

RESEARCH ARTICLE

Open Access



# Accelerated construction of projection-based reduced-order models via incremental approaches

Eki Agouzal<sup>1,2,3</sup> and Tommaso Taddei<sup>2,3\*</sup> 

\*Correspondence:  
tommaso.taddei@inria.fr

<sup>1</sup>EDF Lab Paris-Saclay, EDF R&D, 7  
Boulevard Gaspard Monge,  
Palaiseau 91120, France

<sup>2</sup>IMB, UMR 5251, University of  
Bordeaux, 351, cours de la  
Libération, Talence 33400, France

<sup>3</sup>Inria Bordeaux Sud-Ouest, Team  
MEMPHIS, 200 Av. de la Vieille  
Tour, Talence 33405, France

## Abstract

We present an accelerated greedy strategy for training of projection-based reduced-order models for parametric steady and unsteady partial differential equations. Our approach exploits hierarchical approximate proper orthogonal decomposition to speed up the construction of the empirical test space for least-square Petrov–Galerkin formulations, a progressive construction of the empirical quadrature rule based on a warm start of the non-negative least-square algorithm, and a two-fidelity sampling strategy to reduce the number of expensive greedy iterations. We illustrate the performance of our method for two test cases: a two-dimensional compressible inviscid flow past a LS89 blade at moderate Mach number, and a three-dimensional nonlinear mechanics problem to predict the long-time structural response of the standard section of a nuclear containment building under external loading.

**Keywords:** Parameterized partial differential equations, Model order reduction, Adaptive sampling, Hyper-reduction

## Introduction

### Projection-based model reduction of parametric systems

In the past few decades, several studies have shown the potential of model order reduction (MOR) techniques to speed up the solution to many-query and real-time problems, and ultimately enable the use of physics-based three-dimensional models for design and optimization, uncertainty quantification, real-time control and monitoring tasks. The distinctive feature of MOR methods is the *offline/online computational decomposition*: during the *offline stage*, high-fidelity (HF) simulations are employed to generate an empirical reduced-order approximation of the solution field and a parametric reduced-order model (ROM); during the *online stage*, the ROM is solved to estimate the solution field and relevant quantities of interest for several parameter values. Projection-based ROMs (PROMs) rely on the projection of the equations onto a suitable low-dimensional test space. Successful MOR techniques should hence achieve significant online speedups at acceptable offline training costs. This work addresses the reduction of offline training costs of PROMs for parametric steady and unsteady partial differential equations (PDEs).

We are interested in the solution to steady parametric conservation laws. We denote by  $\mu$  the vector of  $p$  model parameters in the compact parameter region  $\mathcal{P} \subset \mathbb{R}^p$ ; given the domain  $\Omega \subset \mathbb{R}^d$  ( $d = 2$  or  $d = 3$ ), we introduce the Hilbert spaces  $(\mathcal{X}, \|\cdot\|)$  and  $(\mathcal{Y}, \|\cdot\|)$  defined over  $\Omega$ . Then, we consider problems of the form

$$\text{find } u_\mu \in \mathcal{X} : \mathfrak{R}_\mu(u_\mu, v) = 0, \quad \forall v \in \mathcal{Y}, \quad \mu \in \mathcal{P}, \tag{1}$$

where  $\mathfrak{R} : \mathcal{X} \times \mathcal{Y} \times \mathcal{P} \rightarrow \mathbb{R}$  is the parametric residual associated with the PDE of interest. We here focus on linear approximations, that is we consider reduced-order approximations of the form

$$\widehat{u}_\mu = Z\widehat{\alpha}_\mu, \tag{2}$$

where  $Z : \mathbb{R}^n \rightarrow \mathcal{X}$  is a linear operator, and  $n$  is much smaller than the size of the HF model;  $\widehat{\alpha} : \mathcal{P} \rightarrow \mathbb{R}^n$  is the vector of generalized coordinates. We here exploit the least-square Petrov-Galerkin (LSPG, [1]) ROM formulation proposed in [2], which is well-adapted to approximating advection-dominated problems: the approach relies on the definition of a low-dimensional empirical test space  $\widehat{\mathcal{Y}} \subset \mathcal{Y}$ ; furthermore, it relies on the approximation of the HF residual  $\mathfrak{R}$  through hyper-reduction [3] to enable fast online calculations of the solution to the ROM. In more detail, we consider an empirical quadrature (EQ) procedure [4,5] for hyper-reduction: EQ methods recast the problem of finding a sparse quadrature rule to approximate  $\mathfrak{R}$  as a sparse representation problem and then resort to optimization algorithms to find an approximate solution. Following [4], we resort to the non-negative least-square (NNLS) method to find the quadrature rule.

We further consider the application to unsteady problems: to ease the presentation, we consider one-step time discretizations based on the time grid  $\{t^{(k)}\}_{k=0}^K$ . Given  $\mu \in \mathcal{P}$ , we seek the sequence  $\approx_\mu = \{u_\mu^{(k)}\}_{k=0}^K$  such that

$$\begin{cases} \mathfrak{R}_\mu^{(k)}(u_\mu^{(k)}, u_\mu^{(k-1)}, v) = 0 \quad \forall v \in \mathcal{Y}; \\ u_\mu^{(0)} = \bar{u}_\mu^{(0)}; \end{cases} \tag{3}$$

for all  $\mu \in \mathcal{P}$ . As for the steady-state case, we consider linear ansatzs of the form  $\widehat{u}_\mu^{(k)} = Z\widehat{\alpha}_\mu^{(k)}$ , where  $Z : \mathbb{R}^n \rightarrow \mathcal{X}$  is a linear time- and parameter-independent operator, and  $\widehat{\alpha}^{(0)}, \dots, \widehat{\alpha}^{(K)} : \mathcal{P} \rightarrow \mathbb{R}^n$  are obtained by projecting the Eq. (3) onto a low-dimensional test space. To speed up online costs, we also replace the HF residual in (3) with a rapidly-computable surrogate through the same hyper-reduction technique considered for steady-state problems.

Following the seminal work by Veroy [6], numerous authors have resorted to greedy methods to adaptively sample the parameter space, with the ultimate aim of reducing the number of HF simulations performed during the training phase. Algorithm 1 summarizes the general methodology. First, we initialize the reduced-order basis (ROB)  $Z$  and the ROM based on *a priori* sampling of the parameter space; second, we repeatedly solve the ROM and we estimate the error over a range of parameters  $\mathcal{P}_{\text{train}} \subset \mathcal{P}$ ; third, we compute the HF solution for the parameter that maximizes the error indicator; fourth, if the error is above a certain threshold, we update the ROB  $Z$  and the ROM, and we iterate; otherwise, we terminate. Note that the algorithm depends on an *a posteriori* error indicator  $\Delta$ : if

$\Delta$  is a rigorous *a posteriori* error estimator, we might apply the termination criterion directly to the error indicator (and hence save one HF solve). The methodology has been extended to unsteady problems in [7]: the method in [7] combines a greedy search driven by an *a posteriori* error indicator with proper orthogonal decomposition (POD, [8,9]) to compress the temporal trajectory.

---

**Algorithm 1** Abstract weak greedy algorithm

---

- 1: Choose  $\mathcal{P}_\star = \{\mu^{*,i}\}_{i=1}^{n_0}$  and compute the HF solutions  $\mathcal{S}_\star = \{u_\mu^{\text{hf}} : \mu \in \mathcal{P}_\star\}$ .
  - 2: **for**  $n = n_0 + 1, \dots, n_{\text{max}}$  **do**
  - 3:     Update the ROB  $Z$  and the ROM.
  - 4:     Compute the estimate  $\widehat{u}_\mu^{\text{hf}}$  and evaluate the error indicator  $\Delta_\mu$  for all  $\mu \in \mathcal{P}_{\text{train}}$ .
  - 5:     Compute  $u_{\mu^{*,n}}^{\text{hf}}$  for  $\mu^{*,n} = \arg \max_{\mu \in \mathcal{P}_{\text{train}}} \Delta_\mu$ ; update  $\mathcal{P}_\star$  and  $\mathcal{S}_\star$ .
  - 6:     **if**  $\|u_{\mu^{*,n}}^{\text{hf}} - \widehat{u}_{\mu^{*,n}}^{\text{hf}}\| < \text{tol} \|u_{\mu^{*,n}}^{\text{hf}}\|$  **then**
  - 7:         Update the ROB  $Z$  and the ROM.
  - 8:         break
  - 9:     **end if**
  - 10: **end for**
- 

As observed by several authors, greedy methods enable effective sampling of the parameter space [10]; however, they suffer from several limitations that might ultimately limit their effectiveness compared to standard *a priori* sampling. First, Algorithm 1 is inherently sequential and cannot hence benefit from parallel architectures. Second, Algorithm 1 requires the solution to the ROM and the evaluation of the error indicator for several parameter values at each iteration of the offline stage; similarly, it requires the update of the ROM—i.e., the trial and test bases, the reduced quadrature, and possibly the data structures employed to evaluate the error indicator. These observations motivate the development of more effective training strategies for MOR.

**Contributions and relation to previous work**

We propose an acceleration strategy for Algorithm 1 based on three ingredients: (i) a hierarchical construction of the empirical test space for LSPG ROMs based on the hierarchical approximate POD (HAPOD, [11]); (ii) a progressive construction of the empirical quadrature rule based on a warm start of the NNLS algorithm; (iii) a two-fidelity sampling strategy that is based on the application of the strong-greedy algorithm (see, e.g., [12, section 7.3]) to a dataset of coarse simulations. Re (iii), sampling based on coarse simulations is employed to initialize Algorithm 1 (cf. Line 1) and ultimately reduce the number of expensive greedy iterations. We illustrate the performance of our method for two test cases: a two-dimensional compressible inviscid flow past a LS89 blade at moderate Mach number, and a three-dimensional nonlinear mechanics problem to predict the long-time structural response of the standard section of a nuclear containment building (NCB) under external loading.

Our method shares several features with previously-developed techniques. Incremental POD techniques have been extensively applied to avoid the storage of the full snapshot set for unsteady simulations (see, e.g., [11] and the references therein): here, we adapt

the incremental approach described in [11] to the construction of the test space for LSPG formulations. Chapman [13] extensively discussed the parallelization of the NNLS algorithm for MOR applications: we envision that our method can be easily combined with the method in [13] to further reduce the training costs. Several authors have also devised strategies to speed up the greedy search through the vehicle of a surrogate error model [14,15]; our multi-fidelity strategy extends the work by Barral [16] to unsteady problems and to a more challenging—from the perspective of the HF solver—compressible flow test. We remark that our approach is similar in scope to the work by Benaceur [17] that devised a progressive empirical interpolation method (EIM, [18]) for hyper-reduction of nonlinear problems. Finally, we observe that multi-fidelity techniques have been extensively considered for non-intrusive MOR (see, e.g., [19] and the references therein).

The paper is organized as follows. In “[Projection-based model reduction of parametric systems](#)”, we review relevant elements of the construction of MOR techniques based on LSPG projection for steady conservation laws; we further address the construction of Galerkin ROMs for problems of the form (3). Then, in “[Accelerated construction of PROMs](#)”, we present the accelerated strategy for both classes of ROMs. “[Numerical results](#)” illustrates the performance of the method for two model problems. “[Conclusions](#)” draws some conclusions and discusses future developments.

### Projection-based model reduction of parametric systems

“[Preliminary definitions and tools](#)” summarizes notation that is employed throughout the paper and reviews three algorithms—POD, strong-greedy, and the active set method for NNLS problems—that are used afterwards. “[Least-square Petrov–Galerkin formulation of steady](#)” reviews the LSPG formulation employed in the present work for steady conservation laws and illustrates the strategies employed for the definition of the quadrature rule and the empirical test space. Finally, “[Vanilla POD-greedy for Galerkin time-marching ROMs](#)” addresses model reduction of unsteady systems of the form (3).

#### Preliminary definitions and tools

We denote by  $\mathcal{T}_{\text{hf}} = (\{x_j^{\text{hf}}\}_{j=1}^{N_{\text{nd}}}, \mathbb{T})$  the HF mesh of the domain  $\Omega$  with nodes  $\{x_j^{\text{hf}}\}_j$  and connectivity matrix  $\mathbb{T}$ ; we introduce the elements  $\{D_k\}_{k=1}^{N_e}$  and the facets  $\{F_j\}_{j=1}^{N_f}$  of the mesh and we define the open set  $\tilde{F}_j$  in  $\Omega$  as the union of the elements of the mesh that share the facet  $F_j$ , with  $j = 1, \dots, N_f$ . We further denote by  $\mathcal{X}_{\text{hf}}$  the HF space associated with the mesh  $\mathcal{T}_{\text{hf}}$  and we define  $N_{\text{hf}} := \dim(\mathcal{X}_{\text{hf}})$ .

Exploiting the previous definitions, we can introduce the HF residual:

$$\mathfrak{R}_{\mu}^{\text{hf}}(w, v) = \sum_{k=1}^{N_e} r_{k,\mu}^e(w|_{D_k}, v|_{D_k}) + \sum_{j=1}^{N_f} r_{j,\mu}^f(w|_{\tilde{F}_j}, v|_{\tilde{F}_j}), \tag{4}$$

for all  $w, v \in \mathcal{X}_{\text{hf}}$ . To shorten notation, in the remainder, we do not explicitly include the restriction operators in the residual. The global residual can be viewed as the sum of local element-wise residuals and local facet-wise residuals, which can be evaluated at a cost that is independent of the total number of elements and facets, and is based on local information. The nonlinear infinite-dimensional statement (1) translates into the

high-dimensional problem: find  $u_\mu^{\text{hf}} \in \mathcal{X}_{\text{hf}}$  such that

$$\mathfrak{R}_\mu^{\text{hf}}(u_\mu^{\text{hf}}, v) = 0 \quad \forall v \in \mathcal{X}_{\text{hf}}. \tag{5}$$

Since (5) is nonlinear, the solution requires to solve a nonlinear system of  $N_{\text{hf}}$  equations with  $N_{\text{hf}}$  unknowns. Towards this end, we here resort to the pseudo-transient continuation (PTC) strategy proposed in [20].

