HAL
open science

A progressive reduced basis/empirical interpolation method for nonlinear parabolic problems

Amina Benaceur, Virginie Ehrlacher, Alexandre Ern, Sébastien Meunier

To cite this version:

Amina Benaceur, Virginie Ehrlacher, Alexandre Ern, Sébastien Meunier. A progressive reduced basis/empirical interpolation method for nonlinear parabolic problems. SIAM Journal on Scientific Computing, 2018, 40 (5), pp.A2930-A2955. hal-01599304v4

HAL Id: hal-01599304

https://hal.science/hal-01599304v4

Submitted on 17 Jul 2018

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.

A progressive reduced basis/empirical interpolation method for nonlinear parabolic problems*

Amina Benaceur[†‡], Virginie Ehrlacher[†], Alexandre Ern[†] and Sébastien Meunier[‡]

Abstract

We investigate new developments of the combined Reduced-Basis and Empirical Interpolation Methods (RB-EIM) for parametrized nonlinear parabolic problems. In many situations, the cost of the EIM in the offline stage turns out to be prohibitive since a significant number of nonlinear time-dependent problems need to be solved using the high-fidelity (or full-order) model. In the present work, we develop a new methodology, the Progressive RB-EIM (PREIM) method for nonlinear parabolic problems. The purpose is to reduce the offline cost while maintaining the accuracy of the RB approximation in the online stage. The key idea is a progressive enrichment of both the EIM approximation and the RB space, in contrast to the standard approach where the EIM approximation and the RB space are built separately. PREIM uses high-fidelity computations whenever available and RB computations otherwise. Another key feature of each PREIM iteration is to select twice the parameter in a greedy fashion, the second selection being made after computing the high-fidelity trajectory for the firstly selected value of the parameter. Numerical examples are presented on nonlinear heat transfer problems.

1 Introduction

The Reduced-Basis (RB) method devised in [14, 17] (see also the recent textbooks [11, 18]) is a computationally effective approach to approximate parametrized Partial Differential Equations (PDEs) encountered in many problems in science and engineering. For instance, the RB method is often used in real-time simulations, where a problem needs to be solved very quickly under limited computational resources, or in multi-query simulations, where a problem has to be solved repeatedly for a large number of parameter values. Let $\mathcal{P}$ denote the parameter set. The RB method is split into two stages: (i) an offline stage where a certain number of so-called High-Fidelity (HF) trajectories are computed for a training subset of parameters $\mathcal{P}^{\mathrm{tr}} \subsetneq \mathcal{P}$(typically a finite element space based on a fine mesh); (ii) an online stage for real-time or multi-query simulations where the parameter set $\mathcal{P}$ is explored more extensively. The output of the offline phase includes an approximation space of small dimension spanned by the so-called RB functions. The reduced space then replaces the much larger HF space in the online stage. The crucial point for the computational efficiency of the overall procedure is that computations in the HF space are allowed only in the offline stage.

*This work is partially supported by Electricité De France (EDF) and a CIFRE PhD fellowship from ANRT

[†]University Paris-Est, CERMICS (ENPC), 77455 Marne la Vallée Cedex 2 and INRIA Paris, 75589 Paris, France

[‡]EDF Lab Les Renardières, 77250 Ecuelles Cedex, France

In the present work, we are interested in nonlinear parabolic problems for which a RB method has been successfully developed in [6, 7]. A key ingredient to treat the nonlinearity so that the online stage avoids HF computations is the Empirical Interpolation Method (EIM) [1, 15]. The EIM provides an approximation of the nonlinear (or non-affine) terms in the PDE. This approximation is built using a greedy algorithm as the sum of M functions, where the dependence on the space variable is separated from the dependence on the parameter (and the time variable for parabolic problems). The integer M is called the rank of the EIM and controls the accuracy of the approximation. Although the EIM is performed during the offline stage of the RB method, its cost can become a critical issue since the EIM can require an important number of HF computations for an accurate approximation of the nonlinearity. The cost of the EIM typically scales with the size of the training set $\mathcal{P}^{\mathrm{tr}}$.

The goal of the present work is to overcome this issue. To this purpose, we devise a new methodology, the Progressive RB-EIM (PREIM) method, which aims at reducing the computational cost of the offline stage while maintaining the accuracy of the RB approximation in the online stage. The key idea is a progressive enrichment of both the EIM approximation and the RB space, in contrast to the standard approach, where the EIM approximation and the RB space are built separately. In PREIM, the number of HF computations is at most M, and it is in general much lower than M in a time-dependent context where the greedy selection of the pair (μ, k) to build the EIM approximation (where μ is the parameter and k refers to the discrete time node) can lead to repeated values of μ for many different values of k. In other words, PREIM can select multiple space fields within the same HF trajectory to build the EIM space functions. In this context, only a modest number of HF trajectories needs to be computed, yielding significant computational savings with respect to the standard offline stage. PREIM is driven by convergence criteria on the quality of both the EIM and the RB approximation, as in the standard RB-EIM procedure. PREIM is devised in order to have a guaranteed termination, and in the worst-case scenario, the same number of HF trajectories is computed as in the standard RB-EIM algorithm, thus reaching the same level of accuracy for the representation of the nonlinearity and the construction of the RB functions (if this level of accuracy turns out to be insufficient, the parameter training set has to be enlarged as usual in the standard algorithm). In this worst-case scenario, the computational cost of PREIM may be slightly larger than that of the standard algorithm because of the way the intermediate calculations of trajectories are organized in PREIM. However, we expect that in many practical situations, e.g., when the computation of HF trajectories dominates the cost of the progressive construction of the EIM, PREIM can bring computational benefits with respect to the standard approach. These benefits, which are particularly sizeable whenever the nonlinearity can be represented by an EIM approximation of relatively modest rank, are illustrated in this work on three test cases, including one derived from a three-dimensional industrial prototype. Yet, the present study remains heuristic, and a theoretical analysis of the possible computational gains of PREIM can be pursued in future work.

The idea of a progressive enrichment of both the EIM approximation and the RB space has been recently proposed in [2] for stationary nonlinear PDEs, where it is called Simultaneous EIM/RB (SER). Thus, PREIM extends this idea to time-dependent PDEs. In addition, there is an important difference in the greedy algorithms between SER and PREIM. Whereas SER uses only RB computations, PREIM uses HF computations whenever available, both for the greedy selection of the parameters and the time nodes, as well as for the space-dependent functions in the EIM approximation. These aspects are particularly relevant since they improve the accuracy of the EIM approximation. This is illustrated in our numerical experiments on nonlinear parabolic PDEs. The progressive construction of the EIM and the RB has been recently addressed within the Empirical Interpolation Operator Method in [3]. Therein, the enrichment criterion is common to both the EIM

and the RB, and the snapshot maximizing an a posteriori error estimator is selected to enrich both bases. Instead, PREIM has dedicated criteria for the quality of the EIM approximation and for the RB approximation. Furthermore, PREIM systematically exploits the knowledge of the HF trajectories whenever available, and an update step is performed in order to confirm the current parameter selection. We also mention the Proper Orthogonal Empirical Interpolation Method from [20], where a progressive construction of the EIM approximation is devised using POD-based approximations of the HF trajectories.

The paper is organized as follows. In Section 2, we introduce the model problem. In Section 3, we briefly recall the main ideas of the nonlinear RB method devised in [6, 7], and in Section 4, we briefly recall the EIM procedure in the standard offline stage as devised in [1, 15]. The reader familiar with the material can jump directly to Section 5, where PREIM is introduced and discussed. Section 6 presents numerical results illustrating the performance of PREIM on nonlinear parabolic problems related to heat transfer including a three-dimensional valve prototpye for flow regulation. Finally, Section 7 draws some conclusions and outlines some perspectives.

2 Model problem

In this section, we present a prototypical example of a nonlinear parabolic PDE. The methodology we propose is illustrated on this model problem but can be extended to other types of parabolic equations. We consider a spatial domain (open, bounded, connected subset) $\Omega \subset \mathbb{R}^d$, $d \geq 1$, with a Lipschitz boundary, a finite time interval $I = [0, T]$, with $T > 0$, and a parameter set $\mathcal{P} \subset \mathbb{R}^p$, $p \geq 1$, whose elements are generically denoted by $\mu \in \mathcal{P}$. Our goal is to solve the following nonlinear parabolic PDE for many values of the parameter $\mu \in \mathcal{P}$: find $u_\mu : I \times \Omega \to \mathbb{R}$ such that

$$\begin{cases} \dfrac{\partial u_\mu}{\partial t} - \nabla \cdot \big((\kappa_0 + \Gamma(\mu, u_\mu))\nabla u_\mu\big) = f, & \text{in } I \times \Omega, \\ -\big(\kappa_0 + \Gamma(\mu, u_\mu)\big)\dfrac{\partial u_\mu}{\partial n} = \phi_e, & \text{on } I \times \partial\Omega, \\ u_\mu(t = 0, \cdot) = u_0(\cdot), & \text{in } \Omega, \end{cases} \tag{1}$$

where $\kappa_0 > 0$ is a fixed positive real number, $\Gamma : \mathcal{P} \times \mathbb{R} \to \mathbb{R}$ is a given nonlinear function, $f : I \times \Omega \to \mathbb{R}$ is the source term, $\phi_e : I \times \partial\Omega \to \mathbb{R}$ is the time-dependent Neumann boundary condition on $\partial\Omega$, and $u_0 : \Omega \to \mathbb{R}$ is the initial condition. For simplicity, we assume without loss of generality that f, ϕ_e, and u_0 are parameter-independent. We assume that $f \in L^2(I; L^2(\Omega))$ and $\phi_e \in L^2(I; L^2(\partial\Omega))$ (this means that $f(t) \in L^2(\Omega)$ and $\phi_e(t) \in L^2(\partial\Omega)$ for (almost every) $t \in I$), and we also assume that $u_0 \in H^1(\Omega)$. We make the standard uniform ellipticity assumption $\beta_1 \leq \kappa_0 + \Gamma(\mu, z) \leq \beta_2$ with $0 < \beta_1 < \beta_2 < \infty$, for all $(\mu, z) \in \mathcal{P} \times \mathbb{R}$. With the above assumptions, it is reasonable to look for a weak solution $u_\mu \in L^2(I; Y) \cap H^1(I; Y')$.

Remark 1 (Initial condition) *For parabolic PDEs, the initial condition is often taken to be in a larger space, e.g., $u_0 \in L^2(\Omega)$. Our assumption that $u_0 \in Y$ is motivated by the RB method, where basis functions in Y are sought as solution snapshots in time and for certain parameter values. In this context, we want to include the possibility to select the initial condition as a RB function.*

Remark 2 (Heat transfer) *One important application we have in mind for* (1) *is heat transfer problems. In this context, the PDE can take the slightly more general form*

$$\alpha(u_\mu)\frac{\partial u_\mu}{\partial t} - \nabla \cdot \big((\kappa_0 + \Gamma(\mu, u_\mu))\nabla u_\mu\big) = f, \quad \text{in } I \times \Omega,$$

where $\alpha(u_\mu)$ stands for the mass density times the heat capacity. Moreover, the quantity $(\kappa_0+\Gamma(\mu,u_\mu))$ represents the thermal conductivity. Note also that $\phi_e > 0$ means that the system is heated.

In practice, one way to solve (1) is to use a Y-conforming Finite Element Method [5] to discretize in space and a time-marching scheme to discretize in time. The Finite Element Method is based on a finite element subspace $X \subsetneq Y$ defined on a discrete nodal subset $\Omega^{\rm tr} \subsetneq \Omega$, where $\mathrm{Card}(\Omega^{\rm tr}) = \mathcal{N}$. To discretize in time, we consider an integer $K \geq 1$, we let $0 = t^0 < \cdots < t^K = T$ be $(K+1)$ distinct time nodes over I, and we set $\mathbb{K}^{\rm tr} = \{1, \cdots, K\}$, $\overline{\mathbb{K}}^{\rm tr} = \{0\} \cup \mathbb{K}^{\rm tr}$, $I^{\rm tr} = \{t^k\}_{k\in\overline{\mathbb{K}}^{\rm tr}}$, and $\Delta t^k = t^k - t^{k-1}$ for all $k \in \mathbb{K}^{\rm tr}$. As is customary with the RB method, we assume henceforth that the mesh-size and the time-steps are small enough so that the above space-time discretization method delivers HF approximate trajectories within the desired level of accuracy. These trajectories, which then replace the exact trajectories solving (1), are still denoted u_μ for all $\mu \in \mathcal{P}$. Henceforth, we use the convention that the superscript k always indicates a time index; thus, we write $u_\mu^k(\cdot) = u_\mu(t^k, \cdot) \in X$, $f^k(\cdot) = f(t^k, \cdot) \in L^2(\Omega)$, and $\phi_e^k(\cdot) = \phi_e(t^k, \cdot) \in L^2(\partial\Omega)$. Applying a semi-implicit Euler scheme, our goal is, given $u_\mu^0 = u_0 \in X$, to find $(u_\mu^k)_{k\in\mathbb{K}^{\rm tr}} \in X^K$ such that, for all $k \in \mathbb{K}^{\rm tr}$,

$$\forall v \in X,\ \ m(u_\mu^k, v) + \Delta t^k a_0\big(u_\mu^k, v\big) + \Delta t^k n_\Gamma\big(\mu, u_\mu^{k-1}, v\big) = m(u_\mu^{k-1}, v) + \Delta t^k l^k(v), \tag{2}$$

with the bilinear forms $m : Y \times Y \to \mathbb{R}$, $a_0 : Y \times Y \to \mathbb{R}$ and the linear forms $l^k : Y \to \mathbb{R}$ such that

$$m(v, w) = \int_\Omega vw, \qquad a_0(v, w) = \kappa_0 \int_\Omega \nabla v \cdot \nabla w, \qquad l^k(v) = \int_\Omega f^k v + \int_{\partial\Omega} \phi_e^k v, \tag{3}$$

and the nonlinear form $n_\Gamma : \mathcal{P} \times Y \times Y \to \mathbb{R}$ such that

$$n_\Gamma(\mu, v, w) = \int_\Omega \Gamma(\mu, v) \nabla v \cdot \nabla w, \tag{4}$$

for all $\mu \in \mathcal{P}$ and all $v, w \in Y$. In (2), the nonlinearity is treated explicitly, whereas the diffusive term is treated implicitly. This choice avoids dealing with a nonlinear solver at each time-step. The computation of derivatives of discrete operators within Newton's method is addressed, e.g., in [3].

