}_k(\tilde{\mathbf{e}}_k(\boldsymbol{\theta}) + \tilde{\mathbf{u}}_k(\boldsymbol{\theta}); \boldsymbol{\theta})$  and  $\mathbf{J}_k(\tilde{\mathbf{e}}_k(\boldsymbol{\theta}) + \tilde{\mathbf{u}}_k(\boldsymbol{\theta}); \boldsymbol{\theta})\mathbf{W}$ , the reduced-order error  $\tilde{\mathbf{e}}_k(\boldsymbol{\theta})$  is replaced in practice by the projection of the error  $\mathbf{e}_{\text{off}}(\boldsymbol{\theta}) + \mathbf{W}\mathbf{W}^T(\mathbf{e}_k(\boldsymbol{\theta}) - \mathbf{u}_{\text{off}}(\boldsymbol{\theta}))$  in order to save the computation time associated with the evaluation of the error indicator.*

**Remark 3.** *Replacing  $\mathbf{e}_k(\boldsymbol{\theta})$  with  $\tilde{\mathbf{e}}_k(\boldsymbol{\theta})$  enables the efficient evaluation of the error criterion (5) as it scales with the size of the ROM and error indicator:*

$$\begin{aligned} \|\tilde{\mathbf{e}}_k(\boldsymbol{\theta}) + \tilde{\mathbf{u}}_k(\boldsymbol{\theta})\|_2^2 &= \|\tilde{\mathbf{e}}_k(\boldsymbol{\theta})\|_2^2 + 2\langle \tilde{\mathbf{e}}_k(\boldsymbol{\theta}), \tilde{\mathbf{u}}_k(\boldsymbol{\theta}) \rangle_2 + \|\tilde{\mathbf{u}}_k(\boldsymbol{\theta})\|_2^2, \\ \|\tilde{\mathbf{e}}_k(\boldsymbol{\theta})\|_2^2 &= \|\mathbf{e}_{\text{off}}(\boldsymbol{\theta})\|_2^2 + 2\langle \mathbf{W}^T \mathbf{e}_{\text{off}}(\boldsymbol{\theta}), \mathbf{z}_k(\boldsymbol{\theta}) \rangle_2 + \|\mathbf{z}_k(\boldsymbol{\theta})\|_2^2, \\ \langle \tilde{\mathbf{e}}_k(\boldsymbol{\theta}), \tilde{\mathbf{u}}_k(\boldsymbol{\theta}) \rangle_2 &= \langle \mathbf{e}_{\text{off}}(\boldsymbol{\theta}), \mathbf{u}_{\text{off}}(\boldsymbol{\theta}) \rangle_2 + \langle \mathbf{V}^T \mathbf{e}_{\text{off}}(\boldsymbol{\theta}), \mathbf{y}_k(\boldsymbol{\theta}) \rangle_2 \\ &\quad + \langle \mathbf{W}^T \mathbf{u}_{\text{off}}(\boldsymbol{\theta}), \mathbf{z}_k(\boldsymbol{\theta}) \rangle_2 + \langle \mathbf{V}^T \mathbf{W} \mathbf{z}_k(\boldsymbol{\theta}), \mathbf{y}_k(\boldsymbol{\theta}) \rangle_2, \\ \|\tilde{\mathbf{u}}_k(\boldsymbol{\theta})\|_2^2 &= \|\mathbf{u}_{\text{off}}(\boldsymbol{\theta})\|_2^2 + 2\langle \mathbf{V}^T \mathbf{u}_{\text{off}}(\boldsymbol{\theta}), \mathbf{y}_k(\boldsymbol{\theta}) \rangle_2 + \|\mathbf{y}_k(\boldsymbol{\theta})\|_2^2, \end{aligned}$$

where  $\|\mathbf{u}_{\text{off}}(\boldsymbol{\theta})\|_2^2$ ,  $\|\mathbf{e}_{\text{off}}(\boldsymbol{\theta})\|_2^2$ ,  $\langle \mathbf{e}_{\text{off}}(\boldsymbol{\theta}), \mathbf{u}_{\text{off}}(\boldsymbol{\theta}) \rangle_2$ ,  $\mathbf{V}^T \mathbf{u}_{\text{off}}(\boldsymbol{\theta})$ ,  $\mathbf{V}^T \mathbf{e}_{\text{off}}(\boldsymbol{\theta})$ ,  $\mathbf{W}^T \mathbf{u}_{\text{off}}(\boldsymbol{\theta})$ ,  $\mathbf{W}^T \mathbf{e}_{\text{off}}(\boldsymbol{\theta})$ , and  $\mathbf{V}^T \mathbf{W}$  can be precomputed once for all.

### 3.2. Computational complexity

According to Algorithm 1, the computational cost of the hybrid model is given by

$$(c_{\text{rom}} + c_{\text{ind}})N_t + c_{\text{hdm}}n_{\text{hdm}},$$

where  $c_{\text{rom}}$  denotes the cost for solving equation (2),  $c_{\text{ind}}$  denotes the cost for solving equation (6),  $c_{\text{hdm}}$  denotes the cost for solving equation (1), and  $n_{\text{hdm}}$  denotes the number of time instances in which the HDM is employed to compute the solution. In this way, if the ROM is sufficient to approximate the solution most of the time, the hybrid model enables a significant reduction in the computational cost of numerical simulations since  $c_{\text{rom}}, c_{\text{ind}} \ll c_{\text{hdm}}$ .

#### 4. Acceleration of the offline stage of ROMs via an adaptive, hybrid approach

An interesting application of the hybrid model concerns the acceleration of the offline stage of ROMs. The main idea is to generate solution snapshots using the HDM only when the ROM is not sufficiently accurate. In this way, if the ROM is sufficient to approximate the solution snapshots most of the time, then this approach enables a significant reduction in the computational cost of the offline stage. Moreover, instead of storing all solution snapshots, the ROM of the hybrid model is progressively enriched with the new solution snapshots to reduce computational memory requirements. Notably, an efficient procedure for updating the reduced basis  $\mathbf{V}$  is described in Section 4.1. The same procedure is then applied to the reduced bases  $\mathbf{U}$  and  $\mathbf{W}$  in order to fully update the ROM and error indicator. The resulting adaptive-extended version of the hybrid model is presented in Algorithm 2.

---

#### Algorithm 2 Adaptive, hybrid approach

---

**Input:**  $\{\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_{N_\theta}\} \in \mathcal{D}$ .

**Output:**  $\{\hat{\mathbf{u}}_k(\boldsymbol{\theta}_1), \dots, \hat{\mathbf{u}}_k(\boldsymbol{\theta}_{N_\theta})\}_{k=1}^{N_t} \in \mathbb{R}^{N_{\text{dof}}}$ .

- 1: **for**  $\boldsymbol{\theta} \in \{\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_{N_\theta}\}$  **do**
  - 2:     Compute  $\{\hat{\mathbf{u}}_1(\boldsymbol{\theta}), \dots, \hat{\mathbf{u}}_{N_t}(\boldsymbol{\theta})\}$  using Algorithm 1;
  - 3:     Set  $\mathbf{S}_u^{\text{new}} = [\hat{\mathbf{u}}_1(\boldsymbol{\theta}), \dots, \hat{\mathbf{u}}_{N_t}(\boldsymbol{\theta})]$  and  $\mathbf{S}_{u_{\text{off}}}^{\text{new}} = [\mathbf{u}_0(\boldsymbol{\theta}), \dots, \mathbf{u}_0(\boldsymbol{\theta})]$ ;
  - 4:     Update the reduced basis according to Section 4.1;
  - 5:     Update the reduced mesh according to Section 4.2;
  - 6:     Update the error indicator according to Section 4.3;
  - 7: **end**
- 

##### 4.1. Update of the reduced basis

Let  $\mathbf{V}_{\text{old}} \in \mathbb{R}^{N_{\text{dof}} \times N_{\text{vold}}}$  be the POD basis computed from the databases  $\mathbf{S}_u^{\text{old}}, \mathbf{S}_{u_{\text{off}}}^{\text{old}} \in \mathbb{R}^{N_{\text{dof}} \times N_{\text{sold}}}$ , and assume that new snapshots  $\mathbf{S}_u^{\text{new}}, \mathbf{S}_{u_{\text{off}}}^{\text{new}} \in \mathbb{R}^{N_{\text{dof}} \times N_{\text{snew}}}$  become available. The objective is to find the reduced basis  $\mathbf{V}$  that verifies

$$\|\mathbf{S}_u - \mathbf{S}_{u_{\text{off}}} - \mathbf{V}\mathbf{V}^T(\mathbf{S}_u - \mathbf{S}_{u_{\text{off}}})\|_F \leq \varepsilon_{\text{pod}} \|\mathbf{S}_u\|_F, \quad (7)$$

where  $\mathbf{S}_u = [\mathbf{S}_u^{\text{old}}, \mathbf{S}_u^{\text{new}}] \in \mathbb{R}^{N_{\text{dof}} \times N_s}$ ,  $\mathbf{S}_{u_{\text{off}}} = [\mathbf{S}_{u_{\text{off}}}^{\text{old}}, \mathbf{S}_{u_{\text{off}}}^{\text{new}}] \in \mathbb{R}^{N_{\text{dof}} \times N_s}$ , and  $N_s = N_{\text{sold}} + N_{\text{snew}}$ . For this purpose, we consider the approach described in [24], which relies on the compressed snapshot matrix

$$\hat{\mathbf{S}} = \begin{bmatrix} \hat{\mathbf{S}}_{\text{old}} & \hat{\mathbf{S}}_{\text{new}} \end{bmatrix} \in \mathbb{R}^{N_{\text{dof}} \times N_s}.$$