We use the method of snapshots (cf. [8]) to compute POD eigenvalues and eigenvectors. Given the snapshot set  $\{u^k\}_{k=1}^{n_{\text{train}}} \subset \mathcal{X}_{\text{hf}}$  and the inner product  $(\cdot, \cdot)_{\text{pod}}$ , we define the Gramian matrix  $\mathbf{C} \in \mathbb{R}^{n_{\text{train}} \times n_{\text{train}}}$ ,  $\mathbf{C}_{k,k'} = (u^k, u^{k'})_{\text{pod}}$ , and we define the POD eigenpairs  $\{(\lambda_i, \zeta_i)\}_{i=1}^{n_{\text{train}}}$  as

$$\mathbf{C}\zeta_i = \lambda_i \zeta_i, \quad \zeta_i := \sum_{k=1}^{n_{\text{train}}} (\zeta_i)_k u^k, \quad i = 1, \dots, n_{\text{train}},$$

with  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_{n_{\text{train}}} \geq 0$ . In our implementation, we orthonormalize the modes, that is  $(\zeta_n, \zeta_n)_{\text{pod}} = 1$  for  $n = 1, \dots, n_{\text{train}}$ . In the remainder we use notation

$$[\{\zeta_i\}_{i=1}^n, \{\lambda_i\}_{i=1}^n] = \text{POD}\left(\{u^k\}_{k=1}^{n_{\text{train}}}, n, (\cdot, \cdot)_{\text{pod}}\right) \tag{6}$$

to refer to the application of POD to the snapshot set  $\{u^k\}_{k=1}^{n_{\text{train}}}$ . The number of modes  $n$  can be chosen adaptively by ensuring that the retained energy content is above a certain threshold (see, e.g., [12, Eq. (6.12)]).

We further recall the strong-greedy algorithm: the algorithm takes as input the snapshot set  $\{u^k\}_{k=1}^{n_{\text{train}}}$ , an integer  $n$ , an inner product  $(\cdot, \cdot)_{\text{sg}}$  and the induced norm  $\|\cdot\|_{\text{sg}} = \sqrt{(\cdot, \cdot)_{\text{sg}}}$ , and returns a set of  $n$  indices  $\mathcal{I}_{\text{sg}} \subset \{1, \dots, n_{\text{train}}\}$

$$[\mathcal{I}_{\text{sg}}] = \text{strong-greedy}\left(\{u^k\}_{k=1}^{n_{\text{train}}}, n, (\cdot, \cdot)_{\text{sg}}\right). \tag{7}$$

The dimension  $n$  can be chosen adaptively by ensuring that the projection error is below a given threshold  $\text{tol}_{\text{sg}} > 0$ ,

$$n = \min \left\{ n' \in \{1, \dots, n_{\text{train}}\} : \max_{k \in \{1, \dots, n_{\text{train}}\}} \frac{\|\Pi_{\mathcal{Z}_{n'}^\perp} u^k\|_{\text{sg}}}{\|u^k\|_{\text{sg}}} \leq \text{tol}_{\text{sg}} \right\}, \tag{8}$$

where  $\mathcal{Z}_{n'}$  denotes the  $n'$ -dimensional space obtained after  $n'$  steps of the greedy procedure in Algorithm 2,  $\mathcal{Z}_{n'}^\perp$  is the orthogonal complement of the space  $\mathcal{Z}_{n'}$  and  $\Pi_{\mathcal{Z}_{n'}^\perp} : \mathcal{X} \rightarrow \mathcal{Z}_{n'}^\perp$  is the orthogonal projection operator onto  $\mathcal{Z}_{n'}^\perp$ .

We conclude this section by reviewing the active set method [21] that is employed to find a sparse solution to the non-negative least square problem:

$$\min_{\rho \in \mathbb{R}^N} \|\mathbf{G}\rho - \mathbf{b}\|_2 \quad \text{s.t.} \quad \rho \geq 0, \tag{9}$$

for a given matrix  $\mathbf{G} \in \mathbb{R}^{M \times N}$  and a vector  $\mathbf{b} \in \mathbb{R}^M$ . Algorithm 3 reviews the computational procedure. In the remainder, we use notation

$$[\rho] = \text{NNLS}(\mathbf{G}, \mathbf{b}, \delta, P_0), \tag{10}$$

---

**Algorithm 2** Strong-greedy algorithm

---

*Inputs:*  $\{u^k\}_{k=1}^{n_{\text{train}}}$  snapshot set,  $n$  size of the desired reduced space,  $(\cdot, \cdot)_{\text{sg}}$  inner product.

*Outputs:*  $\mathcal{I}_{\text{sg}}$  indices of selected snapshots.

- 1: Choose  $\mathcal{Z}_0 = \emptyset, \mathcal{I}_{\text{sg}} = \emptyset$ , set  $\mathcal{I}_{\text{train}} = \{1, \dots, n_{\text{train}}\}$ .
  - 2: **for**  $i = 1, \dots, n$  **do**
  - 3:     Compute  $i^* = \arg \max_{i \in \mathcal{I}_{\text{train}}} \|\Pi_{\mathcal{Z}^\perp} u^i\|_{\text{sg}}$ .
  - 4:     Update  $\mathcal{Z}_i = \mathcal{Z}_{i-1} \cup \text{span}\{u^{i^*}\}$  and  $\mathcal{I}_{\text{sg}} = \mathcal{I}_{\text{sg}} \cup \{i^*\}$ .
  - 5: **end for**
- 

---

**Algorithm 3** Active set method for (9)

---

*Inputs:*  $\mathbf{G} \in \mathbb{R}^{M \times N}, \mathbf{b} \in \mathbb{R}^M, \delta > 0, P_0$ .

*Output:*  $\rho$  approximate solution to (9),  $it$  number of iterations to meet convergence criterion.

- 1: Choose  $\rho = 0, \mathbf{w} = \mathbf{C}^\top \rho, P = P_0, it = 0$ .
  - 2: **while** true **do**
  - 3:     Compute  $\mathbf{r} = \mathbf{G}(:, P)\mathbf{x}(P) - \mathbf{b}$ .
  - 4:     **if**  $\#P = N$  or  $\|\mathbf{r}\|_2 \leq \delta \|\mathbf{b}\|_2$  **then**
  - 5:         break
  - 6:     **end if**
  - 7:     Set  $i^* = \arg \max_{j \notin P} (\mathbf{w})_j$ .
  - 8:     Set  $P = P \cup \{i^*\}$ .
  - 9:     **while** true **do**
  - 10:          $it = it + 1$
  - 11:         Define  $\mathbf{z} \in \mathbb{R}^N$  s.t.  $\mathbf{z}(P^c) = 0, \mathbf{z}(P) = \mathbf{G}(:, P)^\dagger \mathbf{d}$
  - 12:         **if**  $\mathbf{z} \geq 0$  **then**
  - 13:             Set  $\mathbf{x} = \mathbf{z}, \mathbf{w} = \mathbf{G}^\top (\mathbf{b} - \mathbf{G}\mathbf{x})$
  - 14:             break
  - 15:         **end if**
  - 16:          $\mathcal{I} = \{i \in \{1, \dots, N\} : (\mathbf{z})_i < 0\}$ .
  - 17:          $[\alpha, i^*] = \min_{i \in \mathcal{I}} \frac{(\mathbf{x})_i}{(\mathbf{x})_i - (\mathbf{z})_i + \epsilon}$ .
  - 18:          $P = P \setminus \{i^*\}$ .
  - 19:          $\mathbf{x} = \mathbf{x} - \alpha(\mathbf{x} - \mathbf{z})$ .
  - 20:     **end while**
  - 21: **end while**
- 

to refer to the application of Algorithm 3.

Note that the method takes as input a set of indices—which is initialized with the empty set in the absence of prior information—to initialize the process. Given the matrix  $\mathbf{G} = [\mathbf{g}_1, \dots, \mathbf{g}_N]$ , the vector  $\mathbf{x} \in \mathbb{R}^N$ , and the set of indices  $P = \{p_i\}_{i=1}^m \subset \{1, \dots, N\}$ , we use notation  $\mathbf{G}(:, P) := [\mathbf{g}_{p_1}, \dots, \mathbf{g}_{p_m}] \in \mathbb{R}^{M \times m}$  and  $\mathbf{x}(P) = \text{vec}((\mathbf{x})_{p_1}, \dots, (\mathbf{x})_{p_m}) \in \mathbb{R}^m$ ; we denote by  $\#P$  the cardinality of the discrete set  $P$ , and we introduce the complement of  $P$  in  $\{1, \dots, N\}$  as  $P^c = \{1, \dots, N\} \setminus P$ . Given the vector  $\mathbf{x} \in \mathbb{R}^N$  and the set of indices

$\mathcal{I} \subset \{1, \dots, N\}$ , notation  $[\alpha, i^*] = \min_{i \in \mathcal{I}} (\mathbf{x})_i$  signifies that  $\alpha = \min_{i \in \mathcal{I}} (\mathbf{x})_i$  and  $i^* \in \mathcal{I}$  realizes the minimum,  $\alpha = (\mathbf{x})_{i^*}$ . The constant  $\epsilon > 0$  is intended to avoid division by zero and is set to  $2^{-1022}$ . The computational cost of Algorithm 3 is dominated by the cost to repeatedly solve the least-square problem at Line 11: in “Numerical results”, we hence report the total number of least-square solves needed to achieve convergence (cf. output *it* in Algorithm 3).

**Least-square Petrov–Galerkin formulation of steady problems**

Given the reduced-order basis (ROB)  $Z = [\zeta_1, \dots, \zeta_n] : \mathbb{R}^n \rightarrow \mathcal{X}_{\text{hf}}$ , following [16], we consider the LSPG formulation

$$\widehat{u}_\mu = Z\widehat{\alpha}_\mu \quad \text{with} \quad \widehat{\alpha}_\mu \in \arg \min_{\alpha \in \mathbb{R}^n} \max_{\psi \in \widehat{\mathcal{Y}}} \frac{\mathfrak{R}_\mu^{\text{eq}}(Z\alpha, \psi)}{\|\psi\|} \tag{11a}$$

where  $\widehat{\mathcal{Y}} = \text{span}\{\psi_i\}_{i=1}^m$  is a suitable test space that is chosen below and the empirical residual  $\mathfrak{R}_\mu^{\text{eq}}$  satisfies

$$\mathfrak{R}_\mu^{\text{eq}}(w, v) = \sum_{k=1}^{N_e} \rho_k^{\text{eq},e} r_{k,\mu}^e(w, v) + \sum_{j=1}^{N_f} \rho_k^{\text{eq},f} r_{j,\mu}^f(w, v) \tag{11b}$$

where  $\rho^{\text{eq},e} \in \mathbb{R}_+^{N_e}$  and  $\rho^{\text{eq},f} \in \mathbb{R}_+^{N_f}$  are sparse vectors of non-negative weights.

Provided that  $\{\psi_i\}_{i=1}^m$  is an orthonormal basis of  $\widehat{\mathcal{Y}}$ , we can rewrite (11a) as

$$\min_{\alpha \in \mathbb{R}^n} \|\mathfrak{R}_\mu^{\text{eq}}(\alpha)\|_2, \quad \text{with} \quad (\mathfrak{R}_\mu^{\text{eq}}(\alpha))_i = \mathfrak{R}_\mu^{\text{eq}}(Z\alpha, \psi_i), \quad i = 1, \dots, m, \tag{12}$$

which can be efficiently solved using the Gauss-Newton method (GNM). Note that (12) does not explicitly depend on the choice of the test norm  $\|\cdot\|$ : dependence on the norm is implicit in the choice of  $\{\psi_i\}_{i=1}^m$ . Formulation (11)–(12) depends on the choice of the test space  $\widehat{\mathcal{Y}}$  and the empirical weights  $\rho^{\text{eq},e}, \rho^{\text{eq},f}$ : in the remainder of this section, we address the construction of these ingredients.

We remark that Carlberg [1] considered a different projection method, which is based on the minimization of the Euclidean norm of the discrete residual: due to the particular choice of the test norm, the approach of [1] does not require the explicit construction of the empirical test space. We further observe that Yano and collaborators [22,23] have considered different formulations of the empirical residual (11b). A thorough comparison between different projection methods and different hyper-reduction techniques is beyond the scope of the present study.