3 The Reduced-Basis method

In this section, we briefly recall the Reduced-Basis (RB) method for the nonlinear problem (2) [7, 6]. Let $\hat{X}_N \subset X$ be a so-called reduced subspace such that $N = \dim(\hat{X}_N) \ll \dim(X) = \mathcal{N}$. Let $(\theta_n)_{1\leq n\leq N}$ be a Y-orthonormal basis of $\hat{X}_N$. For all $\mu \in \mathcal{P}$ and $k \in \overline{\mathbb{K}}^{\rm tr}$, the RB solution $\hat{u}_\mu^k \in \hat{X}_N$ that approximates the HF solution $u_\mu^k \in X$ is decomposed as

$$\hat{u}_\mu^k = \sum_{n=1}^N \hat{u}_{\mu,n}^k \theta_n, \tag{5}$$

with real numbers $\hat{u}_{\mu,n}^k$ for all $n \in \{1, \ldots, N\}$. Let us introduce the component vector $\hat{\mathbf{u}}_\mu^k := (\hat{u}_{\mu,n}^k)_{1\leq n\leq N} \in \mathbb{R}^N$, for all $\mu \in \mathcal{P}$ and $k \in \overline{\mathbb{K}}^{\rm tr}$. Let $\hat{u}^0$ be the Y-orthogonal projection of the initial condition $u_0 \in X$ onto $\hat{X}_N$ with associated component vector $\hat{\mathbf{u}}^0 \in \mathbb{R}^N$. Replacing $u_\mu^k \in X$ in the weak form (2) by the approximation $\hat{u}_\mu^k \in \hat{X}_N$ with associated component vector $\hat{\mathbf{u}}_\mu^k \in \mathbb{R}^N$, and

using the test functions $(\theta_p)_{1\le p\le N}$, we obtain the following problem written in algebraic form: Given $\hat{\mathbf{u}}_\mu^0 = \hat{\mathbf{u}}^0 \in \mathbb{R}^N$, find $(\hat{\mathbf{u}}_\mu^k)_{k\in\mathbb{K}^{\rm tr}} \in (\mathbb{R}^N)^K$ such that, for all $k \in \mathbb{K}^{\rm tr}$,

$$(\mathbf{M} + \Delta t^k \mathbf{A}_0)\hat{\mathbf{u}}_\mu^k = \Delta t^k \mathbf{f}^k + \mathbf{M}\hat{\mathbf{u}}_\mu^{k-1} - \Delta t^k \mathbf{g}(\hat{\mathbf{u}}_\mu^{k-1}), \tag{6}$$

with the matrices $\mathbf{M}, \mathbf{A}_0 \in \mathbb{R}^{N\times N}$ and the vectors $\mathbf{f}^k \in \mathbb{R}^N$ such that

$$\mathbf{M} = \Big(m(\theta_n,\theta_p)\Big)_{1\le p,n\le N}, \qquad \mathbf{A}_0 = \Big(a_0(\theta_n,\theta_p)\Big)_{1\le p,n\le N}, \qquad \mathbf{f}^k = \big(l^k(\theta_p)\big)_{1\le p\le N}, \tag{7}$$

and the vector $\mathbf{g}(\hat{\mathbf{u}}_\mu^{k-1}) \in \mathbb{R}^N$ such that

$$\mathbf{g}(\hat{\mathbf{u}}_\mu^{k-1}) = \left(\sum_{n=1}^N \hat{u}_{\mu,n}^{k-1} \int_\Omega \Gamma\left(\mu, \sum_{n'=1}^N \hat{u}_{\mu,n'}^{k-1}\theta_{n'}\right)\nabla\theta_n\cdot\nabla\theta_p\right)_{1\le p\le N}. \tag{8}$$

The difficulty is that the computation of $\mathbf{g}(\hat{\mathbf{u}}_\mu^{k-1})$ requires a parameter-dependent reconstruction using the RB functions in order to compute the integral over Ω. To avoid this, we need to build an approximation γ_M of the nonlinear function $\gamma : \mathcal{P}\times\overline{\mathbb{K}}^{\rm tr}\times\Omega \to \mathbb{R}$ such that

$$\gamma(\mu,k,x) := \Gamma(\mu, u_\mu^k(x)), \tag{9}$$

in such a way that the dependence on x is separated from the dependence on (μ,k). More precisely, for some integer $M > 0$, we are looking for an (accurate) approximation $\gamma_M : \mathcal{P}\times\overline{\mathbb{K}}^{\rm tr}\times\Omega \to \mathbb{R}$ of γ under the separated form

$$\gamma_M(\mu,k,x) := \sum_{j=1}^M \varphi_{\mu,j}^k q_j(x), \tag{10}$$

where M is called the rank of the approximation and $\varphi_{\mu,j}^k$ are real numbers that we find by interpolation over a set of M points $\{x_1,\dots,x_M\}$ in $\Omega^{\rm tr}$ by requiring that

$$\gamma_M(\mu,k,x_i) = \gamma(\mu,k,x_i) = \Gamma(\mu,u_\mu^k(x_i)), \quad \forall i\in\{1,\cdots,M\}. \tag{11}$$

The interpolation property (11) is achieved by setting

$$\varphi_{\mu,j}^k = (\mathbf{B}^{-1}\boldsymbol{\gamma}_\mu^k)_j,\ \forall j\in\{1,\cdots,M\}, \quad \text{where} \quad \boldsymbol{\gamma}_\mu^k := \big(\Gamma(\mu,u_\mu^k(x_i))\big)_{1\le i\le M} \in \mathbb{R}^M, \tag{12}$$

and $\mathbf{B} = (q_j(x_i))_{1\le i,j\le M} \in \mathbb{R}^{M\times M}$ must be an invertible matrix. Therefore, (10) can be rewritten as follows:

$$\gamma_M(\mu,k,x) = \sum_{j=1}^M (\mathbf{B}^{-1}\boldsymbol{\gamma}_\mu^k)_j q_j(x). \tag{13}$$

The points $(x_i)_{1\le i\le M}$ in $\Omega^{\rm tr}$ and the functions $(q_j)_{1\le j\le M}$ defined on Ω are determined by the EIM algorithm [1] which is further described in Section 4 below.

Let us now describe how we can use the EIM approximation (13) to allow for an offline/online decomposition of the computation of the vector $\mathbf{g}(\hat{\mathbf{u}}_\mu^{k-1})$ defined in (8). Under the (reasonable) assumptions $\hat{u}_\mu^k \approx u_\mu^k$ and $\Gamma(\mu,\hat{u}_\mu^k(x)) \approx \Gamma\big(\mu,u_\mu^k(x)\big)$, we obtain

$$\Gamma(\mu,\hat{u}_\mu^k(x)) \approx \Gamma\big(\mu,u_\mu^k(x)\big) = \gamma(\mu,k,x) \approx \gamma_M(\mu,k,x) = \sum_{j=1}^M (\mathbf{B}^{-1}\boldsymbol{\gamma}_\mu^k)_j q_j(x) \approx \sum_{j=1}^M (\mathbf{B}^{-1}\hat{\boldsymbol{\gamma}}_\mu^k)_j q_j(x) \tag{14}$$

with the vector $\hat{\boldsymbol{\gamma}}_\mu^k := (\Gamma(\mu, \hat{u}_\mu^k(x_i)))_{1\le i\le M} \in \mathbb{R}^M$. The problem (6) then becomes: Given $\hat{\mathbf{u}}_\mu^0 = \hat{\mathbf{u}}^0 \in \mathbb{R}^N$, find $(\hat{\mathbf{u}}_\mu^k)_{k\in\mathbb{K}^{\rm tr}} \in (\mathbb{R}^N)^K$ such that, for all $k \in \mathbb{K}^{\rm tr}$,

$$(\mathbf{M} + \Delta t^k \mathbf{A}_0)\hat{\mathbf{u}}_\mu^k = \Delta t^k \mathbf{f}^k + \big(\mathbf{M} - \Delta t^k \mathbf{D}_\mu^{k-1}\big)\hat{\mathbf{u}}_\mu^{k-1}, \tag{15}$$

with the matrix $\mathbf{D}_\mu^{k-1} \in \mathbb{R}^{N\times N}$ such that

$$\mathbf{D}_\mu^{k-1} = \sum_{j=1}^{M} \big(\mathbf{B}^{-1}\hat{\boldsymbol{\gamma}}_\mu^{k-1}\big)_j \mathbf{C}^j \quad \text{where} \quad \mathbf{C}^j = \left(\int_\Omega q_j \nabla\theta_n \cdot \nabla\theta_p\right)_{1\le p,n\le N} \in \mathbb{R}^{N\times N},\ \forall 1 \le j \le M. \tag{16}$$

The overall computational procedure can now be split into two stages:

(i) An offline stage where one precomputes on the one hand the RB functions $(\theta_n)_{1\le n\le N}$ leading to the vectors $\hat{\mathbf{u}}^0 \in \mathbb{R}^N$, $(\mathbf{f}^k)_{k\in\mathbb{K}^{\rm tr}} \in (\mathbb{R}^N)^K$ and the matrices $\mathbf{M}, \mathbf{A}_0 \in \mathbb{R}^{N\times N}$, and on the other hand the EIM points $(x_i)_{1\le i\le M}$ and the functions $(q_j)_{1\le j\le M}$ leading to the matrices $\mathbf{B} \in \mathbb{R}^{M\times M}$ and $\mathbf{C}^j \in \mathbb{R}^{N\times N}$, for all $j \in \{1, \dots, M\}$. The offline stage is discussed in more detail in Section 4.

(ii) An online stage to be performed each time one wishes to compute a new trajectory for a parameter $\mu \in \mathcal{P}$. All what remains to be performed is to compute the vector $\widehat{\boldsymbol{\gamma}}_\mu^{k-1} \in \mathbb{R}^M$ and the matrix $\mathbf{D}_\mu^{k-1} \in \mathbb{R}^{N\times N}$ and to solve the N-dimensional linear problem (15) for all $k \in \mathbb{K}^{\rm tr}$. The online stage is summarized in Algorithm 1.

Algorithm 1 Online stage

Input : μ, $(\theta_n)_{1\le n\le N}$, $\hat{\mathbf{u}}^0$, $(\mathbf{f}^k)_{k\in\mathbb{K}^{\rm tr}}$, $\mathbf{M}$, $\mathbf{A}_0$, $(x_i)_{1\le i\le M}$, $(q_j)_{1\le j\le M}$, and $(\mathbf{C}^j)_{1\le j\le M}$

Set $k = 1$ and $\hat{\mathbf{u}}_\mu^0 = \hat{\mathbf{u}}^0$
while $k \in \mathbb{K}^{\rm tr}$ **do**
  Compute $\mathbf{D}_\mu^{k-1}$ using (16) and $\hat{\mathbf{u}}_\mu^{k-1}$
  Solve the reduced system (15) to obtain $\hat{\mathbf{u}}_\mu^k$
  Set $k = k + 1$
end while

Output : $(\hat{\mathbf{u}}_\mu^k)_{k\in\mathbb{K}^{\rm tr}}$

4 The standard offline stage

There are two tasks to be performed during the offline stage:

(T_1) Build the rank-M EIM approximation (10) of the nonlinear function γ defined by (9);

(T_2) Explore the solution manifold in order to construct a linear subspace $\hat{X}_N \subset X$ of dimension N.

In the standard offline stage, these two tasks are performed independently.

Let us first discuss Task (T_1), i.e., the construction of the rank-M EIM approximation. Recall from Section 3 that the goal is to find the interpolation points $(x_i)_{1\le i\le M}$ in $\Omega^{\rm tr} \subsetneq \Omega$ and the functions $(q_j)_{1\le j\le M}$ with $q_j : \Omega \to \mathbb{R}$. The construction of the EIM approximation additionally uses a training

Algorithm 2 Standard EIM

Input : $\mathcal{P}^{\rm tr}$, $\overline{\mathbb{K}}^{\rm tr}$, $\Omega^{\rm tr}$, and $\epsilon_{\rm EIM} > 0$

Compute $\mathcal{S} = (u_\mu^k)_{(\mu,k)\in\mathcal{P}^{\rm tr}\times\overline{\mathbb{K}}^{\rm tr}}$ `P HF trajectories`

Set $m = 1$ and $\gamma_0 \equiv 0$

Search $(\mu_m, k_m) \in \underset{(\mu,k)\in\mathcal{P}^{\rm tr}\times\overline{\mathbb{K}}^{\rm tr}}{\rm argmax} \|\Gamma(\mu, u_\mu^k(\cdot)) - \gamma_{m-1}(\mu, k, \cdot)\|_{\ell^\infty(\Omega^{\rm tr})}$

Set $r_m(\cdot) := \Gamma(\mu_m, u_{\mu_m}^{k_m}(\cdot)) - \gamma_{m-1}(\mu_m, k_m, \cdot)$ and $x_m \in \underset{x\in\Omega^{\rm tr}}{\rm argmax}\, |r_m(x)|$

while $(|r_m(x_m)| > \epsilon_{\rm EIM})$ **do**

  Set $q_m := r_m/r_m(x_m)$ and compute $(\mathbf{B}_{mi})_{1\le i\le m}$ by setting $\mathbf{B}_{mi} := (q_i(x_m))$

  Set $m = m + 1$

  Search $(\mu_m, k_m) \in \underset{(\mu,k)\in\mathcal{P}^{\rm tr}\times\overline{\mathbb{K}}^{\rm tr}}{\rm argmax} \|\Gamma(\mu, u_\mu^k(\cdot)) - \gamma_{m-1}(\mu, k, \cdot)\|_{\ell^\infty(\Omega^{\rm tr})}$