Here,  $\widehat{\mathbf{S}}_{\text{old}} = \mathbf{\Phi}_{\text{old}} \mathbf{\Sigma}_{\text{old}} \mathbf{\Psi}_{\text{old}}^T \in \mathbb{R}^{N_{\text{dof}} \times N_{\text{sold}}}$  (resp.  $\widehat{\mathbf{S}}_{\text{new}} = \mathbf{\Phi}_{\text{new}} \mathbf{\Sigma}_{\text{new}} \mathbf{\Psi}_{\text{new}}^T \in \mathbb{R}^{N_{\text{dof}} \times N_{\text{snew}}}$ ) denotes the truncated SVD of  $\mathbf{S}_{\text{u}}^{\text{old}} - \mathbf{S}_{\text{uoff}}^{\text{old}}$  (resp.  $\mathbf{S}_{\text{u}}^{\text{new}} - \mathbf{S}_{\text{uoff}}^{\text{new}}$ ), where  $\mathbf{\Phi}_{\text{old}} \in \mathbb{R}^{N_{\text{dof}} \times N_{\widehat{v}_{\text{old}}}}$  (resp.  $\mathbf{\Phi}_{\text{new}} \in \mathbb{R}^{N_{\text{dof}} \times N_{\widehat{v}_{\text{new}}}}$ ) is unitary,  $\mathbf{\Sigma}_{\text{old}} \in \mathbb{R}^{N_{\widehat{v}_{\text{old}}} \times N_{\widehat{v}_{\text{old}}}}$  (resp.  $\mathbf{\Sigma}_{\text{new}} \in \mathbb{R}^{N_{\widehat{v}_{\text{new}}} \times N_{\widehat{v}_{\text{new}}}}$ ) is diagonal,  $\mathbf{\Psi}_{\text{old}} \in \mathbb{R}^{N_{\text{sold}} \times N_{\widehat{v}_{\text{old}}}}$  (resp.  $\mathbf{\Psi}_{\text{new}} \in \mathbb{R}^{N_{\text{snew}} \times N_{\widehat{v}_{\text{new}}}}$ ) is unitary, and the rank  $N_{\widehat{v}_{\text{old}}}$  (resp.  $N_{\widehat{v}_{\text{new}}}$ ) is chosen so that the relative error is less than a user-defined tolerance  $\varepsilon_1 \in [0, 1]$ :

$$\left\| (\mathbf{S}_{\text{u}}^{\text{old}} - \mathbf{S}_{\text{uoff}}^{\text{old}}) - \widehat{\mathbf{S}}_{\text{old}} \right\|_F \leq \varepsilon_1 \|\mathbf{S}_{\text{u}}^{\text{old}}\|_F \quad (8a)$$

$$\left( \text{resp. } \left\| (\mathbf{S}_{\text{u}}^{\text{new}} - \mathbf{S}_{\text{uoff}}^{\text{new}}) - \widehat{\mathbf{S}}_{\text{new}} \right\|_F \leq \varepsilon_1 \|\mathbf{S}_{\text{u}}^{\text{new}}\|_F \right). \quad (8b)$$

The reduced basis  $\mathbf{V}$  is then computed by POD from the compressed snapshot matrix  $\widehat{\mathbf{S}}$  using the user-defined tolerance  $\varepsilon_2 \in [0, 1]$ :

$$\left\| \widehat{\mathbf{S}} - \mathbf{V} \mathbf{V}^T \widehat{\mathbf{S}} \right\|_F \leq \varepsilon_2 \|\mathbf{S}_{\text{u}}\|_F. \quad (9)$$

The advantage of this approach is twofold. On the one hand, only the low-rank representation of  $\widehat{\mathbf{S}}$  needs to be stored in memory, rather than the full matrices  $(\mathbf{S}_{\text{u}}, \mathbf{S}_{\text{uoff}})$ , which would have been computationally prohibitive. On the other hand, the resulting reduced basis  $\mathbf{V}$  verifies

$$\left\| \mathbf{S}_{\text{u}} - \mathbf{S}_{\text{uoff}} - \mathbf{V} \mathbf{V}^T (\mathbf{S}_{\text{u}} - \mathbf{S}_{\text{uoff}}) \right\|_F \leq \sqrt{\varepsilon_1^2 + \varepsilon_2^2} \|\mathbf{S}_{\text{u}}\|_F \quad (10)$$

according to Proposition 1, and we set  $\varepsilon_1 = \varepsilon_2 = \frac{\varepsilon_{\text{pod}}}{\sqrt{2}}$  in the following to ensure that equation (7) holds.

Moreover, we propose in this work to combine this approach with the fast SVD update of Brand [25] in order to reduce the computational cost of updating the reduced basis. The main idea is to progressively update the low-rank representation of  $\widehat{\mathbf{S}}$  with the new solution snapshots, instead of computing the SVD from scratch. To this end, let  $\widehat{\mathbf{S}} = \widehat{\mathbf{\Phi}} \widehat{\mathbf{\Sigma}} \widehat{\mathbf{\Psi}}$  be the SVD of  $\widehat{\mathbf{S}}$ . We introduce the extended reduced basis  $\widehat{\mathbf{V}} \in \mathbb{R}^{N_{\text{dof}} \times N_{\widehat{v}}}$  defined as the left singular vectors of  $\widehat{\mathbf{S}}$  (i.e.,  $\widehat{\mathbf{V}} = \widehat{\mathbf{\Phi}}$ ), where the first  $N_v$  columns of  $\widehat{\mathbf{V}}$  correspond to the reduced basis  $\mathbf{V}$ , while the  $N_{\widehat{v}} - N_v$  additional vectors allow to ensure that equation (7) holds without the need to store the snapshot databases  $(\mathbf{S}_{\text{u}}, \mathbf{S}_{\text{uoff}})$ . In Section 2.2.1, the extended reduced basis  $\widehat{\mathbf{V}}$  and associated singular values  $\widehat{\mathbf{\Sigma}}$  are initialized from the truncated SVD of  $\mathbf{S}_{\text{u}} - \mathbf{S}_{\text{uoff}}$  using the tolerance  $\varepsilon_1$ . Now, assume that  $\widehat{\mathbf{V}}_{\text{old}}$  and  $\widehat{\mathbf{\Sigma}}_{\text{old}}$  are known

and new snapshots ( $\mathbf{S}_u^{\text{new}}, \mathbf{S}_{u\text{off}}^{\text{new}}$ ) become available. Algorithm 3 presents the update of  $(\widehat{\mathbf{V}}, \widehat{\mathbf{\Sigma}})$  and, as a result, of  $(\mathbf{V}, \mathbf{\Sigma})$ . In Line 1,  $\widehat{\mathbf{V}}_{\text{new}}$  and  $\widehat{\mathbf{\Sigma}}_{\text{new}}$  are computed from the truncated SVD of  $\mathbf{S}_u^{\text{new}} - \mathbf{S}_{u\text{off}}^{\text{new}}$  using the tolerance  $\varepsilon_1$ . Next,  $\widehat{\mathbf{V}}$  and  $\widehat{\mathbf{\Sigma}}$  are efficiently updated in Lines 2-6 using the fast SVD update of Brand. If the matrix  $\widehat{\mathbf{V}}$  is not unitary due to numerical error, then a re-orthogonalization step can be performed in Line 7. Finally, the first  $N_v$  columns (resp. rows and columns) of  $\widehat{\mathbf{V}}$  (resp.  $\widehat{\mathbf{\Sigma}}$ ) are extracted in Line 8-11 to obtain  $\mathbf{V}$  (resp.  $\mathbf{\Sigma}$ ). In this way, the right singular vectors of  $\widehat{\mathbf{S}}$  do not need to be computed or stored, and only the left singular vectors  $\widehat{\mathbf{V}}$  and singular values  $\widehat{\mathbf{\Sigma}}$  are updated to save computational cost. In addition, the computational complexity of Algorithm 3 is dominated by the SVD of a  $N_{\text{dof}} \times N_{\text{snew}}$  matrix and the QR factorization of a  $N_{\text{dof}} \times N_{\widehat{v}}$  matrix, which can be efficiently computed using standard algorithms in sequential or parallel.

---

**Algorithm 3** Update of the reduced basis

---

**Input:**  $\widehat{\mathbf{V}}_{\text{old}} \in \mathbb{R}^{N_{\text{dof}} \times N_{\widehat{v}}}$ ,  $\widehat{\mathbf{\Sigma}}_{\text{old}} \in \mathbb{R}^{N_{\widehat{v}} \times N_{\widehat{v}}}$ ,  $\mathbf{S}_u^{\text{new}} \in \mathbb{R}^{N_{\text{dof}} \times N_{\text{snew}}}$ ,  $\mathbf{S}_{u\text{off}}^{\text{new}} \in \mathbb{R}^{N_{\text{dof}} \times N_{\text{snew}}}$ ,  $\varepsilon_1 \in [0, 1]$ ,  $\varepsilon_2 \in [0, 1]$ , and  $C_{\|\mathbf{S}_u^{\text{old}}\|_F^2} \geq 0$ .

**Output:**  $\mathbf{V} \in \mathbb{R}^{N_{\text{dof}} \times N_v}$ ,  $\mathbf{\Sigma} \in \mathbb{R}^{N_v \times N_v}$ ,  $\widehat{\mathbf{V}} \in \mathbb{R}^{N_{\text{dof}} \times \widehat{N}_v}$ ,  $\widehat{\mathbf{\Sigma}} \in \mathbb{R}^{\widehat{N}_v \times \widehat{N}_v}$ , and  $C_{\|\mathbf{S}_u\|_F^2} \geq 0$ .