**Construction of the empirical test space**

As discussed in [2], the test space  $\widehat{\mathcal{Y}}$  should approximate the Riesz representers of the functionals associated with the action of the Jacobian on the elements of the trial ROB. Given the snapshot set of HF solutions  $\{u_\mu^{\text{hf}} : \mu \in \mathcal{P}_{\text{train}}\}$  with  $\mathcal{P}_{\text{train}} = \{\mu^k\}_{k=1}^n \subset \mathcal{P}$  and the ROB  $\{\zeta_i\}_i$ , we hence apply POD to the test snapshot set  $\mathcal{S}_n^{\text{test}} := \{\Psi_{k,i} : k = 1, \dots, n_{\text{train}}, i = 1, \dots, n\}$ , where

$$((\Psi_{k,i}, v)) = \mathfrak{J}_{\mu^k}^{\text{hf}} \left[ \widehat{u}_{\mu^k}^{\text{hf}} \right] (\zeta_i, v), \quad \forall v \in \mathcal{X}_{\text{hf}}, \tag{13}$$

where  $\mathfrak{J}_\mu^{\text{hf}}[w] : \mathcal{X}_{\text{hf}} \times \mathcal{X}_{\text{hf}} \rightarrow \mathbb{R}$  denotes the Fréchet derivative of the HF residual at  $w$ . It is also useful to provide an approximate computation of the right-hand side of (13),

$$((\Psi_{k,i} v)) \approx \frac{1}{\epsilon} \left( \mathfrak{R}_{\mu^k}^{\text{hf}}(u_{\mu^k}^{\text{hf}} + \epsilon \zeta_i v) - \mathfrak{R}_{\mu^k}^{\text{hf}}(u_{\mu^k}^{\text{hf}}, v) \right) \quad \forall v \in \mathcal{X}_{\text{hf}}, \tag{14}$$

with  $|\epsilon| \ll 1$ . The evaluation of the right-hand side of (14) involves the computation of the residual at  $u_{\mu^k}^{\text{hf}} + \epsilon \zeta_i$  for  $i = 1, \dots, n$ ; on the other hand, the evaluation of (13) requires the computation of the Jacobian matrix at  $u_{\mu^k}^{\text{hf}}$  and the post-multiplication by the algebraic counterpart of  $Z$ . Both (13) and (14)—which are equivalent in the limit  $\epsilon \rightarrow 0$ —are used in the incremental approach of “[Progressive construction of empirical test space and quadrature](#)”.

**Construction of the empirical quadrature rule**

Following [5], we seek  $\rho^{\text{eq,e}} \in \mathbb{R}_+^{N_e}$  and  $\rho^{\text{eq,f}} \in \mathbb{R}_+^{N_f}$  in (11b) such that

- (i) (*efficiency constraint*) the number of nonzero entries in  $\rho^{\text{eq,e}}$ ,  $\rho^{\text{eq,f}}$ ,  $\text{nnz}(\rho^{\text{eq,e}})$  and  $\text{nnz}(\rho^{\text{eq,f}})$ , is as small as possible;
- (ii) (*constant function constraint*) the constant function is approximated correctly in  $\Omega$

$$\left| \sum_{k=1}^{N_e} \rho_k^{\text{eq,e}} |D_k| - |\Omega| \right| \ll 1, \quad \left| \sum_{j=1}^{N_f} \rho_j^{\text{eq,f}} |F_j| - \sum_{j=1}^{N_f} |F_j| \right| \ll 1; \tag{15}$$

- (iii) (*manifold accuracy constraint*) for all  $\mu \in \mathcal{P}_{\text{train,eq}} = \{\mu^k\}_{k=1}^{n_{\text{train}} + n_{\text{train,eq}}}$ , the empirical residual satisfies

$$\left\| \mathfrak{R}_\mu^{\text{hf}}(\alpha_\mu^{\text{train}}) - \mathfrak{R}_\mu^{\text{eq}}(\alpha_\mu^{\text{train}}) \right\|_2 \ll 1, \tag{16a}$$

where  $\mathfrak{R}_\mu^{\text{hf}}$  corresponds to substitute  $\rho_1^{\text{eq,e}} = \dots = \rho_{N_e}^{\text{eq,e}} = \rho_1^{\text{eq,f}} = \dots = \rho_{N_f}^{\text{eq,f}} = 1$  in (11b) and  $\alpha_\mu^{\text{train}}$  satisfies

$$\alpha_\mu^{\text{train}} = \begin{cases} \arg \min_{\alpha \in \mathbb{R}^n} \|Z\alpha - u_\mu^{\text{hf}}\|, & \text{if } \mu \in \mathcal{P}_{\text{train}}; \\ \arg \min_{\alpha \in \mathbb{R}^n} \|\mathfrak{R}_\mu^{\text{hf}}(\alpha)\|_2, & \text{if } \mu \notin \mathcal{P}_{\text{train}}; \end{cases} \tag{16b}$$

and  $\mathcal{P}_{\text{train}} = \{\mu^k\}_{k=1}^{n_{\text{train}}}$  is the set of parameters for which the HF solution is available. In the remainder, we use  $\mathcal{P}_{\text{train,eq}} = \mathcal{P}_{\text{train}}$ .

By tedious but straightforward calculations, we find that

$$\mathfrak{R}_{\mu^k}^{\text{eq}}(Z\alpha, \psi_i) = \mathbf{G}_{k,i}^e \cdot \rho^{\text{eq,e}} + \mathbf{G}_{k,i}^f \cdot \rho^{\text{eq,f}} \tag{17a}$$

for some row vectors  $\mathbf{G}_{k,i}^e \in \mathbb{R}^{1 \times N_e}$  and  $\mathbf{G}_{k,i}^f \in \mathbb{R}^{1 \times N_f}$ ,  $k = 1, \dots, n_{\text{train}}$  and  $i = 1, \dots, m$ . Therefore, if we define

$$\mathbf{G}^{\text{cnst,e}} = [ |D_1|, \dots, |D_{N_e}| ], \quad \mathbf{G}^{\text{cnst,f}} = [ |F_1|, \dots, |F_{N_f}| ], \tag{17b}$$



we find that the constraints (15) and (16) can be expressed in the algebraic form

$$\|\mathbf{G}(\boldsymbol{\rho}^{\text{eq}} - \boldsymbol{\rho}^{\text{hf}})\|_2 \ll 1, \quad \text{with } \mathbf{G} = \begin{bmatrix} \mathbf{G}_{1,1}^e & \mathbf{G}_{1,1}^f \\ \vdots & \vdots \\ \mathbf{G}_{n_{\text{train}},m}^e & \mathbf{G}_{n_{\text{train}},m}^f \\ \mathbf{G}^{\text{cnst},e} & 0 \\ 0 & \mathbf{G}^{\text{cnst},f} \end{bmatrix}, \quad \boldsymbol{\rho}^{\text{eq}} = \begin{bmatrix} \boldsymbol{\rho}^{\text{eq},e} \\ \boldsymbol{\rho}^{\text{eq},f} \end{bmatrix}, \quad (17c)$$

and  $\boldsymbol{\rho}^{\text{hf}} = [1, \dots, 1]^\top$ .

In conclusion, the problem of finding the sparse weights  $\boldsymbol{\rho}^{\text{eq},e}, \boldsymbol{\rho}^{\text{eq},f}$  can be recast as a sparse representation problem

$$\min_{\boldsymbol{\rho} \in \mathbb{R}_+^{N_e+N_f}} \text{nnz}(\boldsymbol{\rho}), \quad \|\mathbf{G}\boldsymbol{\rho}^{\text{eq}} - \mathbf{b}\|_2 \leq \delta \|\mathbf{b}\|_2, \quad (18)$$

where  $\text{nnz}(\boldsymbol{\rho})$  is the number of non-zero entries in the vector  $\boldsymbol{\rho}$ , for some user-defined tolerance  $\delta > 0$ , and  $\mathbf{b} = \mathbf{G}\boldsymbol{\rho}^{\text{hf}}$ . Following [4], we resort to the NNLS algorithm discussed in “[Preliminary definitions and tools](#)” (cf. Algorithm 3) to find an approximate solution to (18).

#### ***A posteriori error indicator***

The final element of the formulation is the *a posteriori* error indicator that is employed for the parameter exploration. We here consider the residual-based error indicator (cf. [16]),

$$\Delta : \mu \in \mathcal{P} \mapsto \sup_{v \in \mathcal{X}_{\text{hf}}} \frac{\mathfrak{R}_\mu^{\text{hf}}(\widehat{u}_\mu, v)}{\|v\|}. \quad (19)$$

Note that the evaluation of (19) requires the solution to a symmetric positive definite linear system of size  $N_{\text{hf}}$ : it is hence ill-suited for real-time online computations; nevertheless, in our experience the offline cost associated with the evaluation of (19) is comparable with the cost that is needed to solve the ROM—clearly, this is related to the size of the mesh and is hence strongly problem-dependent.

#### ***Overview of the computational procedure***

Algorithm 4 provides a detailed summary of the construction of the ROM at each step of Algorithm 1 (cf. Line 7). Some comments are in order. The cost of Algorithm 4 is dominated by the assembly of the test snapshot set and by the solution to the NNLS problem. We also notice that the storage of  $\mathcal{S}_n^{\text{test}}$  scales with  $\mathcal{O}(n^2)$  and is hence the dominant memory cost of the offline procedure.

#### **Vanilla POD-greedy for Galerkin time-marching ROMs**

We denote by  $u_\mu \in \mathbb{R}^d$  the displacement field, by  $\sigma_\mu \in \mathbb{R}^{d \times d}$  the Cauchy tensor, by  $\varepsilon_\mu = \nabla_s u_\mu$  the strain tensor with  $\nabla_s \bullet = \frac{1}{2}(\nabla \bullet + \nabla \bullet^\top)$ ; we further introduce the vector of internal variables  $\gamma_\mu \in \mathbb{R}^{d_{\text{int}}}$ . Then, we introduce the quasi-static equilibrium equations

---

**Algorithm 4** Generation of the ROM

---

*Inputs:*  $Z = [\zeta_1, \dots, \zeta_{n-1}]$  current ROB,  $\text{tol} > 0$  tolerance for Algorithm 3,  $u_{\mu^n}^{\text{hf}}$  new snapshot,  $m$  size of the test space

*Outputs:*  $Z = [\zeta_1, \dots, \zeta_n]$  new ROB, ROM for the generalized coordinates.

- 1: Apply Gram-Schmidt orthogonalization to define the new ROB  $Z$ .
  - 2: Assemble the test snapshot set  $S_n^{\text{test}}$  using (13).
  - 3: Apply POD to find  $\widehat{\mathcal{Y}}: \{\psi_i\}_{i=1}^m = \text{POD}(S_n^{\text{test}}, m, ((\cdot, \cdot)))$ .
  - 4: Assemble the matrix and vector  $\mathbf{G}, \mathbf{b}$  (cf. (17)).
  - 5: Solve the NNLS problem (cf. Algorithm 3).
  - 6: Update the ROM data structures.
- 

(completed with suitable boundary and initial conditions)

$$\begin{cases} -\nabla \cdot \sigma_\mu = f & \text{in } \Omega \\ \sigma_\mu = \mathcal{F}_\mu^\sigma(\varepsilon_\mu, \gamma_\mu) & \text{in } \Omega \\ \dot{\gamma}_\mu = \mathcal{F}_\mu^\gamma(\varepsilon_\mu, \gamma_\mu) & \text{in } \Omega \end{cases} \quad (20)$$

where  $\mathcal{F}_\mu^\sigma : \mathbb{R}^{d \times d} \times \mathbb{R}^{d_{\text{int}}} \rightarrow \mathbb{R}^{d \times d}$  and  $\mathcal{F}_\mu^\gamma : \mathbb{R}^{d \times d} \times \mathbb{R}^{d_{\text{int}}} \rightarrow \mathbb{R}^{d_{\text{int}}}$  are suitable parametric functions that encode the material constitutive law—note that the Newton’s law (20)<sub>1</sub> does not include the inertial term; the temporal evolution is hence entirely driven by the constitutive law (20)<sub>3</sub>. Equation (20) is discretized using the FE method in space and a one-step finite difference (FD) method in time. Given the parameter  $\mu \in \mathcal{P}$ , the FE space  $\mathcal{X}_{\text{hf}}$  and the time grid  $\{t^{(k)}\}_{k=1}^K$ , we seek the sequence  $\approx_\mu = \{u_\mu^{(k)}\}_{k=0}^K$  such that

$$\mathfrak{R}_\mu^{(k)}(u_\mu^{(k)}, v) = \sum_{\ell=1}^{N_e} r_{\ell, \mu}^{(k), e}(u_\mu^{(k)}, v) + \sum_{j \in \mathcal{I}_{\text{bnd}}} r_{j, \mu}^{(k), f}(u_\mu^{(k)}, v), \quad (21)$$

where the elemental residuals satisfy

$$\begin{aligned} r_{\ell, \mu}^{(k), e}(w, v) &= \int_{D_\ell} \left( \mathcal{F}_\mu^\sigma(\nabla_s w, \gamma_\mu^{(k)}(w)) : \nabla_s v - f \cdot v \right) dx, \ell = 1, \dots, N_e, \\ \text{with } \gamma_\mu^{(k)}(w) &= \mathcal{F}_{\mu, \Delta t}^\gamma(\nabla_s w, \varepsilon_\mu^{(k-1)}, \gamma_\mu^{(k-1)}), \end{aligned} \quad (22)$$

$\mathcal{I}_{\text{bnd}} \subset \{1, \dots, N_f\}$  are the indices of the boundary facets and the facet residuals  $\{r_{j, \mu}^{(k), f}\}_{j \in \mathcal{I}_{\text{bnd}}}$  incorporate boundary conditions. Note that  $\mathcal{F}_{\mu, \Delta t}^\gamma$  is the FD approximation of the constitutive law (20)<sub>3</sub>.