  Set $r_m(\cdot) := \Gamma(\mu_m, u_{\mu_m}^{k_m}(\cdot)) - \gamma_{m-1}(\mu_m, k_m, \cdot)$ and $x_m \in \underset{x\in\Omega^{\rm tr}}{\rm argmax}\, |r_m(x)|$

end while

Set $M := m - 1$

Output : $(x_i)_{1\le i\le M}$ and $(q_j)_{1\le j\le M}$

set $\mathcal{P}^{\rm tr} \subset \mathcal{P}$ for the parameter values; in what follows, we denote by P the cardinality of $\mathcal{P}^{\rm tr}$. For a real-valued function v defined on $\Omega^{\rm tr}$, we define $\|v\|_{\ell^\infty(\Omega^{\rm tr})} := \max_{x\in\Omega^{\rm tr}}|v(x)|$. Given an iteration counter $m \ge 1$ and a function γ_{m-1} defined on $\mathcal{P}^{\rm tr} \times \overline{\mathbb{K}}^{\rm tr} \times \Omega$, with the convention that $\gamma_0 \equiv 0$, an EIM iteration consists of the following steps. First, one defines $(\mu_m, k_m) \in \mathcal{P}^{\rm tr} \times \overline{\mathbb{K}}^{\rm tr}$ by

$$(\mu_m, k_m) \in \underset{(\mu,k)\in\mathcal{P}^{\rm tr}\times\overline{\mathbb{K}}^{\rm tr}}{\rm argmax} \|\Gamma(\mu, u_\mu^k(\cdot)) - \gamma_{m-1}(\mu, k, \cdot)\|_{\ell^\infty(\Omega^{\rm tr})}, \tag{17}$$

where we notice the use of the HF trajectories for all values of the parameter μ in the training set $\mathcal{P}^{\rm tr}$. Once (μ_m, k_m) has been determined, one sets

$$r_m(\cdot) := \Gamma(\mu_m, u_{\mu_m}^{k_m}(\cdot)) - \gamma_{m-1}(\mu_m, k_m, \cdot), \qquad x_m \in \underset{x\in\Omega^{\rm tr}}{\rm argmax}\, |r_m(x)|, \tag{18}$$

and one checks whether $|r_m(x_m)| > \epsilon_{\rm EIM}$ for some user-defined positive threshold $\epsilon_{\rm EIM}$. If this is the case, one sets

$$q_m(\cdot) := \frac{r_m(\cdot)}{r_m(x_m)}, \tag{19}$$

and one computes the new row of the matrix $\mathbf{B}$ by setting $\mathbf{B}_{mi} := (q_i(x_m))$, for all $1 \le i \le m$. The standard EIM procedure is presented in Algorithm 2.

Let us now briefly discuss Task (T_2) above, i.e., the construction of a set of RB functions with cardinality N. First, as usual in RB methods, the solution manifold is explored by considering a training set for the parameter values; for simplicity, we consider the same training set $\mathcal{P}^{\rm tr}$ as for the EIM approximation. This way, one only explores the collection of points $\{u_\mu^k\}_{(\mu,k)\in\mathcal{P}^{\rm tr}\times\overline{\mathbb{K}}^{\rm tr}}$ in the solution manifold. For this exploration to be informative, the training set $\mathcal{P}^{\rm tr}$ has to be chosen large enough. The exploration can be driven by means of an a posteriori error estimator (see, e.g., [19]) which allows one to evaluate only N HF trajectories. However, in the present setting where HF trajectories are to be computed for all the parameters in $\mathcal{P}^{\rm tr}$ when constructing the

EIM approximation, it is natural to exploit these computations by means of a Proper Orthogonal Decomposition (POD) [12, 13] to define the RB. This technique is often considered in the literature to build the RB in a time-dependent setting, see, e.g., [9, 11, 18]. In practice, a POD of the whole collection of snapshots may be computationally demanding (or even unfeasible) when a very large number of functions is considered. Thus, we adopt a POD-based progressive construction of the reduced basis in the spirit of the POD-greedy algorithm from [9]. Therein, one additional RB function is picked at a time, whereas here we can pick more than one function. The progressive construction of the RB is presented in Algorithm 3, where we have chosen an enumeration of the parameters in $\mathcal{P}^{\mathrm{tr}}$ from 1 to P. The initialization of Algorithm 3 is made by computing $(\theta_n)_{1\leq n\leq N^1} = \mathrm{POD}(\mathcal{S}_1, \epsilon_{\mathrm{POD}})$ for the trajectory $\mathcal{S}_1$ associated with the parameter μ_1. That is, we select the first N^1 POD modes out of the set $\mathcal{S}_1$ with error threshold ϵ_{POD} (for completeness, this procedure is briefly outlined in Appendix A). The next steps of the algorithm are performed in an iterative fashion. For each new trajectory, we first subtract its projection on the current RB, and then perform a POD on the projection and merge the result with the current RB. This specific part of the procedure, called UPDATE_RB, is presented in Algorithm 4; this part of the procedure is presented separately since it will be re-used later on.

Algorithm 3 Progressive RB

Input : $\mathcal{P}^{\mathrm{tr}}$, $\overline{\mathbb{K}}^{\mathrm{tr}}$, and $\epsilon_{\mathrm{POD}} > 0$

1: Compute $\big(\mathcal{S}_p\big)_{1\leq p\leq P} = \big((u^k_{\mu_p})_{k\in\overline{\mathbb{K}}^{\mathrm{tr}}}\big)_{1\leq p\leq P}$ $\quad$ P `HF trajectories`

2: Compute $(\theta_n)_{1\leq n\leq N^1} = \mathrm{POD}(\mathcal{S}_1, \epsilon_{\mathrm{POD}})$

3: Set $p = 1$

4: **while** $p < P$ **do**

5: $\quad$ Set $p = p + 1$

6: $\quad$ Compute $(\theta_n)_{1\leq n\leq N^p} = \mathrm{UPDATE_RB}\big((\theta_n)_{1\leq n\leq N^{p-1}}, \mathcal{S}_p, \epsilon_{\mathrm{POD}}\big)$

7: **end while**

8: Set $N := N^P$

9: Compute $\hat{\mathbf{u}}^0$, $(\mathbf{f}^k)_{k\in\mathbb{K}^{\mathrm{tr}}}$, $\mathbf{M}$, and $\mathbf{A}_0$

10: Compute the matrices $(\mathbf{C}^j)_{1\leq j\leq M}$

Output : $(\theta_n)_{1\leq n\leq N}$, $\hat{\mathbf{u}}^0$, $(\mathbf{f}^k)_{k\in\mathbb{K}^{\mathrm{tr}}}$, $\mathbf{M}$, $\mathbf{A}_0$, and $(\mathbf{C}^j)_{1\leq j\leq M}$

Remark 3 (Threshold ϵ_{pod}) *For the initialization (line 2 of Algorithm 3), one can use a relative error threshold for ϵ_{POD} (for instance, $\epsilon_{\mathrm{POD}} = 1\%$). Instead, for the iterative loop (line 6 of Algorithm 3), the threshold ϵ_{POD} can be set to the greatest singular value that has been truncated at the initialization step.*

Remark 4 (Order of EIM and RB) *Algorithms 2 and 3 can be performed in whatever order. If Algorithm 3 is performed first, the computation of the matrices $(\mathbf{C}^j)_{1\leq j\leq M}$ is postponed to the end of Algorithm 2. Moreover, the HF trajectories $(u^k_\mu)_{(\mu,k)\in\mathcal{P}^{\mathrm{tr}}\times\overline{\mathbb{K}}^{\mathrm{tr}}}$ appearing in both algorithms are computed only once.*

5 The Progressive RB-EIM method (PREIM)

In this section, we first present the main ideas of the PREIM algorithm. Then we describe one important building block called UPDATE_EIM. Finally, using this building block together with the

Algorithm 4 UPDATE_RB

Input : $\Theta = (\theta_n)_{1\leq n\leq N}$, $\mathcal{S}$, and $\epsilon_{\text{POD}} > 0$
if $\mathcal{S} = \emptyset$ **then**
    Θ remains unchanged
else
    Define $\tilde{\mathcal{S}} := (u - \Pi_{\text{span}(\Theta)} u)_{u\in\mathcal{S}}$
    Set $\Xi := \text{POD}(\tilde{\mathcal{S}}, \epsilon_{\text{POD}})$
    if $\Xi = \emptyset$ **then**
        Θ remains unchanged
    else
        Set $\Theta := \Theta \cup \Xi$
    end if
end if
Output : Θ

procedure UPDATE_RB from Algorithm 4, we present the PREIM algorithm.

5.1 Main ideas

PREIM consists in a progressive construction of the EIM approximation and of the RB. The key idea is that, unlike the standard EIM for which HF trajectories are computed for all the parameter values in the training set $\mathcal{P}^{\text{tr}}$ (Algorithm 2, line 1), PREIM works with an additional training subset $\mathcal{P}_m^{\text{HF}} \subset \mathcal{P}^{\text{tr}}$ that is enriched progressively with the iteration index m of PREIM. The role of $\mathcal{P}_m^{\text{HF}}$ is to collect the parameter values for which a HF trajectory has already been computed. PREIM is designed such that $\text{Card}(\mathcal{P}_m^{\text{HF}}) \leq m$ for all $m \in \{1, \dots, M\}$. This means that when the final rank-M EIM approximation has been computed, at most M HF trajectories have been evaluated, whence the computational gain with respect to the standard offline stage provided $M \ll P$.

At the iteration $m \geq 1$ of PREIM, the trajectories for all $\mu \in \mathcal{P}_m^{\text{HF}}$ are HF trajectories, whereas they are approximated by RB trajectories for all $\mu \in \mathcal{P}^{\text{tr}} \setminus \mathcal{P}_m^{\text{HF}}$. The RB trajectories can be modified at each iteration m of PREIM. This happens whenever a new value of the parameter is selected in the greedy stage of the EIM so that the approximation of the nonlinearity is modified. To reflect this dependency, we add a superscript m to the RB trajectories which are now denoted $(\hat{u}_\mu^{m,k})_{k\in\overline{\mathbb{K}}^{\text{tr}}}$ for all $\mu \in \mathcal{P}^{\text{tr}} \setminus \mathcal{P}_m^{\text{HF}}$. It is convenient to introduce the notation

$$\bar{u}_\mu^{m,k} := \begin{cases} u_\mu^k & \text{if } \mu \in \mathcal{P}_m^{\text{HF}}, \\ \hat{u}_\mu^{m,k} & \text{otherwise,} \end{cases} \tag{20}$$

and the nonlinear function

$$\bar{\gamma}^m(\mu, k, x) := \Gamma(\mu, \bar{u}_\mu^{m,k}(x)). \tag{21}$$

The goal of every PREIM iteration is twofold:

(i) produce a set of RB functions $(\theta_n^m)_{1\leq n\leq N^m}$ (the RB functions and their number depend on m);

(ii) produce a rank-m approximation of the nonlinear function $\bar{\gamma}^m$ defined by (21) in the form

$$\bar{\gamma}^m_{[\mathcal{P}^{\rm HF}_m,\mathcal{X}_m,\mathcal{Q}_m]}(\mu,k,x) := \sum_{j=1}^m (\bar{\varphi}^m)^k_{\mu,j}\bar{q}_j(x). \tag{22}$$

The notation $\bar{\gamma}^m_{[\mathcal{P}^{\rm HF}_m,\mathcal{X}_m,\mathcal{Q}_m]}$ in (22) indicates the data $[\mathcal{P}^{\rm HF}_m,\mathcal{X}_m,\mathcal{Q}_m]$ that is used to build the approximation of the nonlinearity. More precisely, this construction uses the PREIM training set $\mathcal{P}^{\rm HF}_m$, the sequence of interpolation points $\mathcal{X}_m := (\bar{x}_i)_{1\le i\le m}$ in $\Omega^{\rm tr}$ (with $\bar{x}_m$ computed at iteration m), and the sequence of functions $\mathcal{Q}_m := (\bar{q}_j)_{1\le j\le m}$ defined on Ω (with $\bar{q}_m$ computed at iteration m). The progressive construction of these three ingredients is described below. Then, considering the (invertible) lower-triangular matrix $\bar{\mathbf{B}} \in \mathbb{R}^{m\times m}$ whose last row is calculated using $\bar{\mathbf{B}}_{mj} = \bar{q}_j(\bar{x}_m)$ for all $j \in \{1,\dots,m\}$, we compute the real numbers $(\bar{\varphi}^m)^k_{\mu,j}$ in (22) from the relations

$$\sum_{j=1}^m \bar{\mathbf{B}}_{ij}(\bar{\varphi}^m)^k_{\mu,j} = \bar{\gamma}^m(\mu,k,\bar{x}_i), \qquad \forall i \in \{1,\dots,m\}, \tag{23}$$

for all $(\mu,k) \in \mathcal{P}\times\overline{\mathbb{K}}^{\rm tr}$. All the real numbers $(\bar{\varphi}^m)^k_{\mu,j}$ depend on m since the right-hand side of (23) depends on m.

5.2 The procedure UPDATE_EIM

An essential building block of PREIM is the procedure UPDATE_EIM described in Algorithm 5. The input is the RB functions $(\theta_n)_{1\le n\le N^{m-1}}$, the triple $[\mathcal{P}^{\rm HF}_{\rm in},\mathcal{X}_{m-1},\mathcal{Q}_{m-1}]$ describing the current approximation of the nonlinearity (the choice for the indices will be made clearer in the next section, and is not important at this stage), and the threshold $\epsilon_{\rm EIM}$. The output is the flag incr_rk which indicates whether or not the rank of the EIM approximation has been increased, and if incr_rk = TRUE, the additional output is the triple $[\mathcal{P}^{\rm HF}_{\rm out},\mathcal{X}_{\rm out},\mathcal{Q}_{\rm out}]$ to devise the new EIM approximation, possibly a new HF trajectory $\mathcal{S}_{\rm out}$, and a measure $\delta^{\rm EIM}_m$ on the EIM error.