- 1: Compute  $\widehat{\mathbf{V}}_{\text{new}}$  and  $\widehat{\mathbf{\Sigma}}_{\text{new}}$  by POD from  $(\mathbf{S}_u^{\text{new}}, \mathbf{S}_{u\text{off}}^{\text{new}})$  using the tolerance  $\varepsilon_1$ ;
  - 2: Set  $\mathbf{M} = \widehat{\mathbf{V}}_{\text{new}} - \widehat{\mathbf{V}}_{\text{old}} \widehat{\mathbf{V}}_{\text{old}}^T \widehat{\mathbf{V}}_{\text{new}}$ ;
  - 3: Compute the QR-decomposition of  $\mathbf{M} = \mathbf{Q}_M \mathbf{R}_M$ ;
  - 4: Set  $\mathbf{K} = \begin{bmatrix} \widehat{\mathbf{\Sigma}}_{\text{old}} & \widehat{\mathbf{V}}_{\text{old}}^T \widehat{\mathbf{V}}_{\text{new}} \widehat{\mathbf{\Sigma}}_{\text{new}} \\ \mathbf{0} & \mathbf{R}_M \widehat{\mathbf{\Sigma}}_{\text{new}} \end{bmatrix}$ ;
  - 5: Compute the SVD of  $\mathbf{K} = \mathbf{\Phi}_K \mathbf{\Sigma}_K \mathbf{\Psi}_K^T$ ;
  - 6: Set  $\widehat{\mathbf{V}} = \begin{bmatrix} \widehat{\mathbf{V}}_{\text{old}} & \mathbf{Q}_M \end{bmatrix} \mathbf{\Phi}_K$  and  $\widehat{\mathbf{\Sigma}} = \mathbf{\Sigma}_K$ ;
  - 7: *Optional:* orthonormalize  $\widehat{\mathbf{V}}$  and modify  $\widehat{\mathbf{\Sigma}}$  using Algorithm 4;
  - 8: Set  $C_{\|\mathbf{S}_u\|_F^2} = C_{\|\mathbf{S}_u^{\text{old}}\|_F^2} + \|\mathbf{S}_u^{\text{new}}\|_F^2$ ;
  - 9: Find the smallest integer  $N_v$  such that  $\sum_{i=N_v+1}^{N_{\widehat{v}}} \widehat{\sigma}_i^2 \leq \varepsilon_2^2 C_{\|\mathbf{S}_u\|_F^2}$ ;
  - 10: Set  $\mathbf{V}$  as the first  $N_v$  columns of  $\widehat{\mathbf{V}}$ ;
  - 11: Set  $\mathbf{\Sigma}$  as the first  $N_v$  rows and columns of  $\widehat{\mathbf{\Sigma}}$ .
-

---

**Algorithm 4** Orthonormalization process

---

**Input:**  $\widehat{\mathbf{V}} \in \mathbb{R}^{N_{\text{dof}} \times N_{\widehat{\mathbf{v}}}}$ , and  $\widehat{\mathbf{\Sigma}} \in \mathbb{R}^{N_{\widehat{\mathbf{v}}} \times N_{\widehat{\mathbf{v}}}}$ .

**Output:**  $\widehat{\mathbf{V}}_{\text{orth}} \in \mathbb{R}^{N_{\text{dof}} \times N_{\widehat{\mathbf{v}}}}$ , and  $\widehat{\mathbf{\Sigma}}_{\text{orth}} \in \mathbb{R}^{N_{\widehat{\mathbf{v}}} \times N_{\widehat{\mathbf{v}}}}$ .

- 1: Compute the QR-decomposition of  $\widehat{\mathbf{V}} = \mathbf{Q}_V \mathbf{R}_V$ ;
  - 2: Set  $\mathbf{N} = \mathbf{R}_V \widehat{\mathbf{\Sigma}}$ ;
  - 3: Compute the SVD of  $\mathbf{N} = \mathbf{\Phi}_N \mathbf{\Sigma}_N \mathbf{\Psi}_N^T$ ;
  - 4: Set  $\widehat{\mathbf{V}}_{\text{orth}} = \mathbf{Q}_V \mathbf{\Phi}_N$  and  $\widehat{\mathbf{\Sigma}}_{\text{orth}} = \mathbf{\Sigma}_N$ .
- 

#### 4.2. Update of the reduced mesh

Given new snapshots of the reduced-order quantities, the extended reduced basis  $\widehat{\mathbf{U}}_{\text{old}}$  and associated singular values  $\widehat{\mathbf{\Lambda}}_{\text{old}}$  are updated in the same way as  $\widehat{\mathbf{V}}_{\text{old}}$  and  $\widehat{\mathbf{\Sigma}}_{\text{old}}$  using Algorithm 3 in order to obtain  $(\widehat{\mathbf{U}}, \widehat{\mathbf{\Lambda}})$  and therefore  $(\mathbf{U}, \mathbf{\Lambda})$ . The weights  $\boldsymbol{\omega}$  are then solution to the NNLS problem (4) defined by  $(\mathbf{U}, \mathbf{\Lambda})$  as explained in Section 2.2.3.

#### 4.3. Update of the error indicator

The error indicator is finally updated in the same way as the ROM. Specifically, the reduced basis  $\mathbf{W}$  is updated from new snapshots of the error using the procedure presented in Section 4.1, and the reduced mesh is updated from new snapshots of the reduced-order quantities using the procedure described in Section 4.2.

## 5. Numerical experiments

The performance of the proposed methods is assessed for parametrized, time-dependent, nonlinear problems based on the 1D Burger's equation and 2D compressible Euler equations. The first experiment seeks to evaluate the performance of the hybrid model and the accuracy of the error indicator, while the second experiment aims at evaluating the reduction in the computational cost of the offline stage provided by the adaptive, hybrid approach.

### 5.1. One dimensional Burgers' equation

The first experiment focuses on the numerical simulation of a single hump in 1D. The fluid velocity  $u(x, t; \boldsymbol{\theta}) \in \mathbb{R}$  is governed by the Burgers' equation:

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2},$$



where  $x \in [-2, 3]$ ,  $t \in [0, 0.2]$ , and the input parameter  $\boldsymbol{\theta} = [\nu, A]$  corresponds to the diffusion coefficient  $\nu \in [0.1, 1]$  and a constant  $A \in [1, 5]$ . The initial and Dirichlet boundary conditions are supplied by the exact solution

$$u(x, t; \boldsymbol{\theta}) = \sqrt{\frac{\nu}{t + \epsilon}} \left[ \frac{\left( e^{\frac{A}{2\nu}} - 1 \right) e^{\frac{-x^2}{4\nu(t+\epsilon)}}}{\sqrt{\pi} + \frac{\sqrt{\pi}}{2} \left( e^{\frac{A}{2\nu}} - 1 \right) \operatorname{erfc}\left(\frac{x}{\sqrt{4\nu(t+\epsilon)}}\right)} \right],$$

where  $\operatorname{erfc}(x)$  is the complimentary error function, and  $\epsilon = 0.05$ .

This equation is then discretized in the HDM by a second-order finite difference method. The time-interval is divided into  $N_t = 200$  uniform subintervals using the fixed time-step size  $\Delta t = 10^{-3}$ , and the spatial domain is equally partitioned by 500 interior grid points, leading to  $N_{\text{dof}} = 500$  degrees of freedom (DOFs). In Figure 1, we show solution snapshots computed by the HDM for different parameters  $\boldsymbol{\theta}$ .

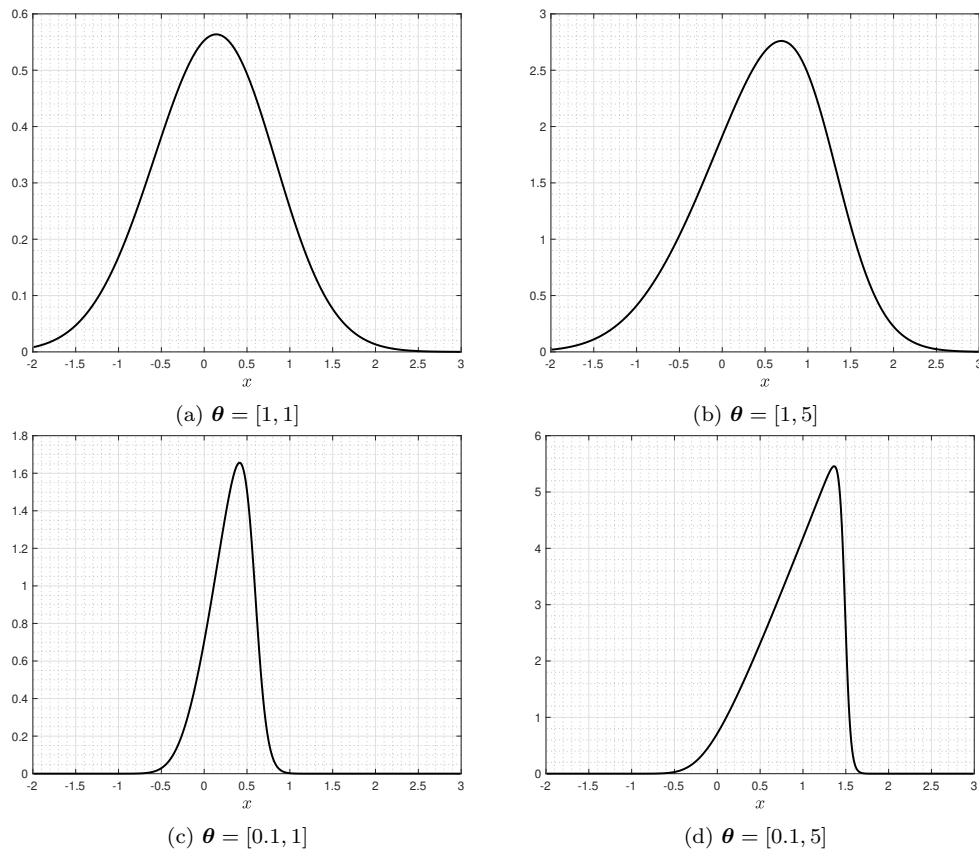


Figure 1: High-dimensional solution at final time for different parameters  $\theta$

### 5.1.1. Performance of the hybrid model

The performance of the hybrid model is evaluated for predicting the solution corresponding to  $\theta = [0.1, 5]$ . Compared to a reproductive test, the ROM is constructed from solution snapshots collected in the time-interval  $[0, 0.05] \cup [0.1, 0.2]$ , and therefore the ROM has to predict the solution for  $t \in (0.05, 0.1)$ . In Figure 2a, we compare the accuracy of four ROMs corresponding to different tolerances  $\varepsilon_{\text{pod}} \in \{10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\}$ . For constructing these ROMs, solution snapshots are collected every time-step, leading to  $N_s = 150$  snapshots, and the hyper-reduction tolerance is chosen as  $\varepsilon_{\text{ecsw}} = 10^{-6}$  to provide sufficient accuracy. The reader can see that the error increases for  $t > 0.05$  since no snapshots were collected. Moreover, this error remains large even for small value of  $\varepsilon_{\text{pod}}$ , which means that the corresponding solution cannot be accurately approximated using only the snapshots

available.