Given the reduced space  $\mathcal{Z} = \text{span}\{\zeta_i\}_{i=1}^n \subset \mathcal{X}_{\text{hf}}$ , the time-marching hyper-reduced Galerkin ROM of (21)–(22) reads as: given  $\mu \in \mathcal{P}$ , find  $\widehat{\approx}_\mu = \{\widehat{u}_\mu^{(k)}\}_{k=0}^K$  such that

$$\mathfrak{R}_\mu^{(k), \text{eq}}(\widehat{u}_\mu^{(k)}, v) = \sum_{\ell=1}^{N_e} \rho_\ell^{\text{eq}, e} r_{\ell, \mu}^{(k), e}(\widehat{u}_\mu^{(k)}, v) + \sum_{j \in \mathcal{I}_{\text{bnd}}} \rho_j^{\text{eq}, f} r_{j, \mu}^{(k), f}(\widehat{u}_\mu^{(k)}, v), \quad \forall v \in \mathcal{Z}, \quad (23)$$

for  $k = 1, \dots, K$ , where  $\rho^{\text{eq}, e} \in \mathbb{R}_+^{N_e}$  and  $\rho^{\text{eq}, f} \in \mathbb{R}_+^{N_f}$  are suitably-chosen sparse vectors of weights. We observe that the Galerkin ROM (23) does not reduce the total number of time steps: solution to (23) hence requires the solution to a sequence of  $K$  nonlinear problems

of size  $n$  and is thus likely much more expensive than the solution to (Petrov-) Galerkin ROMs for steady-state problems. Furthermore, several independent studies have shown that residual-based *a posteriori* error estimators for (23) are typically much less sharp and reliable than their counterparts for steady-state problems. These observations have motivated the development of space-time formulations [24]: to our knowledge, however, space-time methods have not been extended to problems with internal variables.

The abstract Algorithm 1 can be readily extended to time-marching ROMs for unsteady PDEs: the POD-Greedy method [7] combines a greedy search in the parameter domain with a temporal compression based on POD. At each iteration  $it$  of the algorithm, we update the reduced space  $\mathcal{Z}$  using the newly-computed trajectory  $\{u_{\mu^*,it}^{(k)}\}_{k=1}^K$  and we update the quadrature rule. The reduced space  $\mathcal{Z}$  should be updated using hierarchical methods that avoid the storage of the full space-time trajectory for all sampled parameters: the hierarchical approximate POD (HAPOD, see [11] and also “Empirical test space”) or the hierarchical approach in [25, section 3.5] that generates nested spaces: we refer to [26, section 3.2.1] for further details and extensive numerical investigations for a model problem with internal variables. Here, we consider the nested approach of [25]: we denote by  $n_{it}$  the dimension of the reduced space  $\mathcal{Z}_{it}$  at the  $it$ -th iteration, and we denote by  $n_{\text{new}} = n_{it} - n_{it-1}$  the number of modes added at each iteration,

$$\mathcal{Z}_{it} = \mathcal{Z}_{it-1} \oplus \mathcal{Z}^{\text{new}}, \quad \text{where } \mathcal{Z}^{\text{new}} = \text{POD} \left( \left\{ \Pi_{\mathcal{Z}_{it-1}^\perp} u_{\mu^*,it}^{(k)} \right\}_{k=1}^K, n_{\text{new}}(\text{tol}), (\cdot, \cdot) \right), \tag{24a}$$

where  $n_{\text{new}}(\text{tol})$  satisfies  $n_{\text{new}}(\text{tol}) =$

$$\min \left\{ n' : \frac{\sum_{k=1}^K \Delta t^{(k)} \|u_{\mu^*,it}^{(k)} - \Pi_{\mathcal{Z}_{it-1} \oplus \mathcal{Z}_{n'}^{\text{new}}} u_{\mu^*,it}^{(k)}\|^2}{\sum_{k=1}^K \Delta t^{(k)} \|u_{\mu^*,it}^{(k)}\|^2} \leq \text{tol}^2, \mathcal{Z}_{n'}^{\text{new}} = \text{span} \left\{ s_{it,i}^{\text{new}} \right\}_{i=1}^{n'} \right\}, \tag{24b}$$

for some tolerance  $\text{tol} > 0$ , with  $\Delta t^{(k)} = t^{(k)} - t^{(k-1)}, k = 1, \dots, K$ . The quadrature rule is obtained using the same procedure described in “Empirical quadrature”: exploiting (24), it is easy to verify that the matrix  $\mathbf{G}$  (cf. (17c)) can be rewritten as (we omit the details)

$$\mathbf{G}^{(it)} = \begin{bmatrix} \mathbf{G}^{\text{cnst,e}} & 0 \\ 0 & \mathbf{G}^{\text{cnst,f}} \\ & \mathbf{G}_{\text{acc}}^{(it-1)} \\ & \mathbf{G}_{\text{new}}^{(it)} \end{bmatrix}, \quad \text{with } \begin{cases} \mathbf{G}_{\text{acc}}^{(it-1)} \in \mathbb{R}^{(n_{it-1}(it-1)K) \times (N_e + N_f)}, \\ \mathbf{G}_{\text{new}}^{(it)} \in \mathbb{R}^{(n_{\text{new}}(it-1)K + n_{it}K) \times (N_e + N_f)}. \end{cases} \tag{25}$$

Note that  $\mathbf{G}_{\text{acc}}^{it}$  corresponds to the columns of the matrix  $\mathbf{G}$  associated with the manifold accuracy constraints (cf. (16)) at the  $it$ -th iteration of the greedy procedure.

### Accelerated construction of PROMs

“Progressive construction of empirical test space and quadrature” discusses the incremental strategies proposed in this work, to reduce the costs associated with the construction of the empirical test space and the empirical quadrature in Algorithm 4; “Multi-fidelity sampling” summarizes the multi-fidelity sampling strategy, which is designed to reduce

the total number of greedy iterations required by Algorithm 1. “Progressive construction of empirical test space and quadrature” and “Multi-fidelity sampling” focus on LSPG ROMs for steady problems; in “Extension to unsteady problems” we discuss the extension to Galerkin ROMs for unsteady problems.

**Progressive construction of empirical test space and quadrature**

**Empirical test space**

By reviewing Algorithms 1 and 4, we notice that the test snapshot set  $\mathcal{S}_n^{\text{test}}$  satisfies

$$\mathcal{S}_n^{\text{test}} = \mathcal{S}_{n-1}^{\text{test}} \cup \{\Psi_{k,n}\}_{k=1}^{n-1} \cup \{\Psi_{n,i}\}_{i=1}^n; \tag{26}$$

therefore, at each iteration, it suffices to solve  $2n - 1$ —as opposed to  $n^2$ —Riesz problems of the form (13) to define  $\mathcal{S}_n^{\text{test}}$ . As in [16], we rely on Cholesky factorization with fill-in reducing permutations of rows and columns, to reduce the cost of solving the Riesz problems. In the numerical experiments, we rely on (13), which involves the assembly of the Jacobian matrix, to compute  $\{\Psi_{n,1}\}_{k=1}^n$ , while we consider the finite difference approximation (14) to compute  $\{\Psi_{k,n}\}_{k=1}^{n-1}$  with  $\epsilon = 10^{-6}$ .

In order to lower the memory costs associated with the storage of the test snapshot set  $\mathcal{S}_n^{\text{test}}$  and also the cost of performing POD, we consider a hierarchical approach to construct the test space  $\widehat{\mathcal{Y}}$ . In this work, we apply the (distributed) hierarchical approximate proper orthogonal decomposition: HAPOD approach is related to incremental singular value decomposition [27] and guarantees near-optimal performance with respect to the standard POD (cf. [11]). Given the POD space  $\{\psi_i\}_{i=1}^m$  such that  $((\psi_i, \psi_j)) = \delta_{ij}$  and the corresponding eigenvalues  $\{\lambda_i\}_{i=1}^m$ , and the new set of snapshots  $\{\Psi_{k,n}\}_{k=1}^{n-1} \cup \{\Psi_{n,i}\}_{i=1}^n$ , HAPOD corresponds to applying POD to a suitable snapshot set that combines information from current and previous iterations,

$$[\{\psi_i^{\text{new}}\}_{i=1}^{m_{\text{new}}}, \{\lambda_i^{\text{new}}\}_{i=1}^{m_{\text{new}}}] = \text{POD}(\mathcal{S}_n^{\text{incr}}, m_{\text{new}}, ((\cdot, \cdot))), \tag{27}$$

with  $\mathcal{S}_n^{\text{incr}} := \{\sqrt{\lambda_i}\psi_i\}_{i=1}^m \cup \{\Psi_{k,n}\}_{k=1}^{n-1} \cup \{\Psi_{n,i}\}_{i=1}^n$ . Note that the modes  $\{\psi_i\}_{i=1}^m$  are scaled to properly take into account the energy content of the modes in the subsequent iterations. Note also that the storage cost of the method scales with  $m + 2n - 1$ , which is much lower than  $n^2$ , provided that  $m \ll n^2$ . Note also that the POD spaces  $\{\widehat{\mathcal{Y}}_n\}_n$ , which are generated at each iteration of Algorithm 1 using (27), are not nested.

**Empirical quadrature**

Exploiting (17), it is straightforward to verify that if the test spaces are nested — that is,  $\widehat{\mathcal{Y}}_{n-1} \subset \widehat{\mathcal{Y}}_n$ , then the EQ matrix  $\mathbf{G}_n$  at iteration  $n$  satisfies

$$\mathbf{G}_n = \begin{bmatrix} \mathbf{G}_{n-1} \\ \mathbf{G}_n^{\text{new}} \end{bmatrix}, \tag{28}$$

where  $\mathbf{G}_n^{\text{new}}$  has  $k = n \times \dim(\widehat{\mathcal{Y}}_n \setminus \widehat{\mathcal{Y}}_{n-1}) + \dim(\widehat{\mathcal{Y}}_{n-1})$  rows. We hence observe that we can reduce the cost of assembling the EQ matrix  $\mathbf{G}_n$  by exploiting (28), provided that we rely on nested test spaces: since, in our experience, the cost of assembling  $\mathbf{G}_n$  is negligible, the use of non-nested test spaces does not hinder offline performance.

On the other hand, due to the strong link between consecutive EQ problems that are solved at each iteration of the greedy procedure, we propose to initialize the NNLS Algorithm 3 using the solution from the previous time step, that is

$$P = \left\{ i \in \{1, \dots, N_e + N_f\} : (\rho^{\text{eq},(n-1)})_i \neq 0 \right\}. \tag{29}$$

We provide extensive numerical investigations to assess the effectiveness of this choice.

**Summary of the incremental weak-greedy algorithm**

Algorithm 5 reviews the full incremental generation of the ROM. In the numerical experiments, we set  $m = 2n$ : this implies that the storage of  $\mathcal{S}^{\text{incr}}$  scales with  $4n - 3$  as opposed to  $n^2$ .

---

**Algorithm 5** Incremental generation of the ROM

---

*Inputs:*  $Z = [\zeta_1, \dots, \zeta_{n-1}]$  current ROB,  $\delta > 0$  tolerance for Algorithm 3,  $u_{\mu^n}^{\text{hf}}$  new snapshot,  $\{(\psi_i^{\text{old}}, \lambda_i^{\text{old}})\}_{i=1}^{m'}$  POD eigenpairs for test space,  $m$  size of the test space,  $P^{(n-1)}$  initial condition for Algorithm 3.

*Outputs:*  $Z = [\zeta_1, \dots, \zeta_n]$  new ROB, ROM for the generalized coordinates,  $\{(\psi_i, \lambda_i)\}_{i=1}^m$  new POD eigenpairs for test space,  $P^{(n)}$  initial condition for Algorithm 3 at subsequent iteration.

- 1: Apply Gram-Schmidt orthogonalization to define the new ROB  $Z$ .
  - 2: Assemble the test snapshot set  $\mathcal{S}_n^{\text{incr}}$  in (27).
  - 3: Apply POD to find  $\widehat{\mathcal{Y}}, [\{\psi_i\}_{i=1}^m, \{\lambda_i\}_{i=1}^m] = \text{POD}(\mathcal{S}_n^{\text{incr}}, m, ((\cdot, \cdot)))$ .
  - 4: Assemble the matrix and vector  $\mathbf{G}, \mathbf{b}$  (cf. (17)).
  - 5: Solve the NNLS problem (cf. Algorithm 3) with initial condition given by  $P^{(n-1)}$ .
  - 6: Update the ROM data structures and compute  $P$  using (29).
- 

**Multi-fidelity sampling**

The incremental strategies of “[Progressive construction of empirical test space and quadrature](#)” do not affect the total number of greedy iterations required by Algorithm 1 to achieve the desired accuracy; furthermore, they do not address the reduction of the cost of the HF solves. Following [16], we here propose to resort to coarser simulations to learn the initial training set  $\mathcal{P}_\star$  (cf. Line 1 Algorithm 1) and to initialize the HF solver. Algorithm 6 summarizes the computational procedure.

**Algorithm 6** Multi-fidelity weak-greedy algorithm

- 
- 1: Define two meshes  $\mathcal{T}_{\text{hf}}^0$  and  $\mathcal{T}_{\text{hf}}$ , where  $\mathcal{T}_{\text{hf}}^0$  is coarser than  $\mathcal{T}_{\text{hf}}$ .
  - 2: Generate a ROM for the HF model associated with the coarse mesh  $\mathcal{T}_{\text{hf}}^0$  (e.g. using Algorithm 1),  $Z^0, \hat{\alpha}^0 : \mathcal{P} \rightarrow \mathbb{R}^n$ .
  - 3: Estimate the solution for all  $\mu \in \mathcal{P}_{\text{train}}$  and store the generalized coordinates  $\{\hat{\alpha}_\mu^0 : \mu \in \mathcal{P}_{\text{train}}\}$ .
  - 4: Apply Algorithm 2 to  $\{\hat{\alpha}_\mu^0 : \mu \in \mathcal{P}_{\text{train}}\}$  to obtain the initial sample  $\mathcal{P}_\star$ .
  - 5: Apply Algorithm 1 to the HF model associated with the fine mesh  $\mathcal{T}_{\text{hf}}$ ; use  $\mathcal{P}_\star$  to initialize the greedy method, and use the estimate  $\mu \mapsto Z^0 \hat{\alpha}_\mu^0$  to initialize the HF solver.
- 

Some comments are in order.