First (see line 2), one selects a new pair $(\mu_m,k_m) \in \mathcal{P}^{\rm tr}\times\overline{\mathbb{K}}^{\rm tr}$ in a greedy fashion as follows:

$$(\mu_m,k_m) \in \operatorname*{argmax}_{(\mu',k')\in\mathcal{P}^{\rm tr}\times\overline{\mathbb{K}}^{\rm tr}} \|\Gamma\big(\mu',\bar{u}^{k'}_{\mu'}(\cdot)\big) - \bar{\gamma}^{m-1}_{[\mathcal{P}^{\rm HF}_{\rm in},\mathcal{X}_{m-1},\mathcal{Q}_{m-1}]}(\mu',k',\cdot)\|_{\ell^\infty(\Omega^{\rm tr})}. \tag{24}$$

In (24), $\bar{u}^{k'}_{\mu'}$ is defined as in (20) using the set $\mathcal{P}^{\rm HF}_{\rm in}$. Therefore, the selection criterion (24) exploits the knowledge of the HF trajectory for all the parameter values in $\mathcal{P}^{\rm HF}_{\rm in}$, and otherwise uses a RB trajectory. This is an important difference with respect to the standard offline stage. There are now two possibilities: (i) either μ_m is already in $\mathcal{P}^{\rm HF}_{\rm in}$; then, no new HF trajectory is computed and we set $\mathcal{P}^{\rm HF}_{\rm out} := \mathcal{P}^{\rm HF}_{\rm in}$ (line 8); (ii) or μ_m is not in $\mathcal{P}^{\rm HF}_{\rm in}$; then we compute a new HF trajectory for the parameter μ_m and we set $\mathcal{P}^{\rm HF}_{\rm out} := \mathcal{P}^{\rm HF}_{\rm in} \cup \{\mu_m\}$ (line 5). Our numerical experiments reported in Section 6 below will show that at many iterations of PREIM, the pair (μ_m,k_m) selected in (24) differs from the previously selected pair by the time index and not by the parameter value; this means that for many PREIM iterations, no additional HF computation is performed. In case of non-uniqueness of the maximizer in (24), one selects, if possible, a trajectory for which the parameter is not already in the set $\mathcal{P}^{\rm HF}_{\rm in}$ in order to trigger a computation of a new HF trajectory.

An additional feature of PREIM is that, whenever a new HF trajectory is actually computed, one can either confirm or update the selected pair (μ_m,k_m) using the following HF-based re-selection

Algorithm 5 UPDATE_EIM

Input : $(\theta_n)_{1\le n\le N^{m-1}}$, $\mathcal{P}_{\text{in}}^{\text{HF}}$, $\mathcal{X}_{m-1}$, $\mathcal{Q}_{m-1}$, and ϵ_{EIM}

1: Compute $(\bar{u}_\mu^k)_{(\mu,k)\in\mathcal{P}^{\text{tr}}\times\overline{\mathbb{K}}^{\text{tr}}}$ using $(\theta_n)_{1\le n\le N^{m-1}}$

2: Search $(\mu_m, k_m) \in \underset{(\mu',k')\in\mathcal{P}^{\text{tr}}\times\overline{\mathbb{K}}^{\text{tr}}}{\text{argmax}} \|\Gamma\big(\mu', \bar{u}_{\mu'}^{k'}(\cdot)\big) - \bar{\gamma}^{m-1}_{[\mathcal{P}_{\text{in}}^{\text{HF}},\mathcal{X}_{m-1},\mathcal{Q}_{m-1}]}(\mu', k', \cdot)\|_{\ell^\infty(\Omega^{\text{tr}})}$ `based on RB/HF`

3: Define $\widetilde{r}_m(\cdot) = \Gamma\big(\mu_m, \bar{u}_{\mu_m}^{k_m}(\cdot)\big) - \bar{\gamma}^{m-1}_{[\mathcal{P}_{\text{in}}^{\text{HF}},\mathcal{X}_{m-1},\mathcal{Q}_{m-1}]}(\mu_m, k_m, \cdot)$.

4: **if** $\mu_m \notin \mathcal{P}_{\text{in}}^{\text{HF}}$ **then**

5: Compute $\mathcal{S}_{\text{out}} = (u_{\mu_m}^k)_{k\in\overline{\mathbb{K}}^{\text{tr}}}$ and set $\mathcal{P}_{\text{out}}^{\text{HF}} = \mathcal{P}_{\text{in}}^{\text{HF}} \cup \{\mu_m\}$ `one HF trajectory`

6: Search $(\bar{\mu}_m, \bar{k}_m) \in \underset{(\mu',k')\in\mathcal{P}_{\text{out}}^{\text{HF}}\times\overline{\mathbb{K}}^{\text{tr}}}{\text{argmax}} \|\Gamma\big(\mu', u_{\mu'}^{k'}(\cdot)\big) - \bar{\gamma}^{m-1}_{[\mathcal{P}_{\text{in}}^{\text{HF}},\mathcal{X}_{m-1},\mathcal{Q}_{m-1}]}(\mu', k', \cdot)\|_{\ell^\infty(\Omega^{\text{tr}})}$

7: **else**

8: Set $\mathcal{S}_{\text{out}} = \emptyset$, $\mathcal{P}_{\text{out}}^{\text{HF}} = \mathcal{P}_{\text{in}}^{\text{HF}}$, and $(\bar{\mu}_m, \bar{k}_m) = (\mu_m, k_m)$

9: **end if**

10: Define $\bar{r}_m(\cdot) := \Gamma\big(\bar{\mu}_m, u_{\bar{\mu}_m}^{\bar{k}_m}(\cdot)\big) - \bar{\gamma}^{m-1}_{[\mathcal{P}_{\text{in}}^{\text{HF}},\mathcal{X}_{m-1},\mathcal{Q}_{m-1}]}(\bar{\mu}_m, \bar{k}_m, \cdot)$

11: **if** $\|\bar{r}_m\|_{\ell^\infty(\Omega^{\text{tr}})} < \epsilon_{\text{EIM}}$ **then**

12: Set incr_rk = FALSE

13: Define $r_m(\cdot) = \widetilde{r}_m(\cdot)$ `discard the EIM selection`

14: Set $\mathcal{X}_{\text{out}} = \mathcal{X}_{m-1}$ and $\mathcal{Q}_{\text{out}} = \mathcal{Q}_{m-1}$

15: **else**

16: Set incr_rk = TRUE

17: Define $r_m(\cdot) = \bar{r}_m(\cdot)$

18: Set $\mathcal{X}_{\text{out}} = (\mathcal{X}_{m-1}, \bar{x}_m)$ and $\mathcal{Q}_{\text{out}} = (\mathcal{Q}_{m-1}, \bar{q}_m)$ with $\bar{x}_m$, $\bar{q}_m$ as in Algorithm 2 (lines 6 and 9).

19: **end if**

20: Define $\delta_m^{\text{EIM}} = \|r_m\|_{\ell^\infty(\Omega^{\text{tr}})}$

Output : incr_rk, $\mathcal{P}_{\text{out}}^{\text{HF}}$, $\mathcal{X}_{\text{out}}$, $\mathcal{Q}_{\text{out}}$, $\mathcal{S}_{\text{out}}$, and δ_m^{EIM}

criterion (see line 6):

$$(\bar{\mu}_m, \bar{k}_m) \in \underset{(\mu',k')\in\mathcal{P}_{\text{out}}^{\text{HF}}\times\overline{\mathbb{K}}^{\text{tr}}}{\text{argmax}} \|\Gamma\big(\mu', u_{\mu'}^{k'}(\cdot)\big) - \bar{\gamma}^{m-1}_{[\mathcal{P}_{\text{in}}^{\text{HF}},\mathcal{X}_{m-1},\mathcal{Q}_{m-1}]}(\mu', k', \cdot)\|_{\ell^\infty(\Omega^{\text{tr}})}. \tag{25}$$

We notice that this re-selection criterion only handles HF trajectories since the parameter values are in $\mathcal{P}_{\text{out}}^{\text{HF}}$. Moreover, (25) only requires to probe the values for μ_m, since the values for the other parameters, which are in $\mathcal{P}_{\text{in}}^{\text{HF}}$, have already been evaluated in (24). Finally, to prevent division by small quantities, the value of the residual $\|\bar{r}_m\|_{\ell^\infty(\Omega^{\text{tr}})}$ is checked in line 11. If this value is too small, the pair $(\bar{\mu}_m, \bar{k}_m)$ is rejected and the rank of the EIM approximation is not increased.

5.3 The PREIM algorithm

We are now ready to describe the PREIM procedure, see Algorithm 6. PREIM is an iterative method that builds progressively the RB and the EIM approximation. The iteration is controlled by three tolerances: $\epsilon_{\text{POD}} > 0$ which is used in the progressive increment of the RB, $\epsilon_{\text{EIM}} > 0$ which is used to check the quality of the EIM approximation, and $\epsilon_{\text{RB}} > 0$ which is used to check the quality of the RB. The termination criterion involves the quality of both the EIM and the RB approximations, see line 7. Note that this is the same criterion as in the standard RB-EIM approach.

Within each PREIM iteration, the two previously-described procedures UPDATE_EIM and UPDATE_RB are called. First, one attempts to improve the EIM approximation (line 9). If this is successful (i.e., if incr_rk = TRUE), the RB is updated by using the possibly new HF trajectory $\mathcal{S}_m$ (line 21). Otherwise (i.e., if incr_rk = FALSE), the RB is possibly updated (line 12) and a new improvement of the EIM is attempted (line 13). In general, the RB is improved because a new HF trajectory has been computed. Whenever this is not the case, a new HF trajectory is anyway computed in line 18 (cf. Remark 5 below). The choice of this new HF trajectory can be driven by a standard greedy algorithm based on the use of a classical a posteriori error estimator. More precisely, for a given reduced basis $(\theta_n)_{1\leq n\leq N}$ and given sets of training points $\mathcal{X}$ and functions $\mathcal{Q}$ used for the current EIM approximation of the nonlinearity, the associated a posteriori error estimator for a given value of the parameter $\mu \in \mathcal{P}$ is denoted by $\Delta^{\mathcal{X},\mathcal{Q}}_{(\theta_n)_{1\leq n\leq N}}(\mu)$. Finally, we observe that the reduced matrices and vectors in line 22 of Algorithm 6 need to be updated since these quantities depend on the RB functions which can change at every iteration.

Remark 5 (Worst-case scenario) *The worst-case scenario is that in which PREIM would compute as many trajectories as the standard EIM. In this situation, the RB space would be identical to that of the standard RB-EIM. Regarding the approximation of the nonlinearity, if PREIM is carried on until $M = M_{\max} := P \times K$, the resulting rank-$M$ approximation would be exact for all the parameters in $\mathcal{P}^{\rm tr}$. Hence, as $\epsilon_{\rm RB}$, $\epsilon_{\rm POD}$, and $\epsilon_{\rm EIM}$ tend to zero, RB-EIM and PREIM produce the same approximations at termination (recall that termination is guaranteed for both algorithms).*

Let us now discuss the initialization of PREIM. In line 2, one can choose an initial PREIM training set $\mathcal{P}_1^{\rm HF}$ composed of a single parameter, as is often the case with greedy algorithms. Although the nonlinearity may not be well-described initially, one can expect that the description will improve progressively. Still, to allow for more robustness in the initialization, one can consider an initial set $\mathcal{P}_1^{\rm HF}$ composed of several parameters. One can then compute the HF trajectories for all $\mu \in \mathcal{P}_1^{\rm HF}$ and compress them using the POD procedure with threshold $\epsilon_{\rm POD}$ (if $\mathcal{P}_1^{\rm HF}$ contains more than one value, a progressive version is used). Finally, one selects

$$(\mu_1, k_1) \in \underset{(\mu',k')\in\mathcal{P}_1^{\rm HF}\times\overline{\mathbb{K}}^{\rm tr}}{\operatorname{argmax}} \|\Gamma\big(\mu', u^{k'}_{\mu'}(\cdot)\big)\|_{\ell^\infty(\Omega^{\rm tr})}, \tag{26}$$

one defines $r_1(\cdot) = \Gamma(\mu_1, u^{k_1}_{\mu_1}(\cdot))$ and computes $\mathcal{X}_1 = (\bar{x}_1)$, $\mathcal{Q}_1 = (\bar{q}_1)$ (as in the standard EIM procedure), and one sets $\delta_1^{\rm EIM} = \|r_1\|_{\ell^\infty(\Omega^{\rm tr})}$. Let us finally point out that a good initialization of PREIM can favor its early termination. For instance, one can try to select the first parameter as one for which the nonlinearity has a sizeable effect.

Remark 6 (PREIM-NR and U-SER variants) *We can consider two variants in the procedure UPDATE_EIM (Algorithm 5) and therefore in PREIM. A first variant consists in skipping the re-selection step in line 6 of Algorithm 5. This variant, which we call PREIM-NR (for 'no re-selection'), will be tested numerically in the next section in order to highlight the actual benefits brought by the re-selection. A second variant is to replace $\bar{u}^{m,k}_\mu$ with $\hat{u}^{m,k}_\mu$ in lines 1 and 2 of Algorithm 5, and to skip the re-selection step in line 6. We call this variant U-SER since it can be viewed as an extension of SER [2] to the unsteady setting. The crucial difference between PREIM-NR and U-SER is that U-SER uses RB trajectories to compute the space-dependent functions in the EIM approximation, whereas PREIM-NR uses HF trajectories.*

6 Numerical results

In this section, we illustrate the above developments on three test cases related to transient heat transfer problems. The first two test cases use the idealized 2D geometry of a perforated square plate; the first test case involves a nonlinearity on the solution, whereas the second test case considers a nonlinearity on its partial derivatives. The third test case is based on the three-dimensional geometry of an industrial valve prototype used for flow regulation in nuclear reactor operation, while we use the same type of nonlinearity as in the first test case. Our goal is to illustrate the computational performance of PREIM and to compare it to the standard EIM approach described in Section 4 and to the variants PREIM-NR and U-SER described in Remark 6. HF trajectories are computed using a Finite Element subspace $X \subset Y = H^1(\Omega)$ consisting of continuous, piecewise affine functions. The HF computations use the industrial software `code_aster` [4] for the first test case, `FreeFem++` [10] for the second test case, and a combination of the industrial software `Salomé` and `FreeFem++` for the third test case. The reduced-order modeling algorithms have been developed in Python. In all the test cases, the dominant error component turns out to be the one resulting from the approximation of the nonlinearity, rather than the one resulting from the RB. For this reason, PREIM has been run using only the stopping criterion $\delta_m^{\mathrm{EIM}} > \epsilon_{\mathrm{EIM}}$ in line 7 of Algorithm 6.