We now consider the hybrid model, where the underlying ROM is built using the tolerances  $\varepsilon_{\text{pod}} = 10^{-3}$  and  $\varepsilon_{\text{ecsw}} = 10^{-6}$ , leading to  $N_v = 19$  basis vectors and  $N_p = 62$  points. Moreover, the size of the reduced basis and reduced mesh in the error indicator is the same than that of this ROM for simplicity. Figure 2b presents the accuracy the hybrid model for different tolerance  $\varepsilon_{\text{err}} \in \{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$ . Compared to the previous ROMs, the solution is accurately approximated for every time-steps since the relative error remains generally smaller than the prescribed tolerance  $\varepsilon_{\text{err}}$ . In particular, the relative error is sometimes larger than  $\varepsilon_{\text{err}}$  since the error indicator underestimates the error norm as illustrated in Figure 3. Furthermore, for  $\varepsilon_{\text{err}} = 10^{-2}$ , the high-dimension system (1) is solved only in the time-interval  $(0.05, 0.1)$ , which enables a speedup factor of 1.58 with respect to the HDM, while maintaining accurate approximations as illustrated in Figure 4.

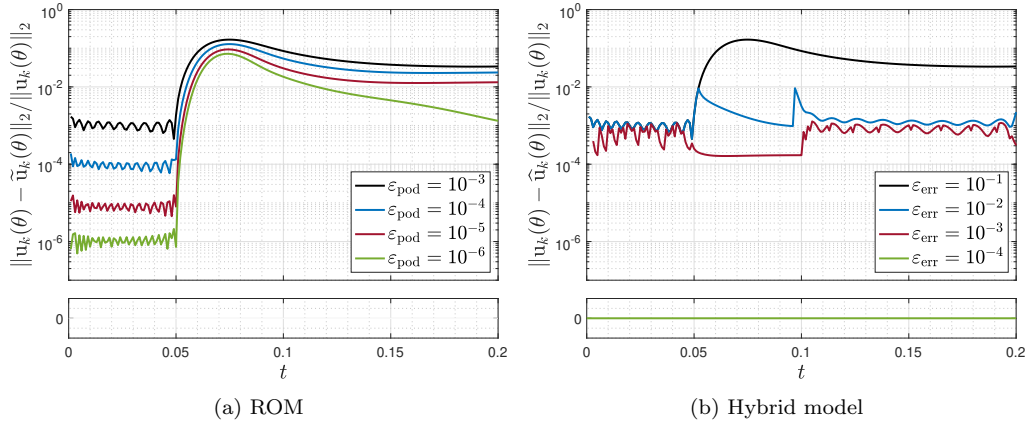


Figure 2: Accuracy of the ROM and hybrid model over time. On the left, we plot the relative error of ROMs constructed using different tolerances  $\varepsilon_{\text{pod}}$ , while the relative error of the hybrid model for different tolerances  $\varepsilon_{\text{err}}$  is plotted on the right

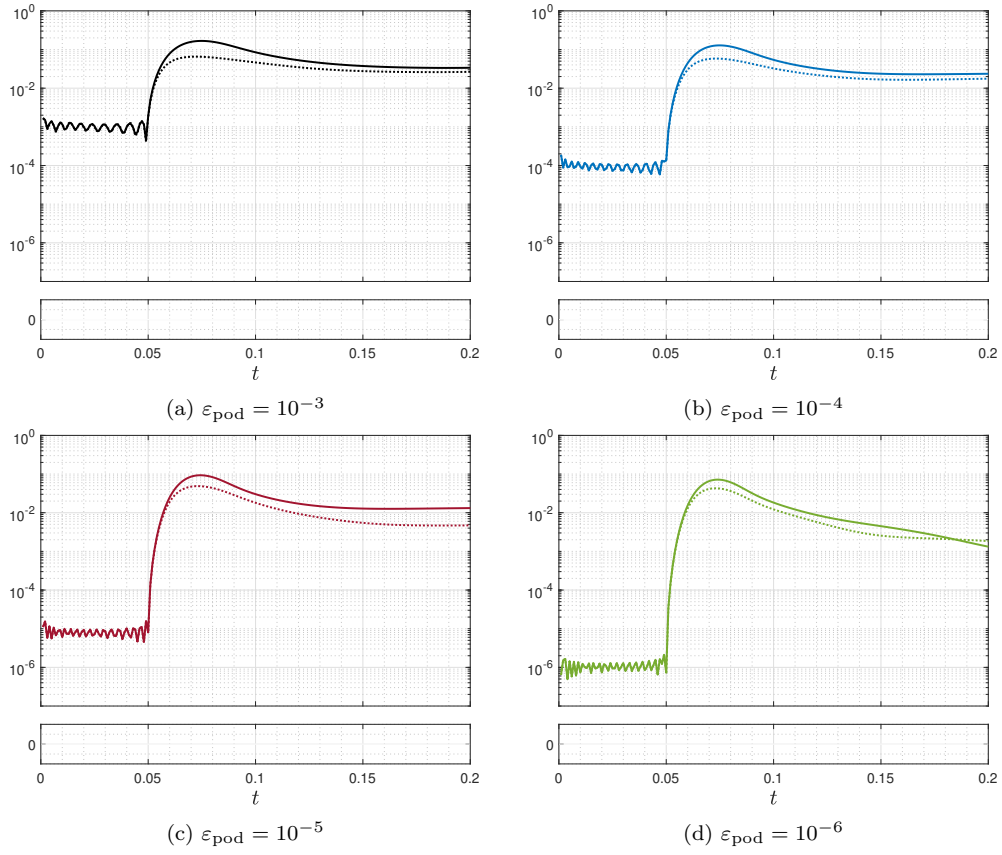


Figure 3: Time-history of the relative error  $\|\mathbf{e}_k(\boldsymbol{\theta})\|_2 / \|\mathbf{u}_k(\boldsymbol{\theta})\|_2$  (solid line) and its estimation  $\|\tilde{\mathbf{e}}_k(\boldsymbol{\theta})\|_2 / \|\tilde{\mathbf{e}}_k(\boldsymbol{\theta}) + \tilde{\mathbf{u}}_k(\boldsymbol{\theta})\|_2$  (dotted line) by the error indicator

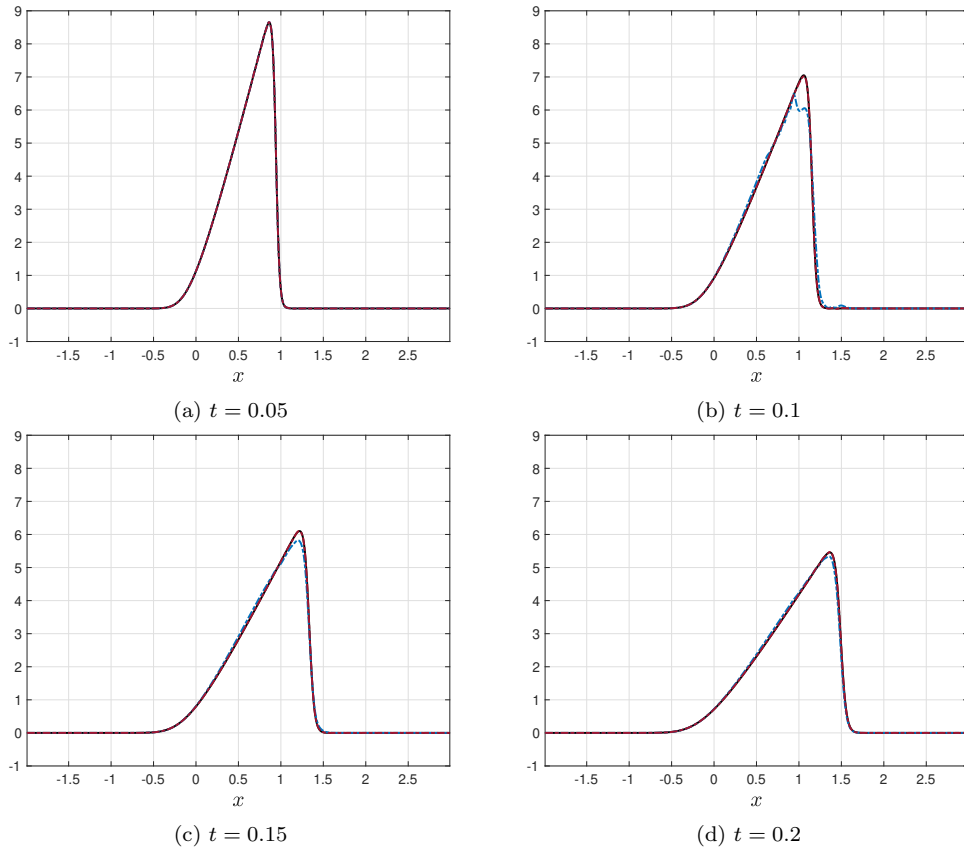


Figure 4: Solution snapshots of the HDM (black), ROM for  $\varepsilon_{\text{pod}} = 10^{-3}$  (blue), and hybrid model for  $\varepsilon_{\text{pod}} = 10^{-3}$  and  $\varepsilon_{\text{err}} = 10^{-2}$  (red)

### 5.1.2. Accuracy of the error indicator

The accuracy of the proposed error indicator is further illustrated for predicting the error corresponding to different out-of-sample parameters  $\theta$ . To this end, the ROM and associated error indicator are constructed using snapshots collected every time-step at the 5 training parameter instances shown in Figure 5, leading to  $N_s = 200 \times 5 = 1000$  snapshots. The reduced basis is then built using the tolerance  $\varepsilon_{\text{pod}} = 10^{-3}$ , which yields to  $N_v = 32$  vectors, and the reduced mesh is extracted using the tolerance  $\varepsilon_{\text{ecsw}} = 10^{-6}$ , leading to  $N_p = 105$  points. In the error indicator, the size of the reduced basis and reduced mesh is the same than that of the ROM to provide sufficient accuracy.

Figure 6 presents the relative error of the resulting ROM and its esti-

mation by the error indicator for the 20 out-of-sample parameters shown in Figure 5. In particular, there are two risks associated with error estimation. First, if the error norm  $\|\mathbf{e}_k(\boldsymbol{\theta})\|_2$  is less than the error tolerance  $\varepsilon_{\text{err}}$ , but the error indicator  $\|\tilde{\mathbf{e}}_k(\boldsymbol{\theta})\|_2$  overestimates the error norm and is greater than the error tolerance, then the hybrid model will employ the HDM whereas the ROM was sufficient in term of accuracy, resulting in unnecessary additional computation time. Second, if the error norm is greater than the error tolerance, but the error indicator underestimates the error norm and is less than the error tolerance, then the HDM will not be used while the ROM is not sufficiently accurate, which can lead to inaccurate solutions. However, despite the coarse training grid used for constructing the error indicator, the estimation  $\|\tilde{\mathbf{e}}_k(\boldsymbol{\theta})\|_2$  accurately approximates the true error  $\|\mathbf{e}_k(\boldsymbol{\theta})\|_2$  for the different out-of-sample parameters. Furthermore, the predictions of the error indicator can be improved by enriching the database  $\mathbf{S}_e$  with new snapshots of the error and by increasing the dimension of the reduced basis  $\mathbf{W}$ .