- In the numerical experiments, we choose the cardinality  $n_0$  of  $\mathcal{P}_\star$  according to (8) with  $\text{tol}_{\text{sg}} = \text{tol}$  (cf. Algorithm 1). Note that, since the strong greedy algorithm is applied to the generalized coordinates of the coarse ROM,  $n_0$  cannot exceed the size  $n$  of the ROM.
- We observe that increasing the cardinality of  $\mathcal{P}_\star$  ultimately leads to a reduction of the number of sequential greedy iterations; it hence enables much more effective parallelization of the offline stage. Note also that Algorithm 6 can be coupled with parametric mesh adaptation tools to build an effective problem-aware mesh  $\mathcal{T}_{\text{hf}}$  (cf. [16]); in this work, we do not exploit this feature of the method.
- We observe that the multi-fidelity Algorithm 6 critically depends on the choice of the coarse grid  $\mathcal{T}_{\text{hf}}^0$ : an excessively coarse mesh  $\mathcal{T}_{\text{hf}}^0$  might undermine the quality of the initial sample  $\mathcal{P}_\star$  and of the initial condition for the HF solve, while an excessively fine mesh  $\mathcal{T}_{\text{hf}}^0$  reduces the computational gain. In the numerical experiments, we provide extensive investigations of the influence of the coarse mesh on performance.

**Extension to unsteady problems**

Acceleration of the POD-greedy algorithm for the Galerkin ROM (23) relies on two independent building blocks: first, the incremental construction of the quadrature rule; second, a two-fidelity sampling strategy. Re the quadrature procedure, we notice that at each greedy iteration, the matrix  $\mathbf{G}$  in (25) admits the decomposition in (28). If the number of new modes is modest compared to  $n_{it-1}$  the rows of  $\mathbf{G}_{\text{new}}^{(it)}$  are significantly less numerous than the columns of  $\mathbf{G}^{(it)}$ : we can hence reduce the cost of the construction of  $\mathbf{G}^{(it)}$  by keeping in memory the EQ matrix from the previous iteration; furthermore, the decomposition (28) motivates the use of the active set of weights from the previous iterations to initialize the NNLS algorithm at the current iteration. On the other hand, the extension of the two-fidelity sampling strategy in Algorithm 6 simply relies on the generalization of the strong-greedy procedure 2 to unsteady problems, which is illustrated in the algorithm below.

---

**Algorithm 7** POD-strong-greedy algorithm

---

*Inputs:*  $\{\approx_i = \{u_i^{(k)}\}_{k=1}^K\}_{i=1}^{n_{\text{train}}}$  snapshot set, `maxit` maximum number of iterations, `tol` tolerance for data compression,  $(\cdot, \cdot)_{\text{sg}}$  inner product.

*Outputs:*  $\mathcal{I}_{\text{sg}}$  indices of selected snapshots.

- 1: Choose  $\mathcal{Z}_0 = \emptyset, \mathcal{I}_{\text{sg}} = \emptyset$ , set  $\mathcal{I}_{\text{train}} = \{1, \dots, n_{\text{train}}\}$ .
  - 2: **for**  $it = 1, \dots, \text{maxit}$  **do**
  - 3:     Compute  $i^* = \arg \max_{i \in \mathcal{I}_{\text{train}}} \sum_{k=1}^K \left( \min_{w \in \mathcal{Z}} \|w - u_i^{(k)}\|_{\text{sg}}^2 \right)$ .
  - 4:     Update the reduced space  $\mathcal{Z}_{it} = \text{data-compression} \left( \mathcal{Z}_{it-1}, \{u_{i^*}^{(k)}\}_{k=1}^K, \text{tol} \right)$ .  
(cf. (24))
  - 5:      $\mathcal{I}_{\text{sg}} = \mathcal{I}_{\text{sg}} \cup \{i^*\}$ .
  - 6: **end for**
- 

Algorithm 7 takes as input a set of trajectories and returns the indices of the selected parameters. To shorten the notation, we here assume that the time grid is the same for all parameters: however, the algorithm can be readily extended to cope with parameter-dependent temporal discretizations. We further observe that the procedure depends on the data compression strategy employed to update the reduced space at each greedy iteration: in this work, we consider the same hierarchical strategy (cf. [25, section 3.5]) employed in the POD-(weak-)greedy algorithm.

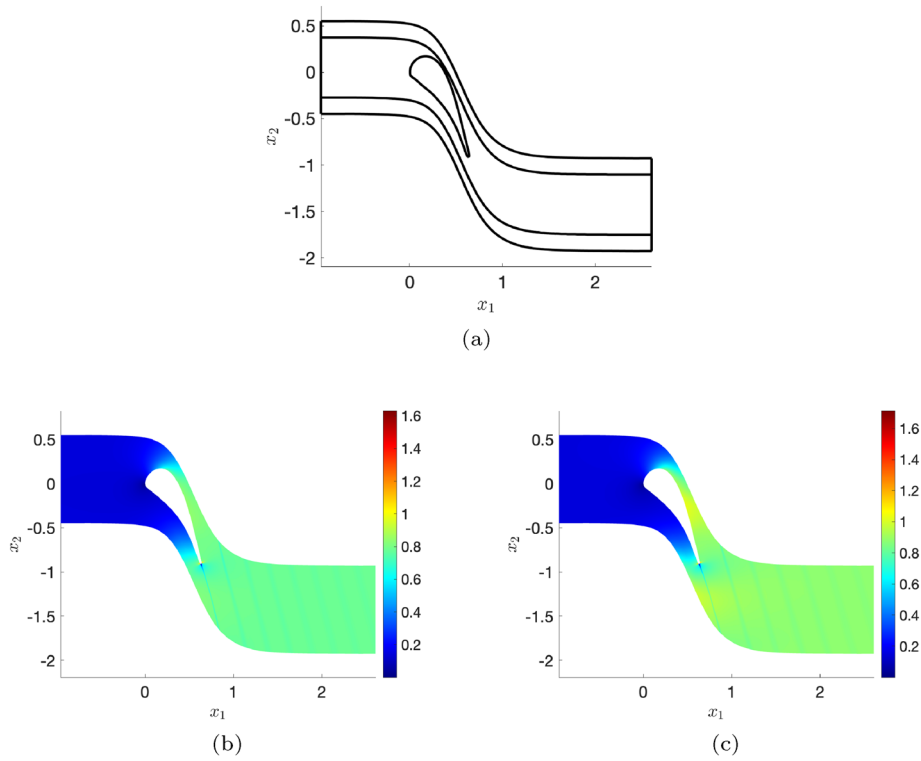
**Numerical results**

We present numerical results for a steady compressible inviscid flow past a LS89 blade (cf. “[Transonic compressible flow past an LS89 blade](#)”), and for an unsteady nonlinear mechanics problem that simulates the long-time mechanical response of the standard section of a containment building under external loading (cf. “[Long-time mechanical response of the standard section of a containment building under external loading](#)”). Simulations of “[Transonic compressible flow past an LS89 blade](#)” are performed in Matlab 2022a [28] based on an in-house code, and executed over a commodity Linux workstation (RAM 32 GB, Intel i7 CPU 3.20 GHz x 12). The HF simulations of “[Long-time mechanical response of the standard section of a containment building under external loading](#)” are performed using the FE software `code_aster` [29] and executed over a commodity Linux workstation (RAM 32 GB, Intel i7-9850H CPU 2.60 GHz x 12); on the other hand, the MOR procedure relies on an in-house Python code and is executed on a Windows workstation (RAM 16 GB, Intel i7-9750H CPU 2.60 GHz x 12).

**Transonic compressible flow past an LS89 blade**

**Model problem**

We consider the problem of estimating the solution to the two-dimensional Euler equations past an array of LS89 turbine blades; the same model problem is considered in [30], for a different parameter range. We consider the computational domain depicted in Fig. 1a; we prescribe total temperature, total pressure and flow direction at the inflow, static pressure at the outflow, non-penetration (wall) condition on the blade and periodic boundary conditions on the lower and upper boundaries. We study the sensitivity of the solution



**Fig. 1** Inviscid flow past an array of LS89 turbine blades. **a** Partition associated with the geometric map. **b, c** Behavior of the Mach number for two parameter values

with respect to two parameters: the free-stream Mach number  $Ma_\infty$  and the height of the channel  $H$ ,  $\mu = [H, Ma_\infty]$ . We consider the parameter domain  $\mathcal{P} = [0.9, 1.1] \times [0.2, 0.9]$ .

We deal with geometry variations through a piecewise-smooth mapping associated with the partition in Fig. 1a. We set  $H_{ref} = 1$  and we define the curve  $x_1 \mapsto f_{btm}(x_1)$  that describes the lower boundary  $\Gamma_{btm}$  of the domain  $\Omega = \Omega(H = 1)$ ; then, we define  $\tilde{H} > 0$  such that  $x_1 \mapsto f_{btm}(x_1) + \tilde{H}$  and  $x_1 \mapsto f_{btm}(x_1) + H - \tilde{H}$  do not intersect the blade for any  $H \in [0.9, 1.1]$ ; finally, we define the geometric mapping

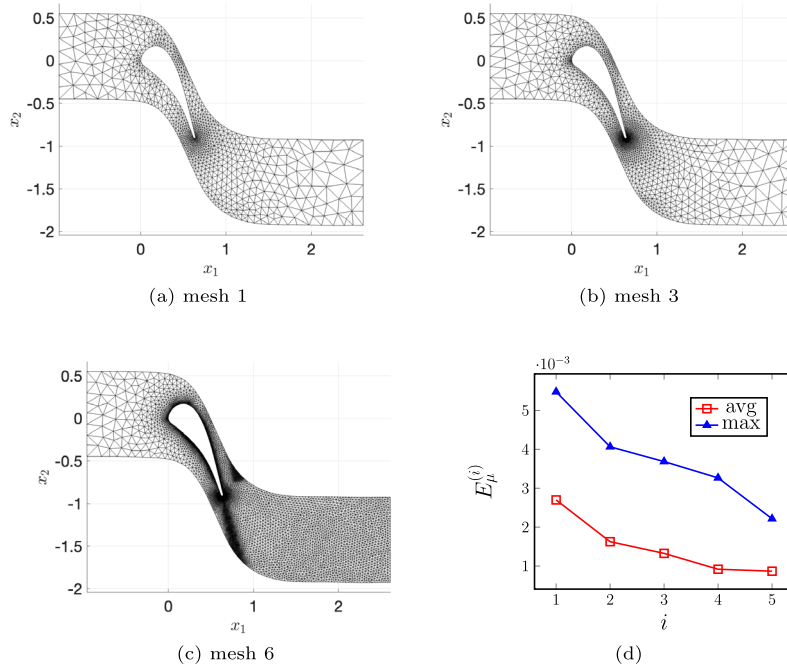
$$\Psi_H^{geo}(x = [x_1, x_2]) = \begin{bmatrix} x_1 \\ \psi_H^{geo}(x) \end{bmatrix}, \tag{30a}$$

where

$$\psi_H^{geo}(x) = \begin{cases} o_1(x_1) + C(H)(x_2 - o_1(x_1)) & x_2 < o_1(x_1), \\ o_2(x_1) + C(H)(x_2 - o_2(x_1)) & x_2 > o_2(x_1), \\ x_2 & \text{otherwise,} \end{cases} \tag{30b}$$

with  $o_1(x_1) = f_{btm}(x_1) + \tilde{H}$ ,  $o_2(x_1) = f_{btm}(x_1) + H_{ref} - \tilde{H}$  and  $C(H) = \frac{H - H_{ref}}{2H} + 1$ . Fig. 1b, c show the distribution of the Mach field for  $\mu^{(1)} = [0.95, 0.78]$  and  $\mu^{(2)} = [1.05, 0.88]$ . We notice that for large values of the free-stream Mach number the solution develops a normal shock on the upper side of the blade and two shocks at the trailing edge: effective approximation for higher values of the Mach number requires the use of nonlinear approximations and is beyond the scope of the present work.





**Fig. 2** Compressible flow past a LS89 blade. **a–c** Three computational meshes. **d** Behavior of the average and maximum error (31)

### Results

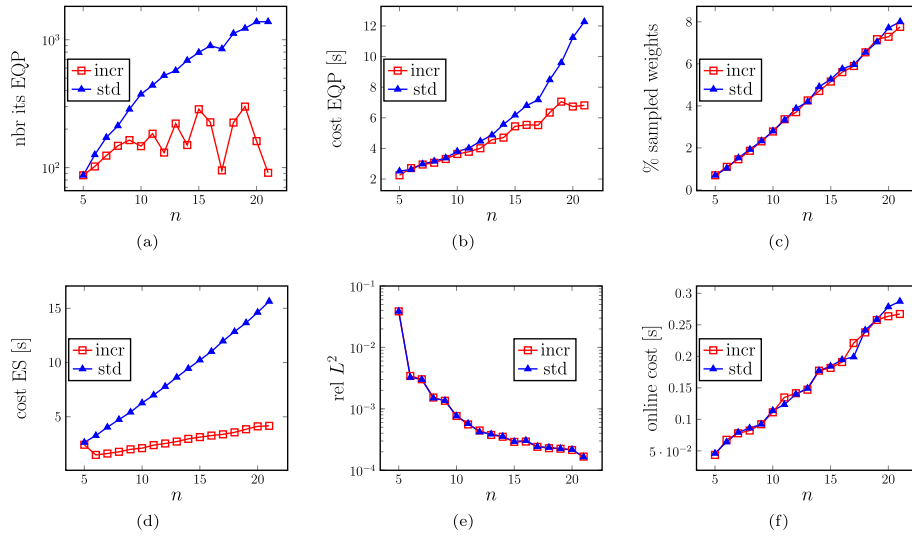
We consider a hierarchy of six P2 meshes with  $N_e = 1827, 2591, 3304, 4249, 7467, 16353$  elements, respectively; Fig. 2a–c show three computational meshes. Figure 2d show maximum and mean errors between the HF solution  $u_{\mu}^{\text{hf},(i)}$  and the corresponding HF solution associated with the  $i$ -th mesh,

$$E_{\mu}^{(i)} = \frac{\|u_{\mu}^{\text{hf},(i)} - u_{\mu}^{\text{hf},(6)}\|}{\|u_{\mu}^{\text{hf},(6)}\|}, \quad (31)$$

over five randomly-chosen parameters in  $\mathcal{P}$ . We remark that the error is measured in the reference configuration  $\Omega$ , which corresponds to  $H = 1$ . We consider the  $L^2(\Omega)$  norm  $\|\cdot\| = \sqrt{\int_{\Omega} (\cdot)^2 dx}$ .