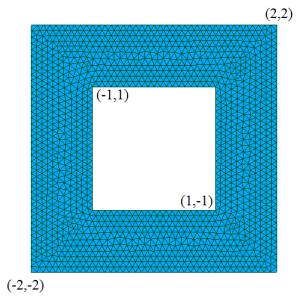

Figure 1: Test cases (a) and (b): The computational domain is a perforated plate.

6.1 Test case (a): Nonlinearity on the solution

We consider a two-dimensional setting based on the perforated plate illustrated in Figure 1 with $\Omega = (-2,2)^2 \backslash [-1,1]^2 \subset \mathbb{R}^2$. We consider the nonlinear parabolic problem (1) with the nonlinear function $\Gamma(\mu, v) := \sin\left(\frac{2\pi\mu}{20}\left(\frac{v-u_0}{u_m-u_0}\right)^2\right)$, with $u_0 = 293$ K (20 ^{o}C) and $u_m = 323$ K (50 ^{o}C). We define $\kappa_0 = 1.05$ m^2·K^{-2}·s^{-1} and $\phi_e = 3$ K·m·s^{-1} (these units result from our normalization by the density times the heat capacity). For space discretization, we use a mesh containing $\mathcal{N} = 1438$ nodes (see Figure 1). Regarding time discretization, we consider the time interval $I = [0, 5]$, the set of discrete times nodes $\mathbb{K}^{\mathrm{tr}} = \{1, \cdots, 50\}$, and a constant time step $\Delta t^k = 0.1$ s for all $k \in \mathbb{K}^{\mathrm{tr}}$. Finally, we consider the parameter interval $\mathcal{P} = [1, 20]$, the training set $\mathcal{P}^{\mathrm{tr}} = \{1, \cdots, 20\}$, and we use the larger set $\{0.25i \mid 0 \leq i \leq 80\}$ to verify our numerical results. In Figure 2, we show the HF temperature profiles over the perforated plate at two different times and for two different parameter values. We can see that, as the simulation time increases, the temperature is, overall, higher for larger values of the parameter μ than for smaller values. Also, for larger values of μ, the temperature variation tends to be less uniform over the plate than for smaller values of μ.

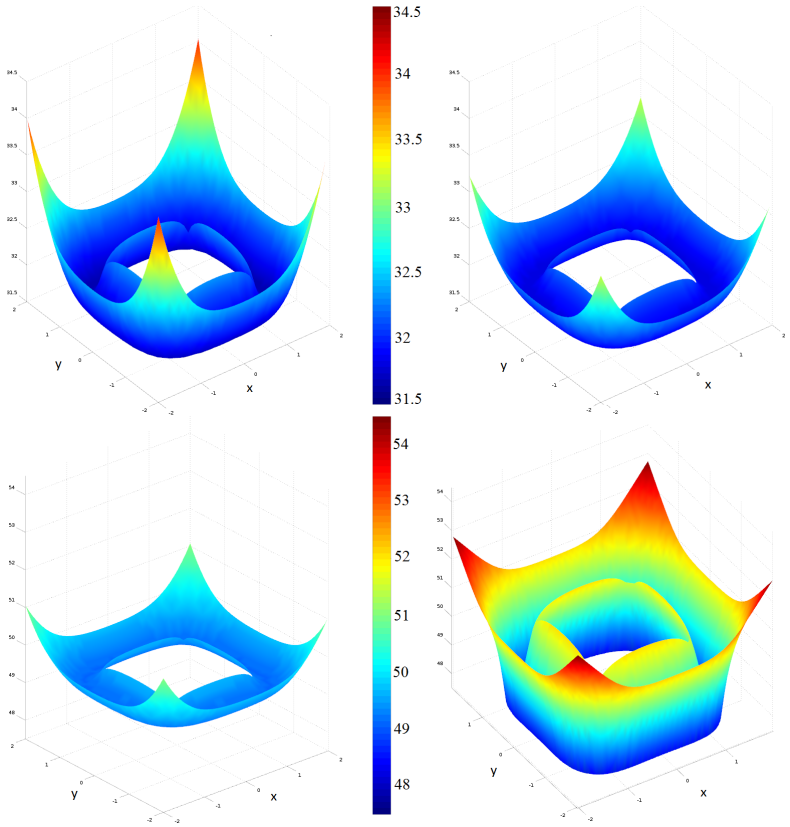

Figure 2: Test case (a): HF solutions for the parameter values $\mu = 1$ (left) and $\mu = 18$ (right) at $t = 2$ s (top) and $t = 5$ s (bottom).

m	1	2	6	14	15	20	25
$\|r_m\|_{\ell^\infty(\Omega^{\mathrm{tr}})}$	2.0	8.1E−1	1.1E−1	5.2E−3	2.6E−3	1.1E−3	1.6E−4

Table 1: Test case (a): Evolution of the standard EIM error. m is the rank of the EIM approximation.

During the standard offline stage, we perform $P = 20$ HF computations. Knowing that $K = 50$, the set $\mathcal{S}$ (Algorithm 2, line 1) contains 1020 fields, each consisting of $\mathcal{N} = 1438$ nodal values. Applying the POD in a progressive manner (see Algorithm 3 with the parameters enumerated using increasing values) based on the H^1-norm and a truncation threshold $\epsilon_{\text{POD}} = 10^{-3}$, we obtain $N = 6$ RB functions. Afterwards, we perform the standard EIM algorithm whose convergence is reported in Table 1 for selected values of the rank of the EIM approximation. For $\epsilon_{\text{EIM}} = 5 \cdot 10^{-2}$, the final rank of the EIM approximation is $M = 8$, whereas for $\epsilon_{\text{EIM}} = 5 \cdot 10^{-3}$, the final rank of the EIM approximation is $M = 15$.

We now investigate PREIM, which we first run with thresholds $\epsilon_{\text{POD}} = 10^{-3}$ and $\epsilon_{\text{EIM}} = 5 \cdot 10^{-2}$. Table 2 shows the selected parameters and discrete time nodes at each stage of PREIM. We can make two important observations from this table. First, after 13 iterations, PREIM has only selected four different parameter values, and has therefore computed only four HF trajectories (the iterations for which a new parameter value is selected are indicated in gray in Table 2). In the other 9 out of the 13 iterations, a different time snapshot of an already existing HF trajectory has been selected. Second, by comparing the lines in Table 2 related to μ and $\bar{\mu}$, we can see that a parameter re-selection

m		1	2	3	4	5	6	7	8	9	10	11	12	13
PREIM	$\bar{\mu}$	1	20	20	20	20	20	20	20	16	20	20	18	20
	μ	1	20	20	20	20	18	20	20	16	20	20	18	20
	k	50	45	48	50	43	42	39	46	50	49	33	50	47

Table 2: Test case (a): Selected parameters and time nodes in PREIM. The gray cells correspond to a new parameter selection and, therefore, to a new HF computation.

happened at iteration $m = 7$.

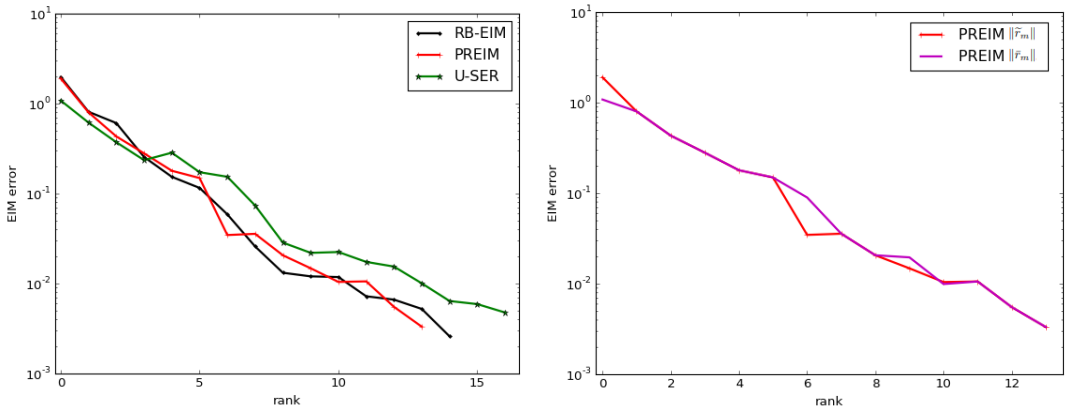

Figure 3: Test case (a): EIM approximation error as a function of m for $\epsilon_{\text{POD}} = 10^{-3}$ and $\epsilon_{\text{EIM}} = 5 \cdot 10^{-2}$. Left: Errors for the standard RB-EIM procedure, PREIM, and U-SER. Right: Errors $\|\widetilde{r}_m\|_{\ell^\infty(\Omega^{\text{tr}})}$ and $\|\bar{r}_m\|_{\ell^\infty(\Omega^{\text{tr}})}$ for PREIM.

The left panel of Figure 3 displays the error on the approximation of the nonlinear function Γ for the standard RB-EIM procedure and for PREIM as a function of the iteration number m (the additional curve concerning U-SER will be commented afterwards), i.e., we plot $\|\bar{r}_m\|_{\ell^\infty(\Omega^{\text{tr}})}$ (line 3 of Algorithm 5) and $\|\widetilde{r}_m\|_{\ell^\infty(\Omega^{\text{tr}})}$ (line 10 of Algorithm 5) as a function of m, see (22). We can see that the quality of the approximation of the nonlinearity is almost the same for PREIM as for the standard RB-EIM procedure; yet, the former achieves this accuracy by computing 20% of the HF trajectories computed by the latter (4 instead of 20 HF trajectories). The right panel of Figure 3 shows the values of $\|\widetilde{r}_m\|_{\ell^\infty(\Omega^{\text{tr}})}$ and $\|\bar{r}_m\|_{\ell^\infty(\Omega^{\text{tr}})}$ as a function of m. The two quantities differ when the parameter μ_m in line 2 of Algorithm 5 is not in the set $\mathcal{P}^{\text{HF}}_{m-1}$ so that $\|\widetilde{r}_m\|_{\ell^\infty(\Omega^{\text{tr}})}$ is computed using a RB approximation, whereas $\|\bar{r}_m\|_{\ell^\infty(\Omega^{\text{tr}})}$ results from a HF trajectory. Discarding the initialization, this happens for $m \in \{6, 9, 10\}$. The fact that $\|\widetilde{r}_m\|_{\ell^\infty(\Omega^{\text{tr}})}$ and $\|\bar{r}_m\|_{\ell^\infty(\Omega^{\text{tr}})}$ take rather close values indicates that the RB provides an accurate approximation of the HF trajectory.

The left panel of Figure 4 compares the space-time errors (measured using the ℓ^2-norm in time and the H^1-norm in space) on the trajectories produced by the standard RB-EIM and the PREIM procedures for the whole parameter range. The error is generically denoted $\|u_\mu - \hat{u}_\mu\|_{\ell^2(I^{\text{tr}};H^1(\Omega^{\text{tr}}))}$. We observe an excellent agreement over the whole parameter range. In the right panel of Figure 4, we also consider the space-time errors on the trajectories produced using the approximation of the nonlinearity resulting from PREIM with the RB resulting from the standard algorithm. We do not observe any significant change with respect to the left panel, which indicates that the dominant error component is that associated with the approximation of the nonlinearity. We consider the

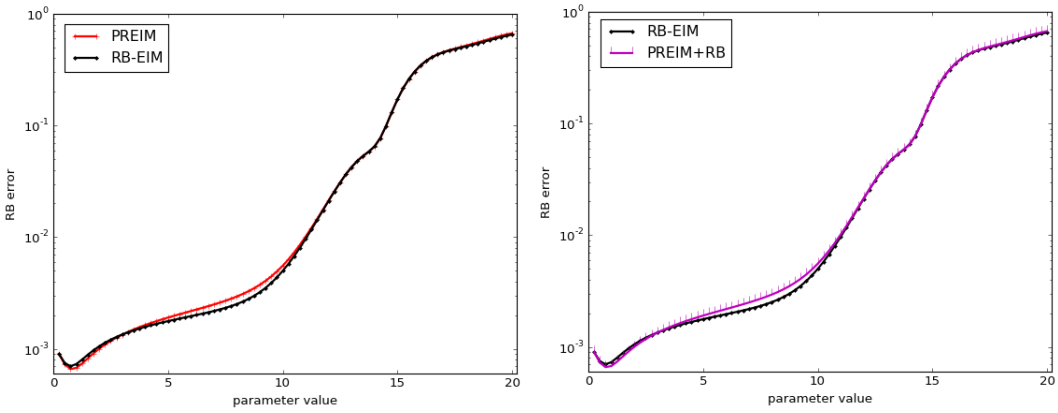

Figure 4: Test case (a): RB approximation error $\|u_\mu - \hat{u}_\mu\|_{\ell^2(I^{\mathrm{tr}};H^1(\Omega^{\mathrm{tr}}))}$ for $\epsilon_{\mathrm{POD}} = 10^{-3}$ and $\epsilon_{\mathrm{EIM}} = 5 \cdot 10^{-2}$.