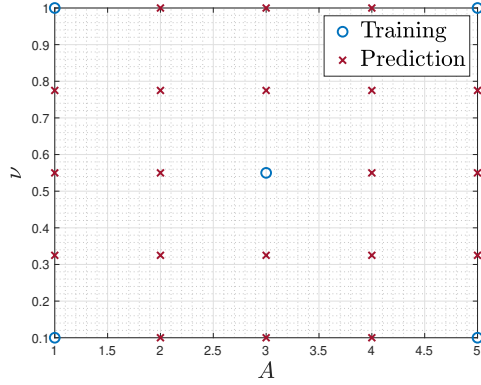


Figure 5: Sampling of the parameter domain  $\mathcal{D}$ . The training parameters are plotted in blue, while the out-of-sample parameters are plotted in red

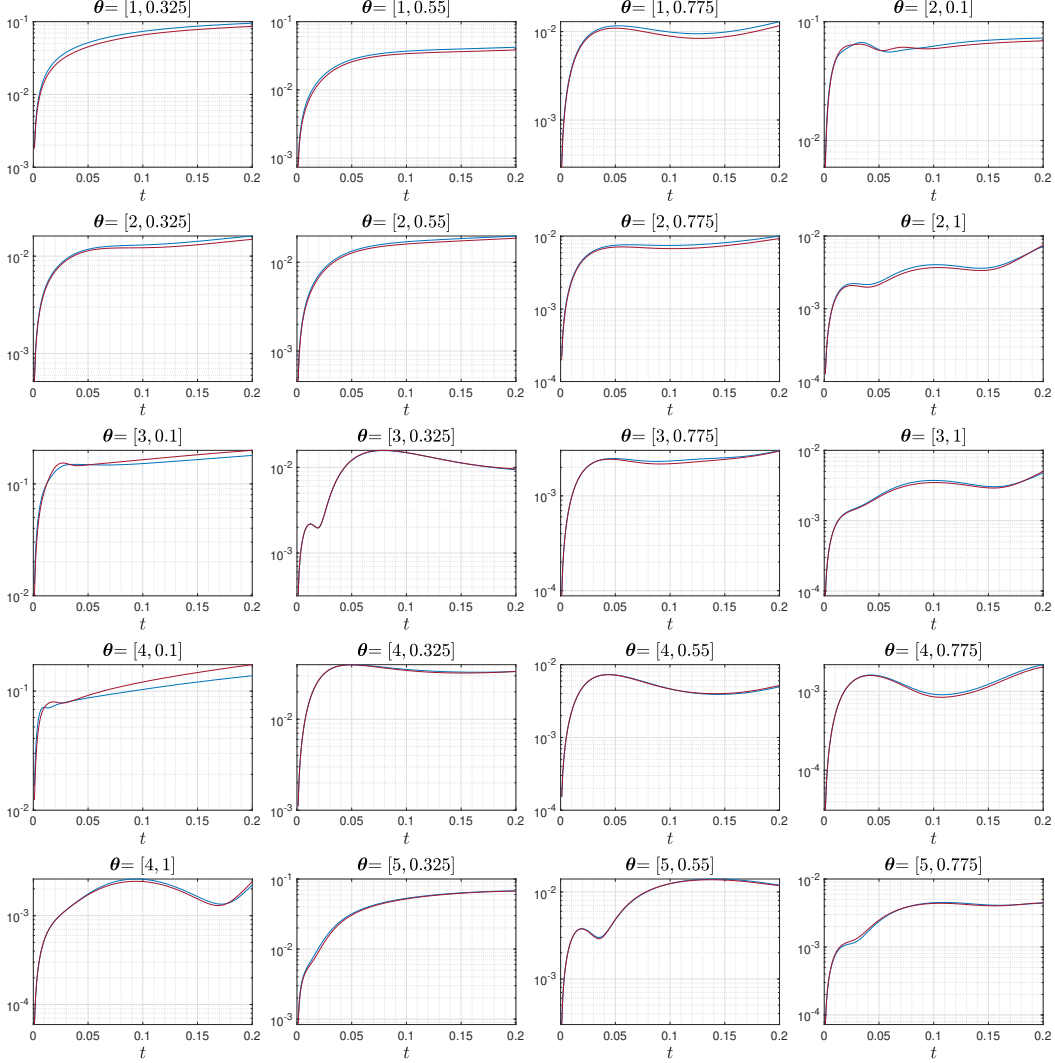


Figure 6: Time-history of the error norm  $\|\mathbf{e}_k(\boldsymbol{\theta})\|_2$  (in blue) and its estimation  $\|\tilde{\mathbf{e}}_k(\boldsymbol{\theta})\|_2$  (in red) for different out-of-sample parameters  $\boldsymbol{\theta}$

### 5.2. Two-dimensional compressible Euler equations

The second experiment concerns the numerical simulation of subsonic and transonic flows over a NACA 0012 airfoil in 2D. The input parameter  $\boldsymbol{\theta} = [M_\infty, \alpha]$  corresponds to the free-stream Mach number  $M_\infty \in [0.3, 0.8]$  and the angle of attack  $\alpha \in [0, 8]$ . The spatial domain is shown in Figure 7, and we consider the time-interval  $[0, 10]$ . The fluid density  $\rho(\mathbf{x}, t; \boldsymbol{\theta}) \geq 0$ , velocity  $\mathbf{u}(\mathbf{x}, t; \boldsymbol{\theta}) = [u(\mathbf{x}, t; \boldsymbol{\theta}), v(\mathbf{x}, t; \boldsymbol{\theta})] \in \mathbb{R}^2$ , and total energy  $E(\mathbf{x}, t; \boldsymbol{\theta}) \geq 0$  are

governed by the compressible Euler equations:

$$\begin{cases} \frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x} + \frac{\partial \rho v}{\partial y} = 0 \\ \frac{\partial \rho u}{\partial t} + \frac{\partial(\rho u^2 + p)}{\partial x} + \frac{\partial \rho uv}{\partial y} = 0 \\ \frac{\partial \rho v}{\partial t} + \frac{\partial \rho uv}{\partial x} + \frac{\partial(\rho v^2 + p)}{\partial y} = 0 \\ \frac{\partial E}{\partial t} + \frac{\partial u(E + p)}{\partial x} + \frac{\partial v(E + p)}{\partial y} = 0, \end{cases}$$

where  $p(\mathbf{x}, t; \boldsymbol{\theta}) \geq 0$  is the fluid pressure, given by the equation of state

$$p = (\gamma - 1) \left( E - \rho \frac{u^2 + v^2}{2} \right)$$

with  $\gamma$ , the specific heat ratio, taken as  $\gamma = 1.4$  in the following. The initial condition is a uniform flow at Mach  $M_\infty$ :

$$\rho_0 = \gamma, \quad u_0 = M_\infty, \quad v_0 = 0, \quad p_0 = 1. \quad (11)$$

The inflow and outflow boundary conditions are set to be a uniform flow at Mach  $M_\infty$  (11), and slip boundary conditions are applied at the airfoil surface.

The HDM is constructed using a first-order finite volume method equipped with the local Lax-Friedrichs flux in space and the implicit Euler scheme in time. The time-interval is divided into  $N_t = 1,000$  uniform subintervals using the fixed time-step size  $\Delta t = 10^{-2}$ , and the spatial domain is partitioned by 1,718 vertices and 2,300 triangular cells, leading to  $N_{\text{dof}} = 4 \times 2,300 = 9,200$  DOFs. In Figure 8, we show snapshots of the Mach number solution computed by the HDM for different parameters  $\boldsymbol{\theta}$ .



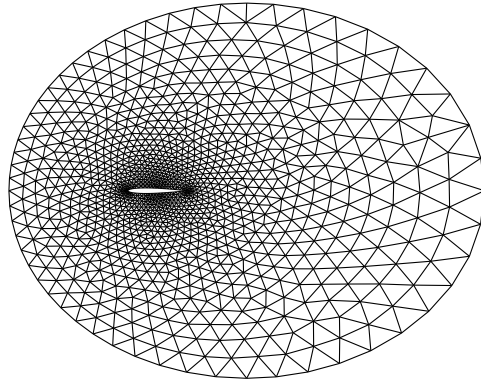


Figure 7: Computational domain in the reference configuration (i.e.,  $\alpha = 0$ ). To account for the variation in the angle of attack, the mesh is rotated through an angle  $\alpha$  about the trailing edge