Figure 3 compares the performance of the standard (“std”) greedy method with the performance of the incremental (“incr”) procedures of [“Progressive construction of empirical test space and quadrature”](#) for the coarse mesh (mesh 1). In all the tests below, we consider a training space  $\mathcal{P}_{\text{train}}$  based on a ten by ten equispaced discretization of the parameter domain. Figure 3a shows the number of iterations of the NNLS algorithm 3, while Fig. 3b shows the wall-clock cost (in seconds) on a commodity laptop, and Fig. 3c shows the percentage of sampled weights  $\frac{\text{nnz}(\rho^{\text{eq}})}{N_e + N_f} \times 100\%$ . We observe that the proposed initialization of the active set method leads to a significant reduction of the total number of iterations, and to a non-negligible reduction of the total wall-clock cost<sup>1</sup>, without affecting the per-

<sup>1</sup>The computational cost reduction is not as significant as the reduction in the total number of iterations, because the cost per iteration depends on the size of the least-square problem to be solved (cf. Line 11, Algorithm 3), which increases as we increase the cardinality of the active set.



**Fig. 3** Compressible flow past a LS89 blade. Progressive construction of quadrature rule and test space, coarse mesh ( $N_e = 1827$ )

formance of the method. Figure 3d shows the cost of constructing the test space: we notice that the progressive construction of the test space enables a significant reduction of offline costs. Finally, Fig. 3e shows the averaged out-of-sample performance of the LSPG ROM over  $n_{\text{test}} = 20$  randomly-chosen parameters  $\mathcal{P}_{\text{test}}$ ,

$$E_{\text{avg}} = \frac{1}{n_{\text{test}}} \sum_{\mu \in \mathcal{P}_{\text{test}}} \frac{\|u_{\mu}^{\text{hf},(1)} - \hat{u}_{\mu}\|}{\|u_{\mu}^{\text{hf},(1)}\|},$$

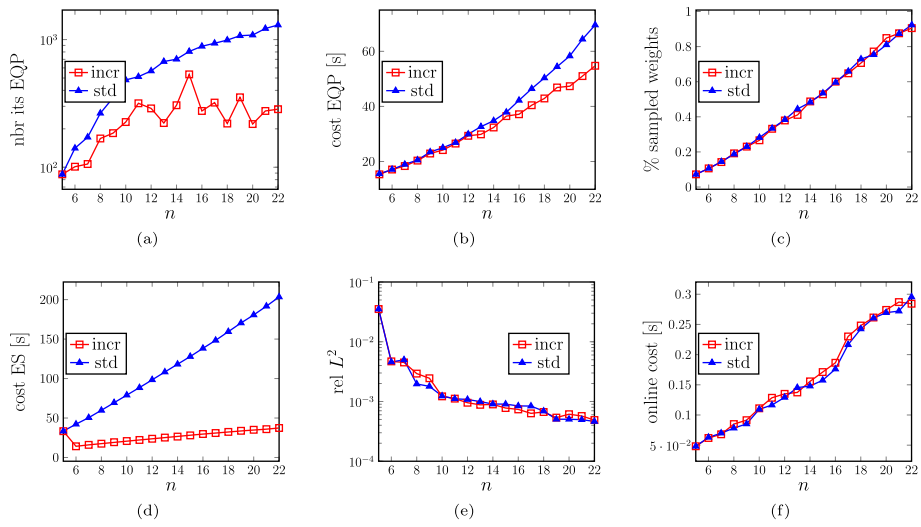
and Fig. 3f shows the online costs: we observe that the standard and the incremental approaches lead to nearly-equivalent results for all values of the ROB size  $n$  considered.

Figure 4 replicates the tests of Fig. 3 for the fine mesh (mesh 6). As for the coarser grid, we find that the progressive construction of the test space and of the quadrature rule do not hinder online performance of the LSPG ROM and ensure significant offline savings. We notice that the relative error over the test set is significantly larger: reduction of the mesh size leads to more accurate approximations of sharp features—shocks, wakes—that are troublesome for linear approximations. On the other hand, we notice that online costs are nearly the same as for the coarse mesh for the corresponding value of the ROB size  $n$ : this proves the effectiveness of the hyper-reduction procedure.

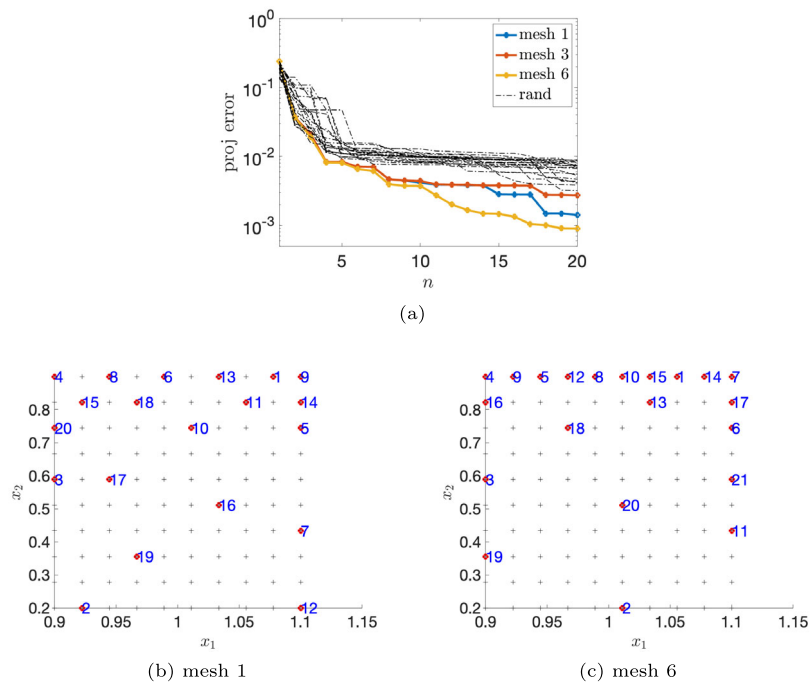
Figure 5 investigates the effectiveness of the sampling strategy based on the strong-greedy algorithm. We rely on Algorithm 6 to identify the training set of parameters  $\mathcal{P}_{\star}$  for different choices of the coarse mesh  $\mathcal{T}_{\text{hf}}^0 = \mathcal{T}_{\text{hf}}^{(i)}$ , for  $i = 1, \dots, 6$  and for  $\mathcal{T}_{\text{hf}} = \mathcal{T}_{\text{hf}}^{(6)}$ . Then, we measure performance in terms of the maximum relative projection error over  $\mathcal{P}_{\text{train}}$ ,

$$E_n^{\text{proj},(i)} = \max_{\mu \in \mathcal{P}_{\text{train}}} \inf_{\zeta \in \mathcal{Z}_n^{(i)}} \frac{\|u_{\mu}^{\text{hf},(6)} - \zeta\|}{\|u_{\mu}^{\text{hf},(6)}\|}, \quad \text{with } \mathcal{Z}_{n'}^{(i)} = \text{span} \left\{ u_{\mu}^{\text{hf},(6)} : \mu \in \mathcal{P}_{\star, n'}^{(i)} \right\}, \quad (32)$$

where  $\mathcal{P}_{\star, n'}^{(i)}$  is the set of the first  $n'$  parameters selected through Algorithm 6 based on the coarse mesh  $\mathcal{T}_{\text{hf}}^{(i)}$ . Figure 5a shows the behavior of the projection error  $E_n^{\text{proj},(i)}$  (32) for three different choices of the coarse mesh; to provide a concrete reference, we also report the performance of twenty sequences of reduced spaces obtained by randomly selecting



**Fig. 4** Compressible flow past a LS89 blade. Progressive construction of quadrature rule and test space; fine mesh ( $N_e = 16353$ )



**Fig. 5** Compressible flow past a LS89 blade; sampling. **a** behavior of  $E_n^{proj,l}$  (32) for six different choices of the coarse mesh, and for random samples. **b, c** Parameters  $\{\mu^{*j}\}$  selected by Algorithm 6 for two different coarse meshes

sequences of parameters in  $\mathcal{P}_{train}$ . Figure 5b, c show the parameters selected through Algorithm 6 for two different choices of the coarse mesh: we observe that the selected parameters are clustered in the proximity of  $Ma_\infty = 0.9$ .

Table 1 compares the costs of the standard weak-greedy algorithm (“vanilla”), the weak-greedy algorithm with progressive construction of the test space and the quadrature rule (“incr”), and the two-fidelity Algorithm 6 with coarse mesh given by  $\mathcal{T}_{hf}^0 = \mathcal{T}_{hf}^{(1)}$

**Table 1** Compressible flow past a LS89 blade

Fine mesh $\mathcal{T}_{\text{hf}}^{(5)}$	ROB	ES	EQP	Greedy search	HF solves	Overhead	Total
Vanilla	1.1	714.5	247.4	930.6	3554.8	0.6	5449.0
incr	1.1	177.9	221.2	928.4	3504.8	0.6	4834.0
incr+MF	0.9	131.0	121.9	362.7	2030.0	692.1	3338.5
Fine mesh $\mathcal{T}_{\text{hf}}^{(6)}$							
Vanilla	3.4	2067.8	668.4	2091.2	10,997.4	2.3	15,830.5
incr	3.2	444.8	538.4	1952.8	10,529.1	2.3	13,470.6
incr+MF	3.6	366.4	362.5	905.8	9597.7	699.9	11,935.9

Overview of offline costs for three computational strategies and two different fine meshes (coarse mesh:  $\mathcal{T}_{\text{hf}}^{(1)}$ )

(“incr+MF”). To ensure a fair comparison, we impose that the final ROM has the same number of modes (twenty) for all cases. Training of the coarse ROM in Algorithm 6 is based on the weak-greedy algorithm with progressive construction of the test space and the quadrature rule, with tolerance  $\epsilon_{\text{ol}} = 10^{-3}$ : this leads to a coarse ROM with  $n_0 = 14$  modes, which corresponds to an initial training set  $\mathcal{P}_*$  of cardinality 14 in the greedy method (cf. Line 5, Algorithm 6). The ROMs associated with three different training strategies show comparable performance in terms of online cost and  $L^2$  errors.

For the fine mesh  $\mathcal{T}_{\text{hf}}^{(6)}$ , the two-fidelity training leads to a reduction of offline costs of roughly 25% with respect to the vanilla implementation and of roughly 10% with respect to the incremental implementation; for the fine mesh  $\mathcal{T}_{\text{hf}}^{(5)}$  (which has one half as many elements as the finer grid), the two-fidelity training leads to a reduction of offline costs of roughly 39% with respect to the vanilla implementation and of roughly 31% with respect to the incremental implementation. In particular, we notice that the initialization based on the coarse model—which is the same for both cases—is significantly more effective for the HF model associated with mesh 5 than for the HF model associated with mesh 6. The empirical finding strengthens the observation made in “Multi-fidelity sampling” that the choice of the coarse approximation is a compromise between overhead costs—which increase as we increase the size of the coarse mesh—and accuracy of the coarse solution.

We remark that for the first two cases the HF solver is initialized using the solution for the closest parameter in the training set

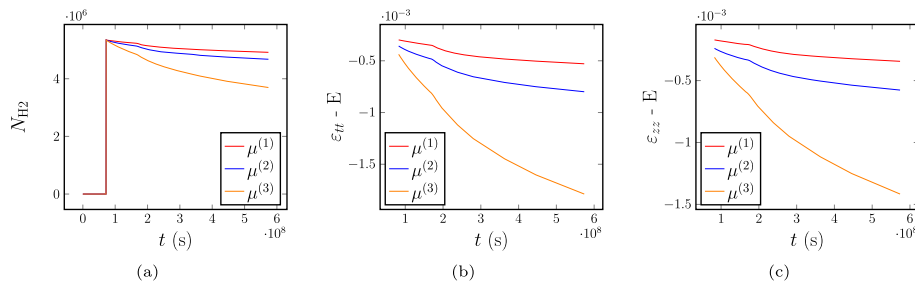
$$u_{\mu^{*,n}}^0 = u_{\bar{\mu}}^{\text{hf}}, \quad \text{with } \bar{\mu} = \arg \min_{\mu' \in \{\mu^{*,i}\}_{i=1}^{n-1}} \|\mu^{*,n} - \mu'\|_2,$$

for  $n = 2, 3, \dots$ ; for  $n = 1$ , we rely on a coarse solver and on a continuation strategy with respect to the Mach number: this initialization is inherently sequential. On the other hand, in the two-fidelity procedure, the HF solver is initialized using the reduced-order solver that is trained using coarse HF data: this choice enables trivial parallelization of the HF solves for the initial parameter sample  $\mathcal{P}_*$ . We hence expect to achieve higher computational savings for parallel computations.