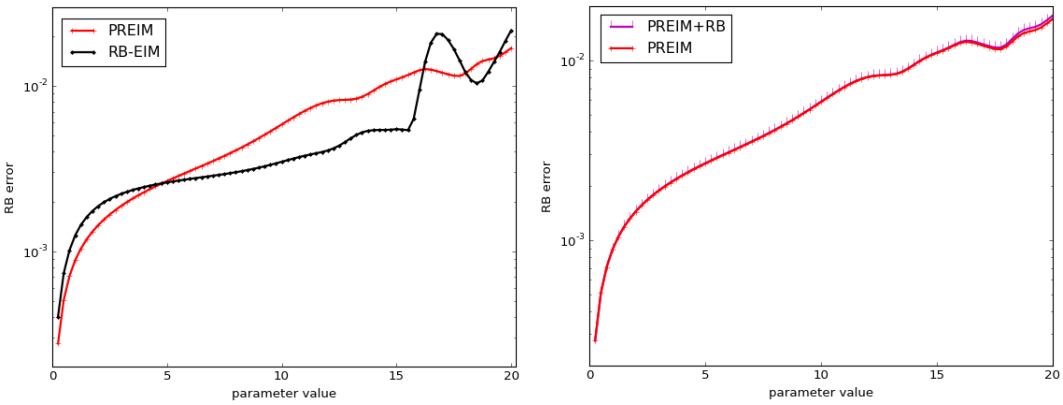

Figure 5: Test case (a): RB approximation error $\|u_\mu - \hat{u}_\mu\|_{\ell^2(I^{\mathrm{tr}};H^1(\Omega^{\mathrm{tr}}))}$ for $\epsilon_{\mathrm{POD}} = 10^{-5}$ and $\epsilon_{\mathrm{EIM}} = 5 \cdot 10^{-3}$.

tighter couple of thresholds $\epsilon_{\mathrm{POD}} = 10^{-5}$ and $\epsilon_{\mathrm{EIM}} = 5 \cdot 10^{-3}$ in Figure 5. Here, we can observe some differences in the errors produced by the standard RB-EIM and PREIM procedures, although both errors remain comparable and reach similar maximum values over the parameter range. While the standard procedure is slightly more accurate for most parameter values, the conclusion is reversed for some other values. Moreover, the curves on the right panel of Figure 5 corroborate the fact that once again, the dominant error component is that associated with the approximation of the nonlinearity.

Let us further explore the PREIM algorithm by comparing it to its variants U-SER and PREIM-NR introduced in Remark 6. Table 3 reports the selected parameters and time nodes in U-SER and PREIM-NR (compare with Table 2 for PREIM). Both U-SER and PREIM-NR need to compute five HF trajectories, which is only 25% of those needed with the standard RB-EIM procedure, but this is still one more HF trajectory than with PREIM. One difference between U-SER and PREIM-NR is that new parameters are selected earlier with U-SER. Interestingly, after 13 iterations, U-SER and PREIM-NR have selected the same five parameters. Another interesting observation is that U-SER actually selects the same couple (μ, k) twice (this happens for $m = 2$ and $m = 6$); this can be interpreted by observing that owing to the improvement of the RB using HF trajectories between iterations $m = 2$ and $m = 6$, the algorithm detects the need to improve the approximation of the nonlinearity by using the same pair (μ, k). The same observation can be made for PREIM-NR (this

m		1	2	3	4	5	6	7	8	9	10	11	12	13
U-SER	μ	1	20	20	20	16	20	19	20	20	19	17	20	19
	k	50	49	50	46	42	49	44	39	50	49	48	47	50
PREIM-NR	μ	1	20	20	20	20	16	20	20	20	20	20	17	19
	k	50	47	50	46	42	49	48	46	39	50	45	50	50

Table 3: Test case (a): Selected parameters and time nodes in U-SER and PREIM-NR. The gray cells correspond to a new parameter selection and, therefore, to a new HF computation.

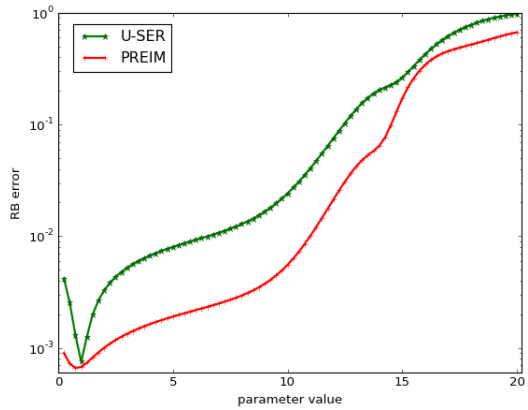

Figure 6: Test case (a): RB approximation error $\|u_\mu - \hat{u}_\mu\|_{\ell^2(I^{\mathrm{tr}};H^1(\Omega^{\mathrm{tr}}))}$ for $\epsilon_{\mathrm{POD}} = 10^{-3}$ and $\epsilon_{\mathrm{EIM}} = 5 \cdot 10^{-2}$.

happens for $m = 4$ and $m = 8$). We emphasize that re-selecting the same pair (μ, k) is not possible within PREIM since the selection is based on HF trajectories. The left panel of Figure 3 displays the error on the approximation of the nonlinear function Γ obtained with U-SER and compares it to the error obtained with the standard RB-EIM and PREIM procedures that were already discussed. The U-SER error is evaluated as $\sup_{(\mu,k)\in\mathcal{P}^{\mathrm{tr}}\times\overline{\mathbb{K}}^{\mathrm{tr}}} \|\Gamma(\mu, \hat{u}^k_\mu(\cdot)) - \overline{\gamma}^m_{[\mathcal{P}^{\mathrm{HF}}_m,\mathcal{X}_m,\mathcal{Q}_m]}(\mu, k, \cdot)\|_{\ell^\infty(\Omega^{\mathrm{tr}})}$. We observe that the approximation of the nonlinearity is somewhat less sharp with U-SER than with PREIM. Figure 6 reports the space-time errors (measured using the ℓ^2-norm in time and the H^1-norm in space) on the trajectories produced by PREIM and U-SER for the whole parameter range. We observe that the U-SER error is always larger, sometimes up to a factor of five, but for the larger parameter values which produce the larger errors, the quality of the results produced by PREIM and U-SER remains comparable.

	RB-EIM	PREIM	U-SER
HF computations	99%	20.0%	25.0%
greedy runtime	1%	1.5%	2.3%
Total runtime	100%	21.5%	26.3%

Table 4: Test case (a): Runtime measurements.

Finally, we provide an assessment of the runtimes in Table 4. We can see that for the standard RB-EIM procedure, the computation of the HF trajectories dominates the cost of the offline phase. For both PREIM and U-SER, the cost of these HF computations is substantially reduced. At the

same time, the cost of the greedy algorithm (which includes the construction of the EIM and of the RB) is increased by 50% with respect to the standard RB-EIM procedure. However, the impact on the total runtime is marginal.

Algorithm 6 Progressive RB-EIM (PREIM)

Input : $\mathcal{P}^{\mathrm{tr}}$, $\overline{\mathbb{K}}^{\mathrm{tr}}$, Ω^{tr}, $\epsilon_{\mathrm{POD}} > 0$, $\epsilon_{\mathrm{EIM}} > 0$, and $\epsilon_{\mathrm{RB}} > 0$

1: Set $m = 1$

2: Choose $\mathcal{P}_1^{\mathrm{HF}} \subsetneq \mathcal{P}^{\mathrm{tr}}$ of cardinality J and compute $\mathcal{S}_1 = (u_\mu^k)_{(\mu,k)\in\mathcal{P}_1^{\mathrm{HF}}\times\overline{\mathbb{K}}^{\mathrm{tr}}}$ $J \geq 1$ `HF trajectories`

3: Compute $(\theta_n^1)_{1\leq n\leq N^1} = \mathrm{POD}(\mathcal{S}_1, \epsilon_{\mathrm{POD}})$.

4: Compute $\hat{\mathbf{u}}^0 \in \mathbb{R}^{N^1}$, $(\mathbf{f}^k)_{k\in\mathbb{K}^{\mathrm{tr}}} \in (\mathbb{R}^{N^1})^K$, $\mathbf{M} \in \mathbb{R}^{N^1\times N^1}$, and $\mathbf{A}_0 \in \mathbb{R}^{N^1\times N^1}$

5: Compute $(\mathcal{X}_1, \mathcal{Q}_1, \delta_1^{\mathrm{EIM}}) = \mathrm{INIT_EIM}(\mathcal{P}_1^{\mathrm{HF}})$ and $\mathbf{C}^1 \in \mathbb{R}^{N^1\times N^1}$

6: Compute $\delta_1^{\mathrm{RB}} = \max_{\mu\in\mathcal{P}^{\mathrm{tr}}} \Delta_1(\mu)$

7: **while** ($\delta_m^{\mathrm{EIM}} > \epsilon_{\mathrm{EIM}}$ or $\delta_m^{\mathrm{RB}} > \epsilon_{\mathrm{RB}}$) **do**

8: Set $m = m + 1$ and $\mathcal{P}_{\mathrm{in}}^{\mathrm{HF}} := \mathcal{P}_{m-1}^{\mathrm{HF}}$

9: (incr_rk, $\mathcal{P}_{\mathrm{out}}^{\mathrm{HF}}$, $\mathcal{X}_{\mathrm{out}}$, $\mathcal{Q}_{\mathrm{out}}$, $\mathcal{S}_{\mathrm{out}}$ δ_m^{EIM}) = UPDATE_EIM $((\theta_n^{m-1})_{1\leq n\leq N^{m-1}}$, $\mathcal{P}_{\mathrm{in}}^{\mathrm{HF}}$, $\mathcal{X}_{m-1}$, $\mathcal{Q}_{m-1}$, $\epsilon_{\mathrm{EIM}})$

10: **while** incr_rk = FALSE **do**

11: $\mathcal{P}_{\mathrm{in}}^{\mathrm{HF}} = \mathcal{P}_{\mathrm{out}}^{\mathrm{HF}}$

12: $(\theta_n^{m-1})_{1\leq n\leq N^{m-1}} = \mathrm{UPDATE_RB}\big((\theta_n^{m-1})_{1\leq n\leq N^{m-1}}, \mathcal{S}_{\mathrm{out}}, \epsilon_{\mathrm{POD}}\big)$

13: (incr_rk, $\mathcal{P}_{\mathrm{out}}^{\mathrm{HF}}$, $\mathcal{X}_{\mathrm{out}}$, $\mathcal{Q}_{\mathrm{out}}$, $\mathcal{S}_{\mathrm{out}}$, δ_m^{EIM}) = UPDATE_EIM $((\theta_n^{m-1})_{1\leq n\leq N^{m-1}}$, $\mathcal{P}_{\mathrm{in}}^{\mathrm{HF}}$, $\mathcal{X}_{m-1}$, $\mathcal{Q}_{m-1}$, $\epsilon_{\mathrm{EIM}})$

14: **if** incr_rk = TRUE **then**

15: Step to line 20

16: **end if**

17: Compute $\mu_m \in \underset{\mu\in\mathcal{P}^{\mathrm{tr}}}{\mathrm{argmax}}\ \Delta^{\mathcal{X}_{\mathrm{out}},\mathcal{Q}_{\mathrm{out}}}_{(\theta_n^{m-1})_{1\leq n\leq N^{m-1}}}(\mu)$

18: Compute $\mathcal{S}_{\mathrm{out}} = (u_{\mu_m}^k)_{k\in\overline{\mathbb{K}}^{\mathrm{tr}}}$ `one HF trajectory`

19: **end while**

20: Set $\mathcal{P}_m^{\mathrm{HF}} = \mathcal{P}_{\mathrm{out}}^{\mathrm{HF}}$, $\mathcal{S}_m = \mathcal{S}_{\mathrm{out}}$, $\mathcal{X}_m = \mathcal{X}_{\mathrm{out}}$, and $\mathcal{Q}_m = \mathcal{Q}_{\mathrm{out}}$

21: Compute $(\theta_n^m)_{1\leq n\leq N^m} = \mathrm{UPDATE_RB}\big((\theta_n^{m-1})_{1\leq n\leq N^{m-1}}, \mathcal{S}_m, \epsilon_{\mathrm{POD}}\big)$

22: Update $\hat{\mathbf{u}}^0 \in \mathbb{R}^{N^m}$, $(\mathbf{f}^k)_{k\in\mathbb{K}^{\mathrm{tr}}} \in (\mathbb{R}^{N^m})^K$, and the matrices $\mathbf{M}$, $\mathbf{A}_0$, $(\mathbf{C}^j)_{1\leq j\leq m}$ in $\mathbb{R}^{N^m\times N^m}$

23: Compute $\delta_m^{\mathrm{RB}} = \max\limits_{\mu\in\mathcal{P}^{\mathrm{tr}}} \Delta^{\mathcal{X}_m,\mathcal{Q}_m}_{(\theta_n^m)_{1\leq n\leq N^m}}(\mu)$

24: **end while**

25: Set $M := m$

Output : $(\theta_n)_{1\leq n\leq N^M}$, $\hat{\mathbf{u}}^0$, $(\mathbf{f}^k)_{k\in\mathbb{K}^{\mathrm{tr}}}$, $\mathbf{M}$, $\mathbf{A}_0$, $\mathcal{X}_M$, $\mathcal{Q}_M$, and $(\mathbf{C}^j)_{1\leq j\leq M}$

6.2 Test case (b): Nonlinearity on the partial derivatives

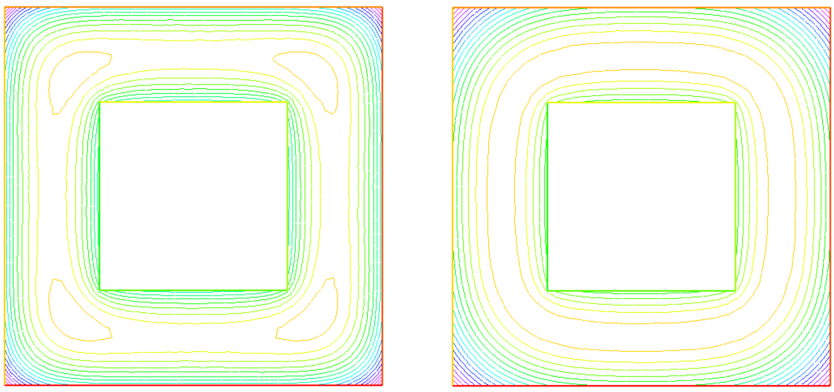

Figure 7: Test case (b): HF solutions for the parameter values $\mu = 1$ at $t = 1$ s (left, values from 20.2 to 22.1) and at $t = 2.5$ s (right, values from 34.5 to 37.3).