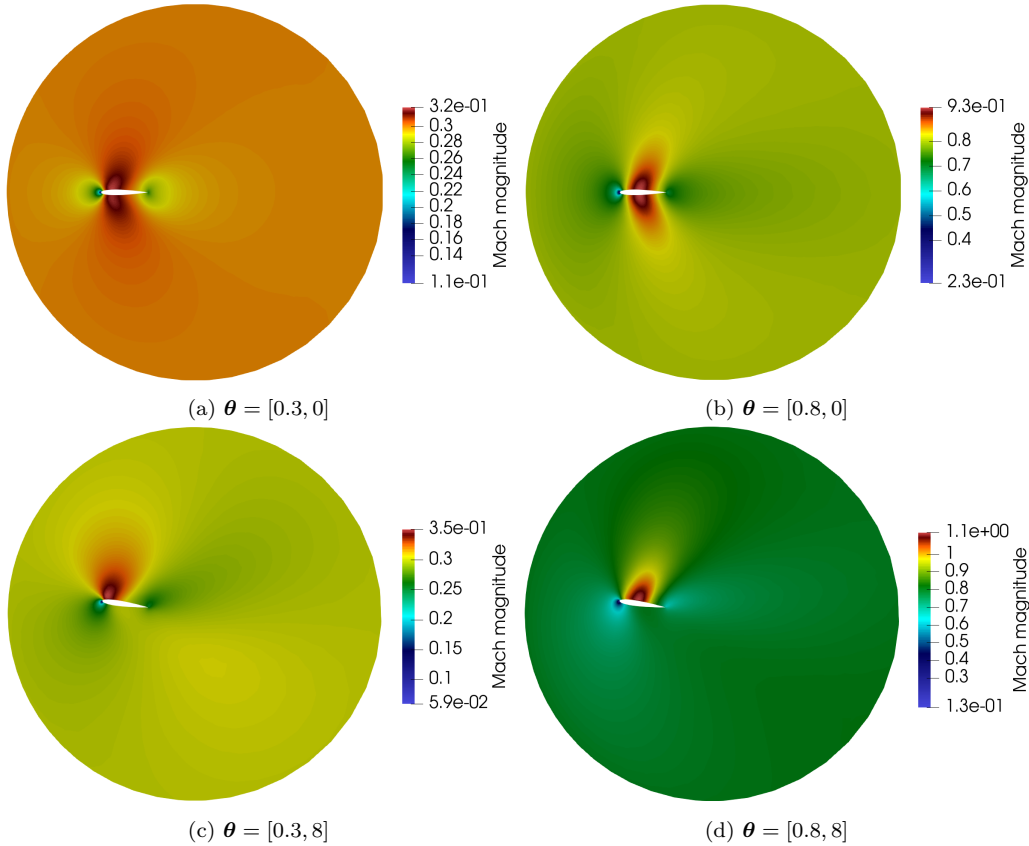


Figure 8: High-dimensional solution at final time for different parameters  $\theta$

### 5.2.1. Performance of the adaptive, hybrid approach

The reduction in the computational cost of the offline stage provided by the adaptive, hybrid model is assessed for sampling the solution at the 25 training parameter instances shown in Figure 9. To this end, we first compare the runtime of the hybrid model with that of the classical approach, where solution snapshots are computed using only the HDM. For initializing the hybrid model, solution snapshots are collected every 4 time-steps using the HDM at the 5 training parameter instances shown in Figure 9, leading to  $N_s = 250 \times 5 = 2,500$  snapshots. The reduced basis is then built using the tolerance  $\varepsilon_{\text{pod}} = 5 \times 10^{-3}$  and the reduced mesh is extracted using the tolerance  $\varepsilon_{\text{ecsw}} = 10^{-6}$ . In the error indicator, the size of the reduced basis and reduced mesh is taken as  $(2N_v, 2N_p)$  to provide sufficient accuracy. Furthermore, the error indicator is employed in the following to select iteratively

the solution to be sampled by identifying the training parameter associated with the maximum error.

When the classical approach is used for sampling solution snapshots, the offline stage takes 2.39 hours and results in  $N_v = 14$  basis vectors and  $N_p = 204$  points. When the hybrid model is employed, the offline stage lasts 44.53 minutes, which corresponds to 28.7 minutes for sampling the 5 initial parameters and 15.83 minutes for sampling the 20 other training parameters. Moreover, the dimension of the resulting reduced basis is  $N_v = 16$  and the size of the reduced mesh is  $N_p = 260$ . The hybrid model therefore provides a speedup factor of 3.22, and the dimension of the ROMs ( $N_v, N_p$ ) obtained using the adaptive approach is slightly higher than that obtained using the classical approach as shown in Figure 10.

Finally, Figure 11 presents the performance of the ROMs resulting from the classical and adaptive approaches. In particular, the accuracy is assessed using the relative space-time error:

$$\text{Relative space-time error} = \sqrt{\frac{\sum_{k=1}^{N_t} \|\mathbf{u}_k(\boldsymbol{\theta}) - \tilde{\mathbf{u}}_k(\boldsymbol{\theta})\|_2^2}{\sum_{k=1}^{N_t} \|\mathbf{u}_k(\boldsymbol{\theta})\|_2^2}} \times 100 (\%),$$

and the reduction in computational cost is evaluated using the relative runtime:

$$\text{Relative runtime} = \frac{\text{Runtime of the ROM}}{\text{Runtime of the HDM}} \times 100 (\%).$$

The reader can see that the accuracy of these ROMs is also almost the same for predicting the solution at the 16 out-of-sample parameters shown in Figure 9. Furthermore, the computational cost of the ROM resulting from the adaptive approach is slightly higher than that resulting from the classical approach, because the dimension ( $N_v, N_p$ ) of the latter is slightly lower.

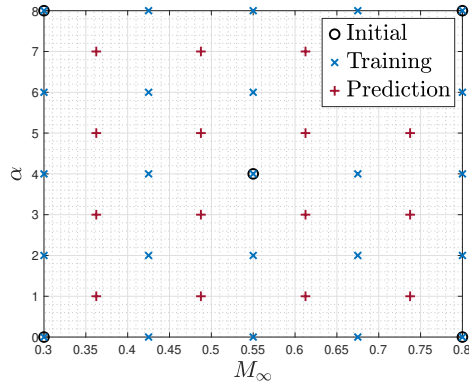


Figure 9: Sampling of the parameter domain  $\mathcal{D}$ . The parameters used to initialize the hybrid model are plotted in black, the training parameters are plotted in blue, and the out-of-sample parameters are plotted in red

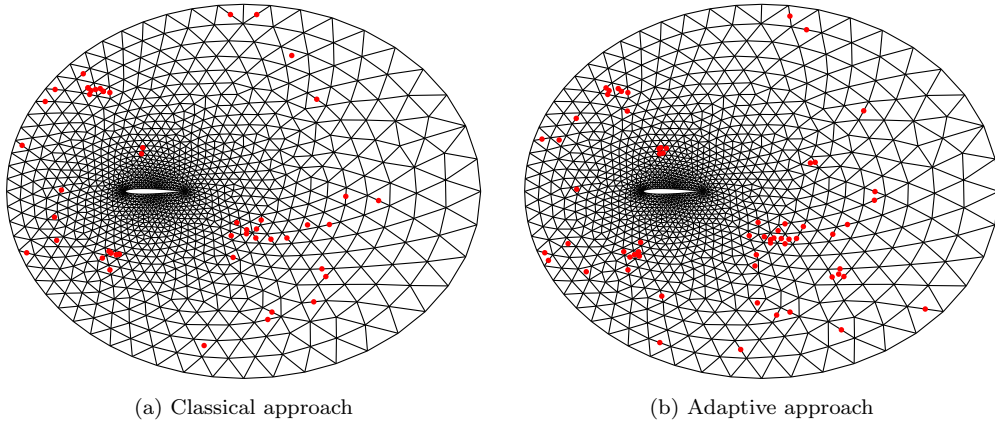


Figure 10: Comparison of the reduced meshes obtained using the classical and adaptive approaches

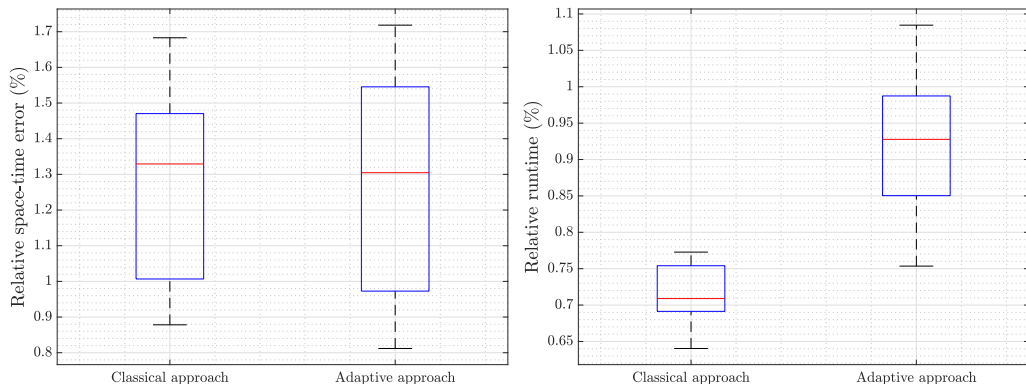


Figure 11: Performance of the ROMs resulting from the classical and adaptive approaches

## 6. Conclusion

In this paper, we have proposed an approach based on the principal directions of the reduced-order residual and reduced-order Jacobian matrix to improve the performance of the ECSW method. Then, we have presented a hybrid HDM/ROM strategy to accelerate numerical simulations while maintaining accurate approximations. In particular, we have developed an error indicator based on a reduced-order approximation of the error to determine when the ROM is not sufficiently accurate and the HDM needs to be solved. Finally, we have proposed an adaptive-extended version of the hybrid model to update the ROM with new solution snapshots.

The performance of the proposed methods has been evaluated for parametrized, time-dependent, nonlinear problems based on the 1D Burger’s equation and 2D compressible Euler equations. The results demonstrated the accuracy and robustness of the hybrid model, capable of controlling the estimated error by switching between the HDM and ROM on the fly, and the significant computational speedup factor delivered by the adaptive, hybrid approach to accelerate the offline stage of ROMs.

In perspective, the performance of the hybrid model could be further improved by using ROMs based on nonlinear manifolds [26, 27, 28]. In addition, the computational cost associated with the HDM solution could also be reduced by considering hybrid snapshots. The latter are generated by employing the HDM in regions where the ROM is not sufficiently accurate, and space-local ROMs [29, 30, 31] elsewhere. These topics will be addressed in future work.

## Appendix A. Construction of the approximation problem (3)

The objective of this section is to describe the construction of the approximation problem (3). Let the training time-parameter instances be  $\{(t_1^{\text{train}}, \boldsymbol{\theta}_1^{\text{train}}), \dots, (t_{N_s}^{\text{train}}, \boldsymbol{\theta}_{N_s}^{\text{train}})\}$ . We start to present different strategies for constructing  $\mathbf{C}$  and consequently  $\mathbf{d}$  since  $\mathbf{d} = \mathbf{C}\mathbf{1}$ .