**Long-time mechanical response of the standard section of a containment building under external loading**

**Model problem**

We study the long-time mechanical response of a three-dimensional standard section of a nuclear power plant containment building: the highly-nonlinear mechanical response is activated by thermal effects; its simulation requires the coupling of thermal, hydraulic and



**Fig. 6** Mechanical response of a NCB under external loading. **a** Normal force on a horizontal cable. **b** Tangential and **(c)** vertical strains on the outer wall of the standard section of the containment building, for  $\eta_{dc} = 5 \cdot 10^9$  and three values of  $\kappa$  on the coarse mesh

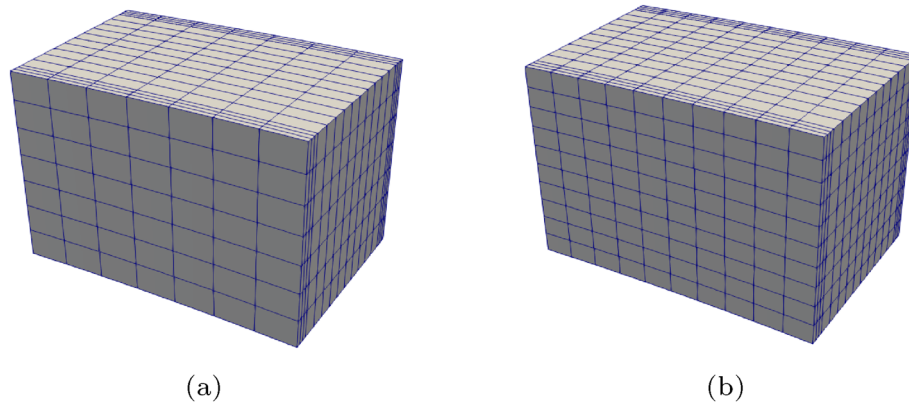
mechanical (THM) responses. A thorough presentation of the mathematical model and of the MOR procedure is provided in [31]. The ultimate goal of the simulation is to predict the temporal behavior of several quantities of interest (QoIs), such as water saturation in concrete, delayed deformations, and stresses: these QoIs are directly related to the leakage rate, whose estimate is of paramount importance for the design of NCBs. The deformation field is also important to conduct validation and calibration studies against real-world data.

Following [32], we consider a weak THM coupling procedure to model deferred deformations within the material; weak coupling is appropriate for large structures under normal operational loads. The MOR process is exclusively applied to estimate the mechanical response: the results from thermal and hydraulic calculations are indeed used as input data for the mechanical calculations, which constitute the computational bottleneck of the entire simulation workflow. To model the mechanical response of the concrete structure, we consider a three-dimensional nonlinear rheological creep model with internal variables; on the other hand, we consider a one-dimensional linear elastic model for the prestressing cables: the state variables are hence the displacement field of the three-dimensional concrete structure and of the one-dimensional steel cables. We assume that the whole structure satisfies the small-strain and small-displacement hypotheses. To establish connectivity between concrete and steel nodes, a kinematic linkage is implemented: a point within the steel structure and its corresponding point within the concrete structure are assumed to share identical displacements.

We study the solution behavior with respect to two parameters: the desiccation creep viscosity ( $\eta_{dc}$ ) and the basic creep consolidation parameter ( $\kappa$ ) in the parameter range  $\mu \in \mathcal{P} = [5 \cdot 10^8, 5 \cdot 10^{10}] \times [10^{-5}, 10^{-3}] \subset \mathbb{R}^2$ . Figure 6 shows the behavior of (a) the normal force on a horizontal cable, and (b) the tangential and (c) vertical strains on the outer wall of the standard section of the containment building, for three distinct parameter values  $\mu^{(i)} = (5 \cdot 10^9, \kappa^{(i)})$ , for  $\kappa^{(i)} \in \{10^{-5}, 10^{-4}, 10^{-3}\}$ ,  $i = 1, 2, 3$ . Notation “-E” indicates that the HF data are associated to the outer face of the structure. Note that the value of the consolidation parameter  $\kappa$  affects the rate of decay of the various quantities.

**Results**

**High-fidelity solver.** We consider the two distinct three-dimensional meshes, depicted in Fig. 7: a coarse mesh of  $N_e = 784$  three-dimensional hexahedral elements and a refined mesh with  $N_e = 1600$  elements. This mesh features the geometry of a portion of the



**Fig. 7** Mechanical response of a NCB under external loading. Tridimensional meshes (HEXA20) used for code\_aster calculations: **a** coarse mesh (mesh 1:  $N_e = 784$ ); **b** refined mesh (mesh 2:  $N_e = 1600$ )

**Table 2** Mechanical response of a NCB under external loading

	Mean	Max	Min	Q1	Median	Q3
Mesh 1	546.91	905.53	386.96	387.04	387.12	387.19
Mesh 2	1034.07	1658.53	747.60	748.20	749.78	749.40

HF CPU cost in seconds [s] for the HF simulations on the coarse (mesh 1) and the refined mesh (mesh 2)

**Table 3** Mechanical response of a NCB under external loading

	$\max_{\mu \in \mathcal{P}_{\text{train}}} E_{\mu}^{\max}(\cdot)$	$E_{\mu}^{\max}(\cdot)/n_{\text{train}}$	$\max_{\mu \in \mathcal{P}_{\text{train}}} E_{\mu}^{\text{avg}}(\cdot)$	$E_{\mu}^{\text{avg}}(\cdot)/n_{\text{train}}$
$N_{H2}$	$2.86 \cdot 10^{-2}$	$6.55 \cdot 10^{-4}$	$2.86 \cdot 10^{-2}$	$3.91 \cdot 10^{-5}$
$\varepsilon_{tt} - E$	$4.42 \cdot 10^{-2}$	$7.68 \cdot 10^{-3}$	$7.87 \cdot 10^{-3}$	$4.84 \cdot 10^{-3}$
$\varepsilon_{zz} - E$	$5.03 \cdot 10^{-2}$	$1.09 \cdot 10^{-2}$	$1.38 \cdot 10^{-2}$	$6.73 \cdot 10^{-3}$

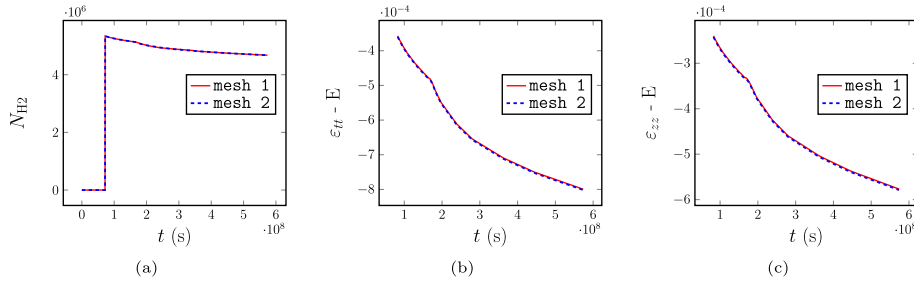
Computation of average (column 1 and 3) and maximum (column 2 and 4) errors over the training set for several errors on the quantities of interest: maximum error over all time steps (cf. (33))

building halfway up the barrel, and is crossed by two vertical and three horizontal cables. We consider an adaptive time-stepping scheme: approximately 45 to 50 time steps are needed to reach the specified final time step, for all parameters in  $\mathcal{P}$  and for both meshes. Table 2 provides an overview of the costs of the HF solver over the training set for the two meshes: we observe that the wall-clock cost of a full HF simulation is roughly nine minutes for the coarse mesh and seventeen minutes for the refined mesh. We consider a 7 by 7 training set  $\mathcal{P}_{\text{train}}$  and a 5 by 5 test set; parameters are logarithmically spaced in both directions.

Figure 8 showcases the evolution of normal forces in the central horizontal cable ( $N_{H2}$ ), the vertical ( $\varepsilon_{zz}$ ) and the tangential ( $\varepsilon_{tt}$ ) deformations on the outer surface of the geometry. Table 3 shows the behavior of the maximum and average relative errors

$$E_{\mu}^{\max} = \frac{\max_{k=1, \dots, K} |q_{\mu}^{\text{hf}(1)}(t^{(k)}) - q_{\mu}^{\text{hf}(2)}(t^{(k)})|}{\max_{k=1, \dots, K} |q_{\mu}^{\text{hf}(2)}(t^{(k)})|}, \quad E_{\mu}^{\text{avg}} = \frac{\sum_{k=1}^K \Delta t^{(k)} |q_{\mu}^{\text{hf}(1)}(t^{(k)}) - q_{\mu}^{\text{hf}(2)}(t^{(k)})| dt}{\sum_{k=1}^K |q_{\mu}^{\text{hf}(2)}(t^{(k)})|} \tag{33}$$

for the three quantities of interest of Fig. 8. We notice that the two meshes lead to nearly-equivalent results for this model problem.



**Fig. 8** Mechanical response of a NCB under external loading. Comparison of the quantities of interest computed for the two meshes of the standard section: **(a)** normal force on a horizontal cable, **(b)** tangential and **(c)** vertical strains on the outer wall of the standard section of the containment building

**Model reduction.** We assess the performance of the Galerkin ROM over training and test sets. Given the sequence of parameters  $\{\mu^{*,it}\}_{it=1}^{\max it}$ , for  $it = 1, \dots, \max it$ ,

1. we solve the HF problem to find the trajectory  $\{u_{\mu^{*,it}}^{(k)}\}_{k=1}^K$ ;
2. we update the reduced space  $\mathcal{Z}$  using (24) with tolerance  $tol = 10^{-5}$ ;
3. we update the quadrature rule using the (incremental) strategy described in “[Extension to unsteady problems](#)”.

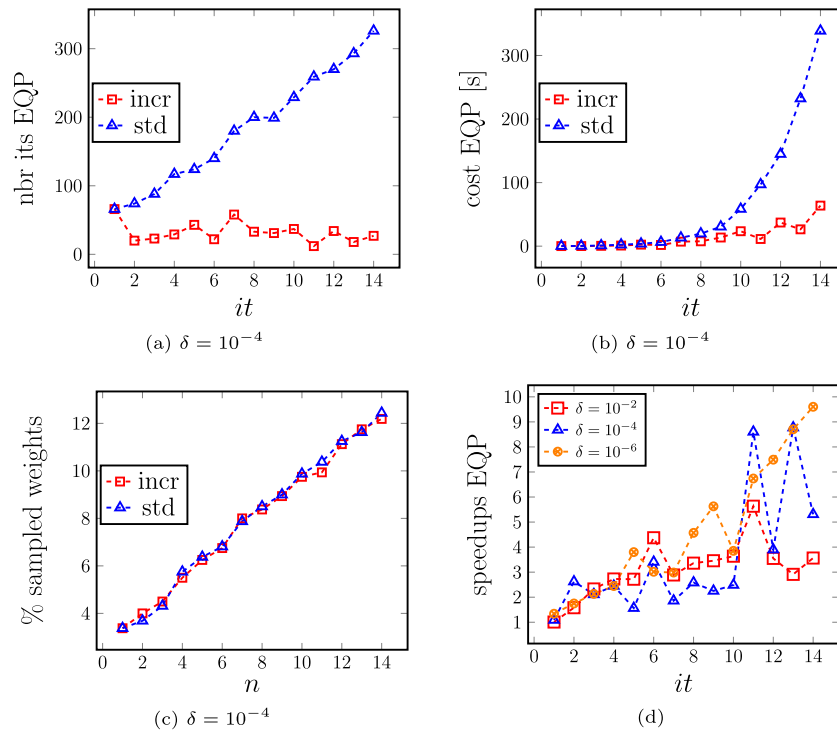
Below, we assess (i) the effectiveness of the incremental strategy for the construction of the quadrature rule, and (ii) the impact of the sampling strategy on performance. Towards this end, we compute the projection error

$$E_{it}^{\text{proj}}(\mathcal{P}) = \max_{\mu \in \mathcal{P}} \frac{\|\approx_{\mu} - \Pi_{\mathcal{Z}_{it}} \approx_{\mu}\|}{\|\approx_{\mu}\|}, \quad \cdot \in \{\text{train, test}\} \tag{34}$$

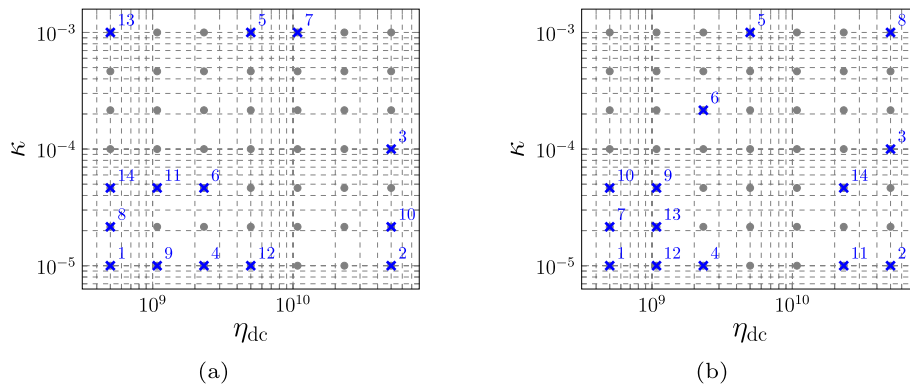
where  $\Pi_{\mathcal{Z}_{it}} \approx_{\mu} := \{\Pi_{\mathcal{Z}_{it}} u_{\mu}^{(k)}\}$  and  $\|\approx_{\mu}\| = \sqrt{\sum_{k=1}^K \Delta t^{(k)} \|v^{(k)}\|^2}$  is the discrete  $L^2(0, T; \|\cdot\|)$  norm—as in [31], we consider  $\|\cdot\| = \|\cdot\|_2$ . Further results on the prediction error and online speedups of the ROM are provided in [31].