We consider the nonlinear parabolic problem (1) with the nonlinear function $\Gamma(\mu, z) := \sin\Big(\omega\mu\big((\frac{\partial u}{\partial x})^2 + (\frac{\partial u}{\partial y})^2\big)\Big)^2$, where $\omega = 6.25 \cdot 10^{-3}$. We define $u_0 = 293$ K (20 ^{o}C), $\kappa_0 = 1$ m^2·K^{-2}·s^{-1} and $\phi_e = 3$ K·m·s^{-1} (these units result from our normalization by the density times the heat capacity). For the space discretization, we use a mesh containing $\mathcal{N} = 1429$ nodes. Regarding time discretization, we consider the time interval $I = [0, 2.5]$, the set of discrete times nodes $\mathbb{K}^{\text{tr}} = \{1, \ldots, 50\}$, and a constant time step $\Delta t^k = 0.05$ s for all $k \in \mathbb{K}^{\text{tr}}$. Finally, we consider the parameter interval $\mathcal{P} = [1, 40]$ and the training set $\mathcal{P}^{\text{tr}} = \{1, \ldots, 40\}$. In Figure 7, we show the temperature isovalues over the perforated plate at two different times for $\mu = 1$. We can observe different boundary layers depending on the time (the same observation can be made by varying the parameter value).

p	1	2	3	8	20	23	24	26	32	33	36	37	39	40
RB size	3	4	5	6	7	8	9	10	11	12	13	14	15	15

Table 5: Test case (b): Size of the reduced basis in the standard algorithm with $\epsilon_{\text{POD}} = 5 \cdot 10^{-2}$.

m	2	10	13	20	30	36	37	79	96	144
$\|r_m\|_{\ell^\infty(\Omega^{\text{tr}})}$	1.6	1.3	9.7E−1	4.7E−1	1.7E−1	1.2E−1	8.0E−2	9.1E−3	4.6E−3	9.4E−4

Table 6: Test case (b): Evolution of the standard EIM error. m is the rank of the EIM approximation and $\|r_m\|_{\ell^\infty(\Omega^{\text{tr}})}$ is the residual norm in (18).

During the standard offline stage, we perform $P = 40$ HF computations. Knowing that $K = 50$, the set $\mathcal{S}$ (Algorithm 2, line 1) contains 2040 fields, each consisting of $\mathcal{N} = 1429$ nodal values. Applying Algorithm 3 based on the H^1-norm, a truncation threshold $\epsilon_{\text{POD}} = 5 \cdot 10^{-2}$, and parameters enumerated with increasing values, we obtain $N = 15$ RB functions. Table 5 shows the dimension of the RB space as a function of the enumeration index p. Table 6 shows the evolution of the error on the nonlinearity within the standard EIM for selected values of the rank of the EIM approximation. The fact that the nonlinearity depends on the partial derivatives of the solution challenges the EIM;

indeed, the error decay is not as fast as in the previous test case. This observation is corroborated by the fact that the functions $(q_j)_{1\leq j\leq M}$ all look quite different (not shown for brevity).

m	1	2	3	4	5	6	7	8	9
$\bar{\mu}$	21	8	21	8	21	21	21	8	9
μ	21	8	21	8	21	21	21	8	9
k	2	5	3	2	50	4	49	3	4

m	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
$\bar{\mu}$	21	21	21	8	21	21	21	8	9	21	9	21	9	9	9	6	21	21	40	40
μ	21	8	21	8	21	21	21	8	9	21	9	7	6	9	9	5	4	3	40	40
k	2	5	3	2	50	4	49	3	4	10	50	25	49	5	10	4	6	9	15	40

Table 7: Test case (b): Selected parameters and time nodes in PREIM for $\epsilon_{\text{EIM}} = 10^{-1}$ (top) and $\epsilon_{\text{EIM}} = 10^{-3}$ (bottom). The gray cells correspond to a new parameter selection and, therefore, to a new HF computation.

m	1	2	9	17	18	20
RB size	5	6	6	7	9	9

Table 8: Test case (b): Size of RB generated within PREIM for $\epsilon_{\text{POD}} = 5 \cdot 10^{-2}$; for $\epsilon_{\text{EIM}} = 10^{-1}$, one stops at $m = 9$, and for $\epsilon_{\text{EIM}} = 10^{-3}$, one stops at $m = 20$.

We now investigate the performance of PREIM, which we run with thresholds $\epsilon_{\text{POD}} = 5 \cdot 10^{-2}$ and either $\epsilon_{\text{EIM}} = 10^{-1}$ or $\epsilon_{\text{EIM}} = 10^{-3}$. Table 7 shows the selected parameters and time nodes at each iteration. For $\epsilon_{\text{EIM}} = 10^{-1}$, PREIM performs 9 iterations, and three parameters are selected for HF computations, whereas for $\epsilon_{\text{EIM}} = 10^{-3}$, PREIM performs 11 further iterations and six more HF computations to reach the requested threshold. Moreover, the evolution of the size of the reduced basis within PREIM is shown in Table 8 for selected values of the rank of the EIM approximation. As can be noticed, the approximation of the nonlinearity requires more computational effort than that of the solution manifold.

Figure 8 shows the decrease of the EIM approximation error on the nonlinearity for PREIM with $\epsilon_{\text{POD}} = 5 \cdot 10^{-2}$ and $\epsilon_{\text{EIM}} = 10^{-3}$. We observe that each time a new HF trajectory is computed, i.e., whenever the quantities $\|\widetilde{r}_m\|_{\ell^\infty(\Omega^{\text{tr}})}$ and $\|\bar{r}_m\|_{\ell^\infty(\Omega^{\text{tr}})}$ differ, the difference is actually rather small, thereby confirming the already accurate approximation of the nonlinearity by the RB solutions in PREIM. The left panel of Figure 9 illustrates the space-time errors (measured in the $\ell^2(I^{\text{tr}}; H^1(\Omega^{\text{tr}}))$-norm) on the trajectories produced by the standard RB-EIM and the PREIM procedures for the whole parameter range. We observe that for lower parameter values, PREIM delivers somewhat less accurate results, whereas the conclusion is reversed for higher parameter values. Altogether, both errors stay within comparable upper bounds. The right panel of Figure 9 shows that the error component associated with the approximation of the nonlinearity is still the dominant one, except for the parameter range $[1, 5]$, where the RB from the standard algorithm improves the error. Incidentally, we observe that these smaller values of the parameter were not selected within PREIM for approximating the nonlinearity. Finally, Figure 10 shows the same results for the tighter thresholds $\epsilon_{\text{POD}} = 5 \cdot 10^{-2}$ and $\epsilon_{\text{EIM}} = 10^{-4}$. Here, 14 HF computations and 100 PREIM iterations were needed. We can see that the PREIM error closely matches that of the standard RB-EIM procedure.

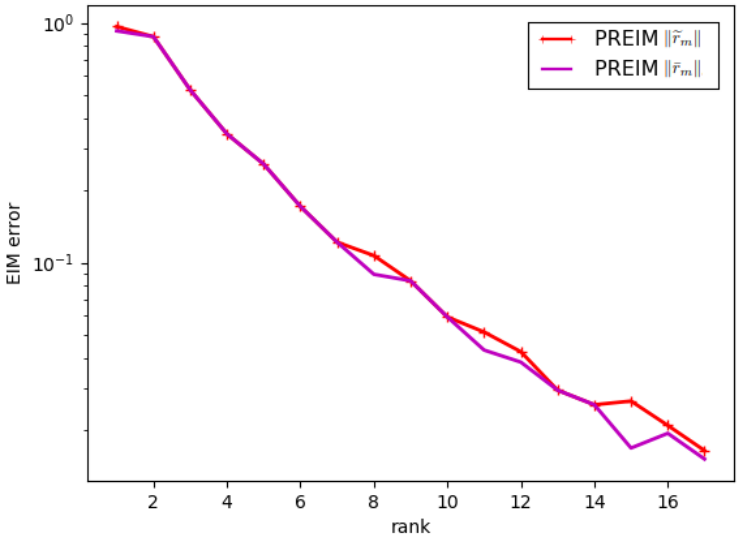

Figure 8: Test case (b): EIM approximation errors $\|\widetilde{r}_m\|_{\ell^\infty(\Omega^{\mathrm{tr}})}$ and $\|\bar{r}_m\|_{\ell^\infty(\Omega^{\mathrm{tr}})}$ as a function of m for PREIM with $\epsilon_{\mathrm{POD}} = 5 \cdot 10^{-2}$ and $\epsilon_{\mathrm{EIM}} = 10^{-3}$.

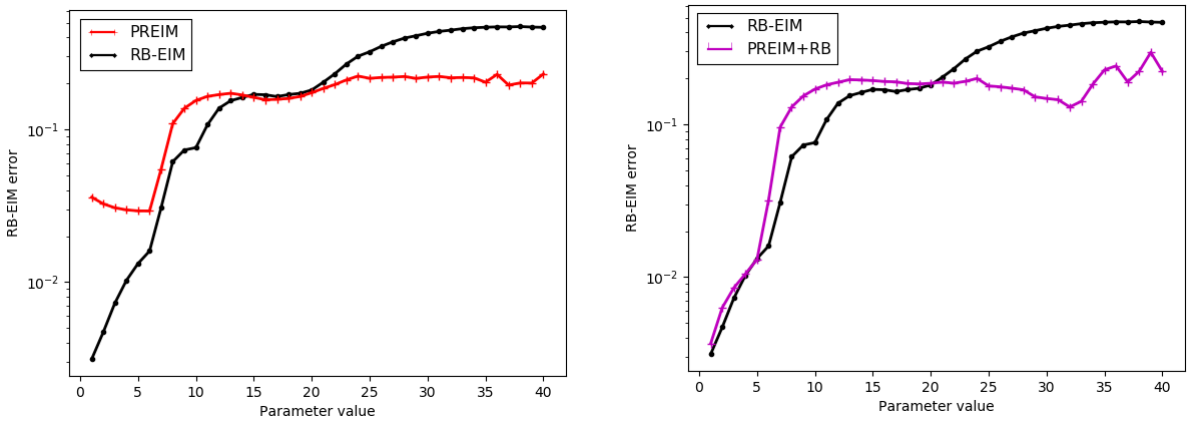

Figure 9: Test case (b): RB approximation error $\|u_\mu - \hat{u}_\mu\|_{\ell^2(I^{\mathrm{tr}};H^1(\Omega^{\mathrm{tr}}))}$ for $\epsilon_{\mathrm{POD}} = 5 \cdot 10^{-2}$ and $\epsilon_{\mathrm{EIM}} = 10^{-3}$.

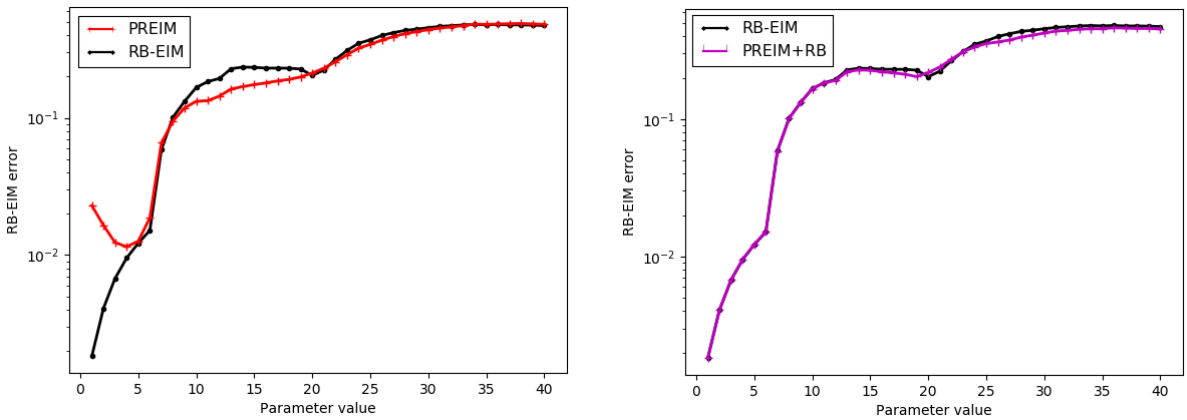

Figure 10: Test case (b): RB approximation error $\|u_\mu - \hat{u}_\mu\|_{\ell^2(I^{\mathrm{tr}};H^1(\Omega^{\mathrm{tr}}))}$ for $\epsilon_{\mathrm{POD}} = 2.5 \cdot 10^{-2}$ and $\epsilon_{\mathrm{EIM}} = 10^{-4}$.

6.3 Test case (c): 3D industrial valve prototype

Here, we present a three-dimensional test case whose geometry is based on a flow regulation valve used in nuclear reactor operation. We consider the nonlinear parabolic problem (1) with the nonlinear function $\Gamma(\mu, v) := \sin\left(\frac{\pi\mu}{20}\left(\frac{v-u_0}{u_m-u_0}\right)^2\right)$, with $u_0 = 293$ K (20 ^{o}C) and $u_m = 303$ K (30 ^{o}C). We define $\kappa_0 = 1.05$ m^2·K^{-2}·s^{-1} and $\phi_e = 3$ K·m·s^{-1}. For space discretization, we use a mesh containing $\mathcal{N} = 46,018$ nodes (see Figure 11). Regarding time discretization, we consider the time interval

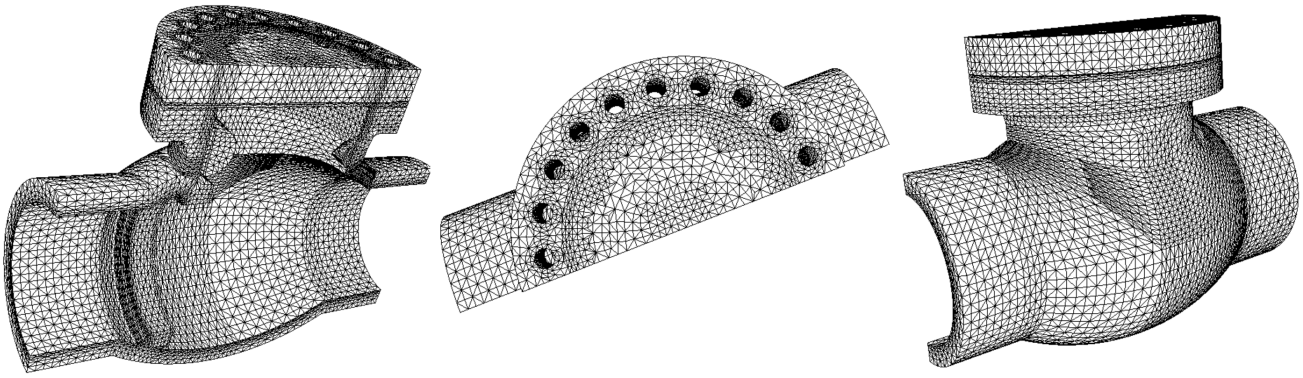

Figure 11: Test case (c): half-section of an industrial flow-regulation valve.