**Residual.** The evaluation of the projected reduced-order residual  $\bar{\mathbf{J}}_k(\boldsymbol{\theta})^T \bar{\mathbf{r}}_k(\boldsymbol{\theta})$  leads to the approximation problem:

$$\mathbf{C}_1 \boldsymbol{\omega} \approx \mathbf{d}_1, \quad (\text{A.1})$$

where

$$\mathbf{C}_1 = \begin{bmatrix} \mathbf{C}_1^{(1)} \\ \vdots \\ \mathbf{C}_1^{(N_s)} \end{bmatrix} \in \mathbb{R}^{(N_s N_v) \times N_{\text{dof}}} \quad \text{and} \quad \mathbf{C}_1^{(k)} \in \mathbb{R}^{N_v \times N_{\text{dof}}}$$

with  $(\mathbf{C}_1^{(k)})_{i,j} = (\bar{\mathbf{J}}_k(\boldsymbol{\theta}))_{j,i} (\bar{\mathbf{r}}_k(\boldsymbol{\theta}))_j$  and  $(t_k, \boldsymbol{\theta}) = (t_k^{\text{train}}, \boldsymbol{\theta}_k^{\text{train}})$ .

**Jacobian.** In the same way, the evaluation of the projected reduced-order Jacobian matrix  $\bar{\mathbf{J}}_k(\boldsymbol{\theta})^T \bar{\mathbf{J}}_k(\boldsymbol{\theta})$  leads to the approximation problem:

$$\mathbf{C}_2 \boldsymbol{\omega} \approx \mathbf{d}_2, \quad (\text{A.2})$$

where

$$\mathbf{C}_2 = \begin{bmatrix} \mathbf{C}_2^{(1)} \\ \vdots \\ \mathbf{C}_2^{(N_s)} \end{bmatrix} \in \mathbb{R}^{(N_s N_v (N_v + 1) / 2) \times N_{\text{dof}}} \quad \text{and} \quad \mathbf{C}_2^{(k)} \in \mathbb{R}^{(N_v (N_v + 1) / 2) \times N_{\text{dof}}}$$

with  $(\mathbf{C}_2^{(k)})_{i_1 + i_2(i_2 - 1) / 2, j} = (\bar{\mathbf{J}}_k(\boldsymbol{\theta}))_{j, i_1} (\bar{\mathbf{J}}_k(\boldsymbol{\theta}))_{j, i_2}$  and  $(t_k, \boldsymbol{\theta}) = (t_k^{\text{train}}, \boldsymbol{\theta}_k^{\text{train}})$  for  $1 \leq i_1 \leq i_2 \leq N_v$ . Note that the symmetry of the problem has been exploited to remove the duplicate rows in  $\mathbf{C}_2$ .

**Residual+Jacobian.** Combining the contributions of (A.1) and (A.2) leads to the approximation problem:

$$\mathbf{C}_3 \boldsymbol{\omega} \approx \mathbf{d}_3, \quad (\text{A.3})$$

where

$$\mathbf{C}_3 = \begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \end{bmatrix} \in \mathbb{R}^{(N_s N_v (N_v + 3) / 2) \times N_{\text{dof}}}.$$

**Principal directions.** In practice, the number of rows in  $\mathbf{C}_3$  can rapidly become very large, which is problematic for solving the NNLS problem (4). In order to drastically reduce the number of rows, we replace  $\mathbf{C}_3$  by its principal directions. Let  $\mathbf{C}_3 \approx \mathbf{\Phi}_C \mathbf{\Sigma}_C \mathbf{\Psi}_C^T$  be the truncated SVD of  $\mathbf{C}_3$ , where  $\mathbf{\Phi}_C \in \mathbb{R}^{(N_s N_v (N_v + 3)/2) \times N_c}$  is unitary,  $\mathbf{\Sigma}_C \in \mathbb{R}^{N_c \times N_c}$  is diagonal,  $\mathbf{\Psi}_C \in \mathbb{R}^{N_{\text{dof}} \times N_c}$  is unitary, and the rank  $N_c$  is defined as the smallest integer such that the relative approximation error is less than the tolerance  $\varepsilon_{\text{pod}}$  (i.e.,  $\|\mathbf{C}_3 - \mathbf{\Phi}_C \mathbf{\Sigma}_C \mathbf{\Psi}_C^T\|_F \leq \varepsilon_{\text{pod}} \|\mathbf{C}_3\|_F$ ). The approximation problem (A.3) becomes

$$\mathbf{C}_4 \boldsymbol{\omega} \approx \mathbf{d}_4, \quad (\text{A.4})$$

where

$$\mathbf{C}_4 = \mathbf{\Psi}_C^T \in \mathbb{R}^{N_c \times N_{\text{dof}}} \quad \text{and} \quad N_c \ll N_s N_v (N_v + 3)/2.$$

**Weighted principal directions.** In the approximation problem (A.4), the principal directions contribute equally to the approximation error, which may not be representative of the original approximation problem (A.3). For this reason, we introduce a weighting of the principal directions to approximately recover the original problem (A.3):

$$\begin{aligned} \|\mathbf{C}_3 \boldsymbol{\omega} - \mathbf{d}_3\|_2 &\approx \|\mathbf{\Phi}_C \mathbf{\Sigma}_C \mathbf{\Psi}_C^T \boldsymbol{\omega} - \mathbf{d}_3\|_2 \\ &= \|\mathbf{\Phi}_C \mathbf{\Sigma}_C (\mathbf{C}_4 \boldsymbol{\omega} - \mathbf{d}_4)\|_2 \\ &= \|\mathbf{\Sigma}_C (\mathbf{C}_4 \boldsymbol{\omega} - \mathbf{d}_4)\|_2. \end{aligned}$$

In this way, the approximation problem (A.4) becomes

$$\mathbf{C}_5 \boldsymbol{\omega} \approx \mathbf{d}_5, \quad (\text{A.5})$$

where

$$\mathbf{C}_5 = \mathbf{\Sigma}_C \mathbf{C}_4 \in \mathbb{R}^{N_c \times N_{\text{dof}}}.$$

**Numerical results.** The different strategies for constructing  $\mathbf{C}$  are now evaluated for the application presented in Section 5.2. In each case, we report the size of the matrix  $\mathbf{C}$ . Moreover, Figure A.12 shows the runtime for solving the NNLS problem (4) and the accuracy of the ROM associated with the resulting weights  $\boldsymbol{\omega}$ , as a function of the reduced mesh size.

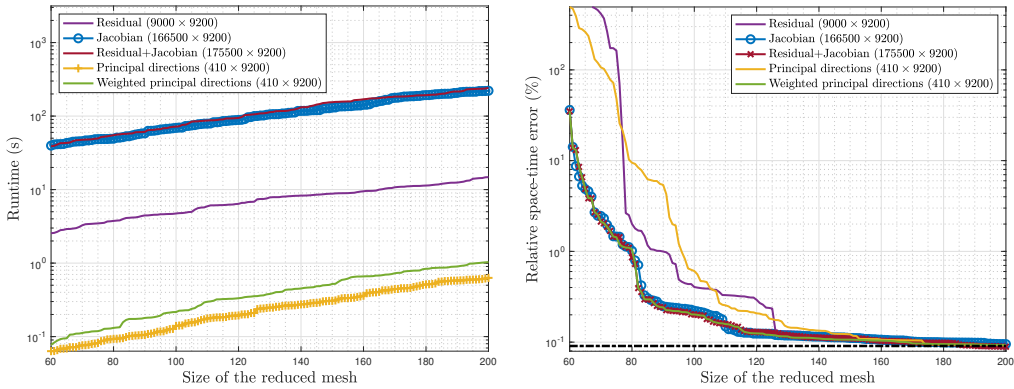


Figure A.12: Results for the application presented in Section 5.2. We consider the reproduction of the solution corresponding to the input parameter  $\theta = [0.8, 8]$ . Solution snapshots are collected every 4 time-steps, leading to  $N_s = 250$  snapshots. The reduced basis is built using the tolerance  $\varepsilon_{\text{pod}} = 10^{-3}$ , which yields to  $N_v = 36$  basis vectors.

The runtime of the "principal directions" and "weighted principal directions" approaches are almost the same and are significantly lower than that of the other approaches, as the number of rows of the corresponding matrix  $\mathbf{C}$  is drastically smaller. In addition, the error of the "jacobian", "residual+jacobian", and "weighted principal direction" approaches are about the same and are generally lower than that of the other approaches, since the "residual+jacobian" approach contains all the available information for computing the weights  $\omega$ . For these reasons, the "weighted principal direction" approach is used in this work as it provides the most accurate ROM, while drastically reducing the runtime to solve the NNLS problem (4).

## Appendix B. Proof of error bound (10)

**Proposition 1.** *Assume that equations (8) and (9) hold. Then, the reduced basis  $\mathbf{V}$  verifies the error bound (10).*



*Proof.* Given (8a), we first prove the inequality

$$\begin{aligned}
& \left\| \mathbf{S}_u^{\text{old}} - \mathbf{S}_{u_{\text{off}}}^{\text{old}} - \mathbf{V}\mathbf{V}^T(\mathbf{S}_u^{\text{old}} - \mathbf{S}_{u_{\text{off}}}^{\text{old}}) \right\|_F^2 \\
&= \left\| \mathbf{S}_u^{\text{old}} - \mathbf{S}_{u_{\text{off}}}^{\text{old}} - \widehat{\mathbf{S}}_{\text{old}} + \widehat{\mathbf{S}}_{\text{old}} - \mathbf{V}\mathbf{V}^T(\mathbf{S}_u^{\text{old}} - \mathbf{S}_{u_{\text{off}}}^{\text{old}}) + \mathbf{V}\mathbf{V}^T\widehat{\mathbf{S}}_{\text{old}} - \mathbf{V}\mathbf{V}^T\widehat{\mathbf{S}}_{\text{old}} \right\|_F^2 \\
&\leq \left\| (\mathbf{I} - \mathbf{V}\mathbf{V}^T)(\mathbf{S}_u^{\text{old}} - \mathbf{S}_{u_{\text{off}}}^{\text{old}} - \widehat{\mathbf{S}}_{\text{old}}) \right\|_F^2 + \left\| \widehat{\mathbf{S}}_{\text{old}} - \mathbf{V}\mathbf{V}^T\widehat{\mathbf{S}}_{\text{old}} \right\|_F^2 \\
&\leq \left\| \mathbf{I} - \mathbf{V}\mathbf{V}^T \right\|_2^2 \left\| \mathbf{S}_u^{\text{old}} - \mathbf{S}_{u_{\text{off}}}^{\text{old}} - \widehat{\mathbf{S}}_{\text{old}} \right\|_F^2 + \left\| \widehat{\mathbf{S}}_{\text{old}} - \mathbf{V}\mathbf{V}^T\widehat{\mathbf{S}}_{\text{old}} \right\|_F^2 \\
&\leq \left\| \mathbf{S}_u^{\text{old}} - \mathbf{S}_{u_{\text{off}}}^{\text{old}} - \widehat{\mathbf{S}}_{\text{old}} \right\|_F^2 + \left\| \widehat{\mathbf{S}}_{\text{old}} - \mathbf{V}\mathbf{V}^T\widehat{\mathbf{S}}_{\text{old}} \right\|_F^2 \\
&\leq \varepsilon_1^2 \left\| \mathbf{S}_u^{\text{old}} \right\|_F^2 + \left\| \widehat{\mathbf{S}}_{\text{old}} - \mathbf{V}\mathbf{V}^T\widehat{\mathbf{S}}_{\text{old}} \right\|_F^2,
\end{aligned}$$