Figure 9 illustrates the performance of the EQ procedure; in this test, we consider the finer mesh (mesh 2), and we select the parameters  $\{\mu^{*,it}\}_{it=1}^{\max it=15}$  using the POD-strong greedy algorithm 7 based on the HF results on mesh 1. Figure 9a–c show the number of iterations required by Algorithm 3 to meet the convergence criterion, the computational cost, and the percentage of sampled elements, which is directly related to the online costs, for  $\delta = 10^{-4}$ . As for the previous test case, we observe that the progressive construction of the quadrature rule drastically reduces the NNLS iterations without hindering performance. Figure 9d shows the speedup of the incremental method for three choices of the tolerance  $\delta$  and for several iterations of the iterative procedure. We notice that the speedup increases as the iteration count increases: this can be explained by observing that the percentage of new columns added during the  $it$ -th step in the matrix  $\mathbf{G}$  decays with  $it$  (cf. (25)). We further observe that the speedup increase as we decrease the tolerance  $\delta$ : this is justified by the fact that the number of iterations required by Algorithm 3 increases as we decrease  $\delta$ .

Figures 10 and 11 investigate the efficacy of the greedy sampling strategy. Figure 10 shows the parameters  $\{\mu^{*,j}\}_j$  selected by Algorithm 7 for (a) the coarse mesh (mesh 1) and (b) the refined mesh (mesh 2): we observe that the majority of the sampled parameters are clustered in the bottom left corner of the parameter domain for both meshes.



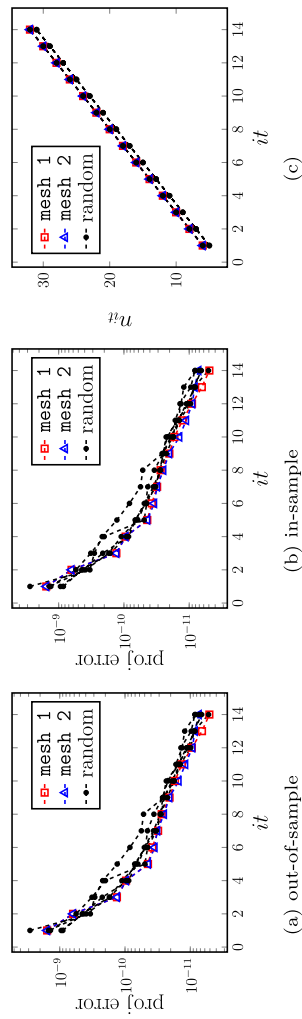
**Fig. 9** Mechanical response of a NCB under external loading. Progressive construction of the quadrature rule on `mesh 2`. **a–c** Performance of standard (“std”) and incremental (“incr”) EQ procedures for several iterations of the greedy procedures, for  $\delta = 10^{-4}$ . **d** Computational speedup for three choices of the tolerance  $\delta$



**Fig. 10** Mechanical response of a NCB under external loading. Parameters  $\{\mu^{*j}\}$  selected by Algorithm 7 for (a) the coarse mesh (`mesh 1`) and (b) the refined mesh (`mesh 2`)

Figure 11 shows the performance (measured through the projection error (34)) of the two samples depicted in Figure 10 on training and test sets; to provide a concrete reference, we also compare performance with five randomly-generated samples. Interestingly, we observe that for this model problem the choice of the sampling strategy has little effects on performance; nevertheless, also in this case, the greedy procedure based on the coarse mesh consistently outperforms random sampling.





**Fig. 11** Mechanical response of a NCB under external loading. Behavior of the projection error  $E_{it}^{proj}$  (34) for parameters selected by Algorithm 7 based on coarse (mesh. 1) and fine data (mesh. 2); comparison with random sampling. **a** Performance on  $\mathcal{P}_{test}$  ( $5 \times 5$ ). **b** Performance on  $\mathcal{P}_{train}$  ( $7 \times 7$ ). **c** Behavior of the basis size  $n_{it}$

## Conclusions

The computation burden of the offline training stage remains an outstanding challenge of MOR techniques for nonlinear, non-parametrically-affine problems. Adaptive sampling of the parameter domain based on greedy methods might contribute to reduce the number of offline HF solves that is needed to meet the target accuracy; however, greedy methods are inherently sequential and introduce non-negligible overhead that might ultimately hinder the benefit of adaptive sampling. To address these issues, in this work, we proposed two new strategies to accelerate greedy methods: first, a progressive construction of the ROM based on HAPOD to speed up the construction of the empirical test space for LSPG ROMs and on a warm start of the NNLS algorithm to determine the empirical quadrature rule; second, a two-fidelity sampling strategy to reduce the number of expensive greedy iterations.

The numerical results of “[Numerical results](#)” illustrate the effectiveness and the generality of our methods for both steady and unsteady problems. First, we found that the warm start of the NNLS algorithm enables a non-negligible reduction of the computational cost without hindering performance. Second, we observed that the sampling strategy based on coarse data leads to near-optimal performance: this result suggests that multi-fidelity algorithms might be particularly effective to explore the parameter domain in the MOR training phase.

The empirical findings of this work motivate further theoretical and numerical investigations that we wish to pursue in the future. First, we wish to analyze the performance of multi-fidelity sampling methods for MOR: the ultimate goal is to devise *a priori* and *a posteriori* indicators to drive the choice of the mesh hierarchy and the cardinality  $n_0$  of the set  $\mathcal{P}_\star$  in Algorithm 6. Second, we plan to extend the two elements of our formulation—progressive ROM generation and multi-fidelity sampling—to optimization problems for which the primary objective of model reduction is to estimate a suitable quantity of interest (*goal-oriented MOR*): in this respect, we envision to combine our formulation with adaptive techniques for optimization [33–35].

## Acknowledgements

The authors thank Dr. Jean-Philippe Argaud, Dr. Guilhem Ferté (EDF R&D) and Dr. Michel Bergmann (Inria Bordeaux) for fruitful discussions. The first author thanks the code aster development team for fruitful discussions on the FE software employed for the numerical simulations of section 4.2.

## Author contributions

Eki Agouzal: methodology, software, investigation. Tommaso Taddei: conceptualization, methodology, software, investigation, writing.

## Funding

This work was partly funded by ANRT (French National Association for Research and Technology) and EDF.

## Availability of data and materials.

The data that support the findings of this study are available from the corresponding author, TT, upon reasonable request. The software developed for the investigations of section 4.2 is expected to be integrated in the open-source Python library `MORdicus` [36], which is funded by a “French Fonds Unique Interministériel” (FUI) project.

## Declarations

### Competing interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Received: 17 January 2024 Accepted: 24 March 2024

Published online: 18 April 2024

## References

1. Carlberg K, Farhat C, Cortial J, Amsallem D. The GNAT method for nonlinear model reduction: effective implementation and application to computational fluid dynamics and turbulent flows. *J Comput Phys.* 2013;242:623–47.
2. Taddei T, Zhang L. Space-time registration-based model reduction of parameterized one-dimensional hyperbolic PDEs. *ESAIM Math Model Numer Anal.* 2021;55(1):99–130.
3. Ryckelynck D. Hyper-reduction of mechanical models involving internal variables. *Int J Numer Method Eng.* 2009;77(1):75–89.
4. Farhat C, Chapman T, Avery P. Structure-preserving, stability, and accuracy properties of the energy-conserving sampling and weighting method for the hyper reduction of nonlinear finite element dynamic models. *Int J Numer Method Eng.* 2015;102(5):1077–110.
5. Yano M, Patera AT. An LP empirical quadrature procedure for reduced basis treatment of parametrized nonlinear PDEs. *Comput Methods Appl Mech Eng.* 2019;344:1104–23.
6. Veroy K, Prud'Homme C, Rovas D, Patera A. A posteriori error bounds for reduced-basis approximation of parametrized noncoercive and nonlinear elliptic partial differential equations. In: 16th AIAA Computational Fluid Dynamics Conference, p. 3847, 2003.
7. Haasdonk B, Ohlberger M. Reduced basis method for finite volume approximations of parametrized linear evolution equations. *ESAIM Math Model Numer Anal.* 2008;42(2):277–302.
8. Sirovich L. Turbulence and the dynamics of  $\Gamma$ . Coherent structures. *Coherent Struct Quart Appl Math.* 1987;45(3):561–71.
9. Volkwein S. Model reduction using proper orthogonal decomposition. Lecture Notes, Institute of Mathematics and Scientific Computing, University of Graz. see <http://www.uni-graz.at/imawww/volkwein/POD.pdf>. 1025; 2011.
10. Cohen A, DeVore R. Approximation of high-dimensional parametric PDEs. *Acta Numer.* 2015;24:1–159.
11. Himpe C, Leibner T, Rave S. Hierarchical approximate proper orthogonal decomposition. *SIAM J Sci Comput.* 2018;40(5):3267–92.
12. Quarteroni A, Manzoni A, Negri F. Reduced basis methods for partial differential equations: an introduction, vol. 92. Berlin: Springer; 2015.
13. Chapman T, Avery P, Collins P, Farhat C. Accelerated mesh sampling for the hyper reduction of nonlinear computational models. *Int J Numer Method Eng.* 2017;109(12):1623–54.
14. Feng L, Lombardi L, Antonini G, Benner P. Multi-fidelity error estimation accelerates greedy model reduction of complex dynamical systems. *Int J Numer Method Eng.* 2023;124(3):5312–33.
15. Paul-Dubois-Taine A, Amsallem D. An adaptive and efficient greedy procedure for the optimal training of parametric reduced-order models. *Int J Numer Method Eng.* 2015;102(5):1262–92.
16. Barral N, Taddei T, Tifouti I. Registration-based model reduction of parameterized PDEs with spatio-parameter adaptivity. *J Comput Phys.* 2023;112727.
17. Benaceur A, Ehrlicher V, Ern A, Meunier S. A progressive reduced basis/empirical interpolation method for nonlinear parabolic problems. *SIAM J Sci Comput.* 2018;40(5):2930–55.
18. Barrault M, Maday Y, Nguyen NC, Patera AT. An empirical interpolation method: application to efficient reduced-basis discretization of partial differential equations. *CR Math.* 2004;339(9):667–72.
19. Conti P, Guo M, Manzoni A, Frangi A, Brunton SL, Kutz JN. Multi-fidelity reduced-order surrogate modeling. arXiv preprint [arXiv:2309.00325](https://arxiv.org/abs/2309.00325) 2023.
20. Yano M, Modisette J, Darmofal D. The importance of mesh adaptation for higher-order discretizations of aerodynamic flows. In: 20th AIAA Computational Fluid Dynamics Conference, p. 3852, 2011.
21. Lawson CL, Hanson RJ. Solving least squares problems. SIAM, 1995.
22. Yano M. Discontinuous Galerkin reduced basis empirical quadrature procedure for model reduction of parametrized nonlinear conservation laws. *Adv Comput Math.* 2019;45(5):2287–320.
23. Du E, Yano M. Efficient hyperreduction of high-order discontinuous Galerkin methods: element-wise and point-wise reduced quadrature formulations. *J Comput Phys.* 2022;466: 111399.
24. Urban K, Patera A. An improved error bound for reduced basis approximation of linear parabolic problems. *Math Comput.* 2014;83(288):1599–615.
25. Haasdonk B. Reduced basis methods for parametrized PDEs—a tutorial introduction for stationary and instationary problems. *Model Reduct Approx Theory Algo.* 2017;15:65.
26. Iollo A, Sambataro G, Taddei T. An adaptive projection-based model reduction method for nonlinear mechanics with internal variables: application to thermo-hydro-mechanical systems. *Int J Numer Method Eng.* 2022;123(12):2894–918.
27. Brand M. Fast online svd revisions for lightweight recommender systems. In: Proceedings of the 2003 SIAM International Conference on Data Mining, pp. 37–46, 2003, SIAM.
28. MATLAB: R2022a. The MathWorks Inc., Natick, Massachusetts. 2022.
29. Finite Element code\_aster, Analysis of Structures and Thermomechanics for Studies and Research. Electricité de France (EDF), Open source on [www.code-aster.org](http://www.code-aster.org) (1989–2024).
30. Taddei T. Compositional maps for registration in complex geometries. arXiv preprint [arXiv:2308.15307](https://arxiv.org/abs/2308.15307) 2023.
31. Agouzal E, Argaud J-P, Bergmann M, Ferté G, Taddei T. Projection-based model order reduction for prestressed concrete with an application to the standard section of a nuclear containment building. arXiv preprint [arXiv:2401.05098](https://arxiv.org/abs/2401.05098) 2024.
32. Bouhjiti D. Analyse probabiliste de la fissuration et du confinement des grands ouvrages en béton armé et précontraint-application aux enceintes de confinement des réacteurs nucléaires (cas de la maquette vercors). *Acad J Civil Eng.* 2018;36(1):464–71.
33. Zahr MJ, Farhat C. Progressive construction of a parametric reduced-order model for PDE-constrained optimization. *Int J Numer Method Eng.* 2015;102(5):1111–35.
34. Alexandrov NM, Lewis RM, Gumbert CR, Green LL, Newman PA. Approximation and model management in aerodynamic optimization with variable-fidelity models. *J Aircraft.* 2001;38(6):1093–101.
35. Yano M, Huang T, Zahr MJ. A globally convergent method to accelerate topology optimization using on-the-fly model reduction. *Comput Methods Appl Mech Eng.* 2021;375: 113635.

36. Mordicus Python Package. Consortium of the FUI Project MOR DICUS. Electricité de France (EDF), Open source on <https://gitlab.com/mordicus/mordicus>. 2022.

### **Publisher's Note**

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.