since  $\left\| \mathbf{I} - \mathbf{V}\mathbf{V}^T \right\|_2 = \left\| \mathbf{V}\mathbf{V}^T \right\|_2 = 1$  for any projector  $\mathbf{V}\mathbf{V}^T \neq \mathbf{0}$  or  $\mathbf{I}$  (see [32]). Similarly, we can show using (8b) that

$$\left\| \mathbf{S}_u^{\text{new}} - \mathbf{S}_{u_{\text{off}}}^{\text{new}} - \mathbf{V}\mathbf{V}^T(\mathbf{S}_u^{\text{new}} - \mathbf{S}_{u_{\text{off}}}^{\text{new}}) \right\|_F^2 \leq \varepsilon_1^2 \left\| \mathbf{S}_u^{\text{new}} \right\|_F^2 + \left\| \widehat{\mathbf{S}}_{\text{new}} - \mathbf{V}\mathbf{V}^T\widehat{\mathbf{S}}_{\text{new}} \right\|_F^2.$$

Finally, combining these two inequalities with assumption (9) yields the desired result:

$$\begin{aligned}
& \left\| \mathbf{S}_u - \mathbf{S}_{u_{\text{off}}} - \mathbf{V}\mathbf{V}^T(\mathbf{S}_u - \mathbf{S}_{u_{\text{off}}}) \right\|_F^2 \\
&= \left\| \mathbf{S}_u^{\text{old}} - \mathbf{S}_{u_{\text{off}}}^{\text{old}} - \mathbf{V}\mathbf{V}^T(\mathbf{S}_u^{\text{old}} - \mathbf{S}_{u_{\text{off}}}^{\text{old}}) \right\|_F^2 + \left\| \mathbf{S}_u^{\text{new}} - \mathbf{S}_{u_{\text{off}}}^{\text{new}} - \mathbf{V}\mathbf{V}^T(\mathbf{S}_u^{\text{new}} - \mathbf{S}_{u_{\text{off}}}^{\text{new}}) \right\|_F^2 \\
&\leq \varepsilon_1^2 (\left\| \mathbf{S}_u^{\text{old}} \right\|_F^2 + \left\| \mathbf{S}_u^{\text{new}} \right\|_F^2) + \left\| \widehat{\mathbf{S}}_{\text{old}} - \mathbf{V}\mathbf{V}^T\widehat{\mathbf{S}}_{\text{old}} \right\|_F^2 + \left\| \widehat{\mathbf{S}}_{\text{new}} - \mathbf{V}\mathbf{V}^T\widehat{\mathbf{S}}_{\text{new}} \right\|_F^2 \\
&= \varepsilon_1^2 \left\| \mathbf{S}_u \right\|_F^2 + \left\| \widehat{\mathbf{S}} - \mathbf{V}\mathbf{V}^T\widehat{\mathbf{S}} \right\|_F^2 \\
&\leq (\varepsilon_1^2 + \varepsilon_2^2) \left\| \mathbf{S}_u \right\|_F^2.
\end{aligned}$$

□

## References

- [1] B. Moore, Principal component analysis in linear systems: Controllability, observability, and model reduction, IEEE transactions on automatic control 26 (1) (1981) 17–32.

- [2] G. Berkooz, P. Holmes, J. L. Lumley, The proper orthogonal decomposition in the analysis of turbulent flows, *Annual review of fluid mechanics* 25 (1) (1993) 539–575.
- [3] K. Willcox, J. Peraire, Balanced model reduction via the proper orthogonal decomposition, *AIAA journal* 40 (11) (2002) 2323–2330.
- [4] C. W. Rowley, T. Colonius, R. M. Murray, Model reduction for compressible flows using POD and Galerkin projection, *Physica D: Nonlinear Phenomena* 189 (1-2) (2004) 115–129.
- [5] N. Ngoc Cuong, K. Veroy, A. T. Patera, Certified real-time solution of parametrized partial differential equations, *Handbook of Materials Modeling: Methods* (2005) 1529–1564.
- [6] M. Bergmann, C.-H. Bruneau, A. Iollo, Enablers for robust POD models, *Journal of Computational Physics* 228 (2) (2009) 516–538.
- [7] K. Carlberg, C. Farhat, J. Cortial, D. Amsallem, The GNAT method for nonlinear model reduction: effective implementation and application to computational fluid dynamics and turbulent flows, *Journal of Computational Physics* 242 (2013) 623–647.
- [8] A. Iollo, D. Lombardi, Advection modes by optimal mass transfer, *Physical Review E* 89 (2) (2014) 022923.
- [9] F. Bernard, A. Iollo, S. Riffaud, Reduced-order model for the BGK equation based on POD and optimal transport, *Journal of Computational Physics* 373 (2018) 545–570.
- [10] L. Sirovich, Turbulence and the dynamics of coherent structures. I. Coherent structures, *Quarterly of applied mathematics* 45 (3) (1987) 561–571.
- [11] L. W. Ng, K. E. Willcox, Multifidelity approaches for optimization under uncertainty, *International Journal for numerical methods in Engineering* 100 (10) (2014) 746–772.
- [12] B. Peherstorfer, K. Willcox, M. Gunzburger, Optimal model management for multifidelity Monte Carlo estimation, *SIAM Journal on Scientific Computing* 38 (5) (2016) A3163–A3194.

- [13] F. Bai, Y. Wang, DEIM reduced order model constructed by hybrid snapshot simulation, *SN Applied Sciences* 2 (12) (2020) 2165.
- [14] F. Bai, Y. Wang, Reduced-order modeling based on hybrid snapshot simulation, *International Journal of Computational Methods* 18 (01) (2021) 2050029.
- [15] L. Feng, G. Fu, Z. Wang, A FOM/ROM hybrid approach for accelerating numerical simulations, *Journal of Scientific Computing* 89 (2021) 1–16.
- [16] V. Zucatti, M. J. Zahr, An adaptive, training-free reduced-order model for convection-dominated problems based on hybrid snapshots, *arXiv preprint arXiv:2301.01718* (2023).
- [17] T. Bui-Thanh, K. Willcox, O. Ghattas, Model reduction for large-scale systems with high-dimensional parametric input space, *SIAM Journal on Scientific Computing* 30 (6) (2008) 3270–3288.
- [18] K. Carlberg, C. Bou-Mosleh, C. Farhat, Efficient non-linear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximations, *International Journal for numerical methods in engineering* 86 (2) (2011) 155–181.
- [19] C. Farhat, P. Avery, T. Chapman, J. Cortial, Dimensional reduction of nonlinear finite element dynamic models with finite rotations and energy-based mesh sampling and weighting for computational efficiency, *International Journal for Numerical Methods in Engineering* 98 (9) (2014) 625–662.
- [20] C. Farhat, T. Chapman, P. Avery, Structure-preserving, stability, and accuracy properties of the energy-conserving sampling and weighting method for the hyper reduction of nonlinear finite element dynamic models, *International journal for numerical methods in engineering* 102 (5) (2015) 1077–1110.
- [21] R. Tezaur, F. As’ad, C. Farhat, Robust and globally efficient reduction of parametric, highly nonlinear computational models and real time online performance, *Computer Methods in Applied Mechanics and Engineering* 399 (2022) 115392.

- [22] C. Eckart, G. Young, The approximation of one matrix by another of lower rank, *Psychometrika* 1 (3) (1936) 211–218.
- [23] C. L. Lawson, R. J. Hanson, Solving least squares problems, SIAM, 1995.
- [24] D. Amsallem, M. J. Zahr, K. Washabaugh, Fast local reduced basis updates for the efficient reduction of nonlinear systems with hyper-reduction, *Advances in Computational Mathematics* 41 (2015) 1187–1230.
- [25] M. Brand, Fast low-rank modifications of the thin singular value decomposition, *Linear algebra and its applications* 415 (1) (2006) 20–30.
- [26] K. Lee, K. T. Carlberg, Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders, *Journal of Computational Physics* 404 (2020) 108973.
- [27] S. Fresca, A. Manzoni, POD-DL-ROM: Enhancing deep learning-based reduced order models for nonlinear parametrized PDEs by proper orthogonal decomposition, *Computer Methods in Applied Mechanics and Engineering* 388 (2022) 114181.
- [28] J. Barnett, C. Farhat, Y. Maday, Neural-network-augmented projection-based model order reduction for mitigating the Kolmogorov barrier to reducibility, *Journal of Computational Physics* 492 (2023) 112420.
- [29] S. Riffaud, M. Bergmann, C. Farhat, S. Grimberg, A. Iollo, The DGDD method for reduced-order modeling of conservation laws, *Journal of Computational Physics* 437 (2021) 110336.
- [30] S. Anderson, C. White, C. Farhat, Space-local reduced-order bases for accelerating reduced-order models through sparsity, *International Journal for Numerical Methods in Engineering* 124 (7) (2023) 1646–1671.
- [31] A. Iollo, G. Sambataro, T. Taddei, A one-shot overlapping Schwarz method for component-based model reduction: application to nonlinear elasticity, *Computer Methods in Applied Mechanics and Engineering* 404 (2023) 115786.
- [32] D. B. Szyld, The many proofs of an identity on the norm of oblique projections, *Numerical Algorithms* 42 (2006) 309–323.