$I = [0, 1.5]$, the set of discrete times nodes $\mathbb{K}^{\text{tr}} = \{1, \cdots, 30\}$, and a constant time step $\Delta t^k = 0.05$ s for all $k \in \mathbb{K}^{\text{tr}}$. Finally, we consider the parameter interval $\mathcal{P} = [1, 20]$ and the training set $\mathcal{P}^{\text{tr}} = \{1, \cdots, 20\}$. During the standard offline stage, we perform $P = 20$ HF computations. Knowing

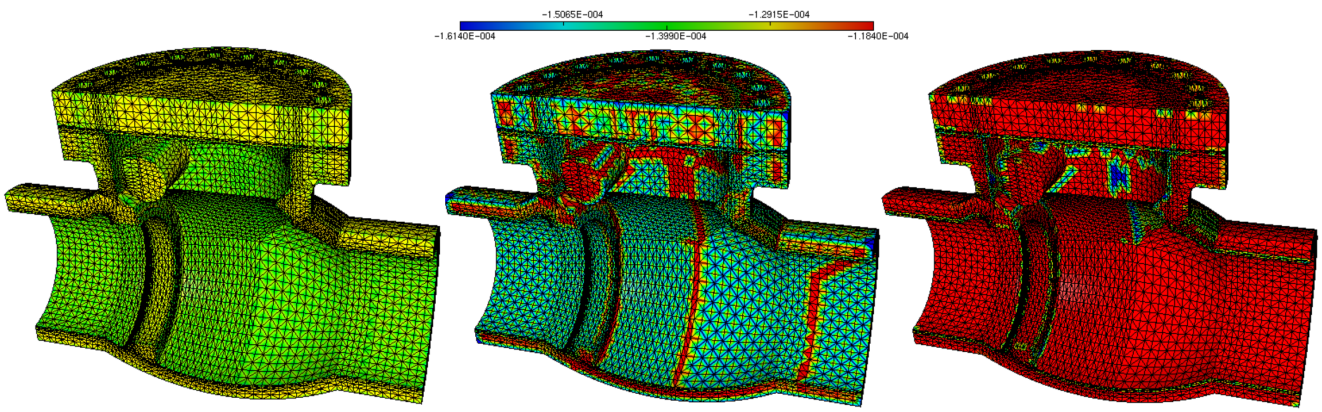

Figure 12: Test case (c): 2nd, 3rd, and 4th progressive-POD modes.

m	1	2	3	5	7	8	11
$\lVert r_m \rVert_{\ell^\infty(\Omega^{\text{tr}})}$	1.0	5.5E−1	2.7E−1	5.2E−2	1.1E−2	7.5E−3	1.5E−3

Table 9: Test case (c): Evolution of the standard EIM error. m is the rank of the EIM approximation and $\lVert r_m \rVert_{\ell^\infty(\Omega^{\text{tr}})}$ is the residual norm in (18).

that $K = 31$, the set $\mathcal{S}$ (Algorithm 2, line 1) contains 620 fields, each consisting of $\mathcal{N} = 46,018$ nodal values. Applying the POD in a progressive manner based on the H^1-norm and a relative truncation

threshold $\epsilon_{\text{POD}} = 10^{-3}$ defined as suggested in Remark 3, we obtain $N = 4$ RB functions. Figure 12 shows three POD modes. Afterwards, we perform the standard EIM algorithm whose convergence is reported in Table 9.

m		1	2	3	4	5	6	7	8	9	10
PREIM	$\bar{\mu}$	20	20	10	20	20	20	8	20	20	13
	$\bar{k}$	30	14	30	21	5	26	22	3	9	30
	μ	20	11	10	20	19	9	8	12	13	7
	k	31	31	31	21	31	31	31	31	31	31

Table 10: Test case (c): Selected parameters and time nodes in PREIM for $\epsilon_{\text{POD}} = 10^{-3}$ and $\epsilon_{\text{EIM}} = 5 \cdot 10^{-3}$. The gray cells correspond to a new parameter selection.

We now investigate PREIM, which we first run with thresholds $\epsilon_{\text{POD}} = 10^{-3}$ and $\epsilon_{\text{EIM}} = 5 \cdot 10^{-3}$. Table 10 shows the selected parameters and discrete time nodes at each stage of PREIM. Out of ten iterations, we can see that a parameter re-selection happened at four iterations. For some of the remaining iterations, the selected time nodes have been changed at the re-selction step. The left

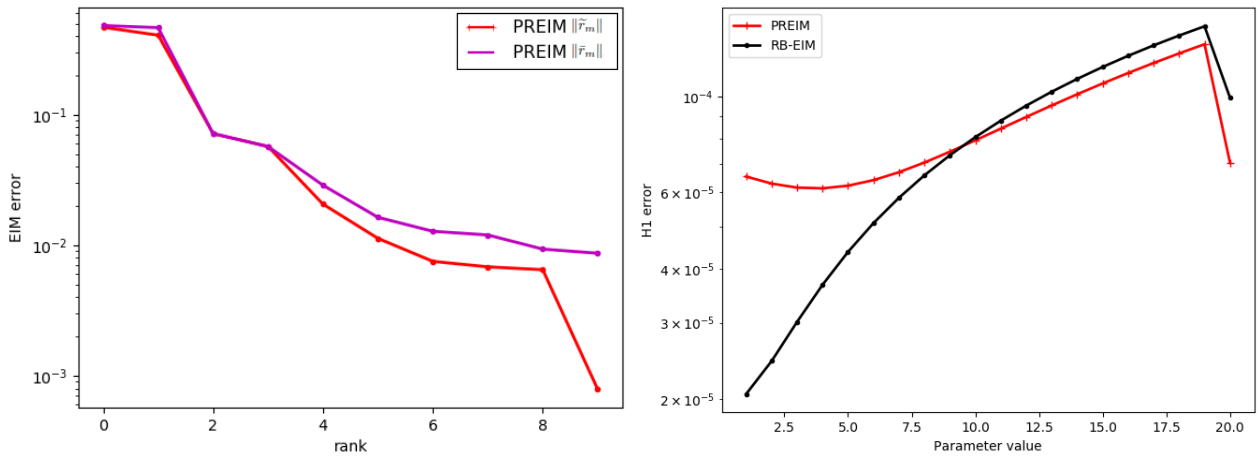

Figure 13: Test case (c): Left: EIM approximation errors $\|\widetilde{r}_m\|_{\ell^\infty(\Omega^{\text{tr}})}$ and $\|\bar{r}_m\|_{\ell^\infty(\Omega^{\text{tr}})}$ as a function of m for PREIM with $\epsilon_{\text{POD}} = 10^{-4}$ and $\epsilon_{\text{EIM}} = 5 \cdot 10^{-3}$. Right: RB approximation error $\|u_\mu - \hat{u}_\mu\|_{\ell^2(I^{\text{tr}};H^1(\Omega^{\text{tr}}))}/\|u_\mu\|_{\ell^2(I^{\text{tr}};H^1(\Omega^{\text{tr}}))}$ for $\epsilon_{\text{POD}} = 10^{-3}$ and $\epsilon_{\text{EIM}} = 5 \cdot 10^{-3}$.

panel of Figure 13 shows the decrease of the EIM approximation error on the nonlinearity within PREIM for $\epsilon_{\text{POD}} = 10^{-4}$ and $\epsilon_{\text{EIM}} = 5 \cdot 10^{-3}$. Overall, the difference between $\|\widetilde{r}_m\|_{\ell^\infty(\Omega^{\text{tr}})}$ and $\|\bar{r}_m\|_{\ell^\infty(\Omega^{\text{tr}})}$ is rather small except for $m = 9$ where Table 10 shows that the first selection has been discarded. The right panel of Figure 13 illustrates the space-time errors (measured in the relative $\ell^2(I^{\text{tr}}; H^1(\Omega^{\text{tr}}))$-norm) on the trajectories produced by the standard RB-EIM and the PREIM procedures for the whole parameter range; here the tolerances are set to $\epsilon_{\text{POD}} = 10^{-3}$ and $\epsilon_{\text{EIM}} = 5 \cdot 10^{-3}$. We observe that for higher parameter values, PREIM delivers somewhat more accurate results, whereas the conclusion is reversed for lower parameter values. Altogether, both errors stay within comparable upper bounds. Figure 14 displays the space-time errors for the tighter tolerance $\epsilon_{\text{POD}} = 5 \cdot 10^{-4}$ that delivers more accurate reluts as expected. Moreover, tightening ϵ_{EIM} makes both the RB-EIM and

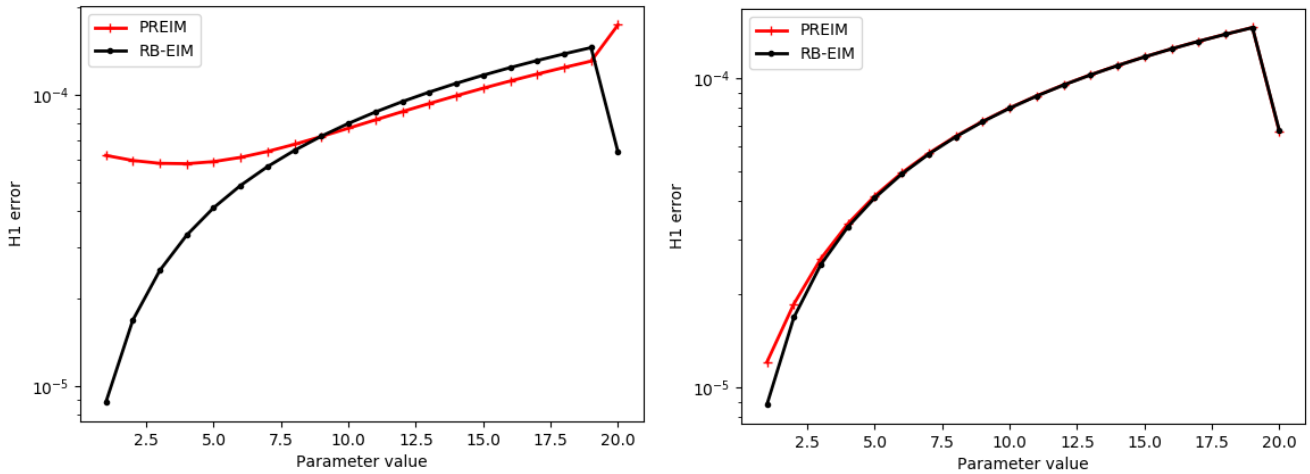

Figure 14: Test case (c): RB approximation error $\|u_\mu - \hat{u}_\mu\|_{\ell^2(I^{\mathrm{tr}};H^1(\Omega^{\mathrm{tr}}))}/\|u_\mu\|_{\ell^2(I^{\mathrm{tr}};H^1(\Omega^{\mathrm{tr}}))}$.
Left: $\epsilon_{\mathrm{POD}} = 10^{-4}$ and $\epsilon_{\mathrm{EIM}} = 5 \cdot 10^{-1}$. Right: $\epsilon_{\mathrm{POD}} = 10^{-4}$ and $\epsilon_{\mathrm{EIM}} = 5 \cdot 10^{-3}$.

PREIM errors overlap. Thus, one can infer that the dominant error in this test case is rather the RB error; whence the numerous HF computations that need to be performed, as seen in Table 10. Still, for such tight tolerances and with quasi-identical output errors (cf. right panel of Figure 14), PREIM makes less than half of the HF computations incurred in the standard RB-EIM.

$\epsilon_{\mathrm{EIM}} = 5.10^{-1}$	RB-EIM	PREIM
HF computations	99.8%	10.0%
greedy runtime	0.2%	0.6%
Total runtime	100%	10.6%

$\epsilon_{\mathrm{EIM}} = 5.10^{-2}$	RB-EIM	PREIM
HF computations	99.6%	20.0%
greedy runtime	0.4%	2.4%
Total runtime	100%	22.4%

Table 11: Test case (c): Runtime measurements with $\epsilon_{\mathrm{RB}} = 10^{-4}$. Left: $\epsilon_{\mathrm{EIM}} = 5 \cdot 10^{-1}$. Right: $\epsilon_{\mathrm{EIM}} = 5 \cdot 10^{-2}$.

Finally, we provide an assessment of the runtimes in Table 11. One can notice that the greedy procedure accounts for a slightly greater percentage of the offline stage in PREIM compared to the standard RB-EIM. This is mainly due to the additional intermediate calculations in PREIM. However, as previously shown, the dominant part of the offline stage are the HF computations; this illustrates again the relevance of using PREIM.

7 Conclusion and perspectives

In this work, we have devised a new methodology, called PREIM, and we have shown on three test cases, including a 3D industrial prototype, that this methodology allows one to diminish the offline expenses incurred in the nonlinear RB method applied to unsteady nonlinear PDEs. The main reason for this computational gain is that the computation of high-fidelity trajectories is the dominant part of the offline cost. In the present study, the computational benefits of PREIM have been evaluated by comparing the results to those of the reference method, i.e. the standard RB-EIM and also to the SER method which is the closest progressive method to PREIM available in the literature. Comparisons with other progressive RB-EIM methods, such as those mentioned in the

introduction, and a more theoretical study of PREIM are left to future work. Moreover, a relevant perspective of the present work is the application of PREIM to a more systematic study of 3D flow regulation in industrial applications related to nuclear reactor operation.

Acknowledgments

This work has been partially supported by Electricité De France (EDF) and a CIFRE PhD fellowship from ANRT. The authors are grateful to G. Blatman (EDF) and M. Abbas (EDF) for stimulating discussions and for their help on `code_aster`. The authors are thankful to the two anonymous reviewers for their valuable comments.

A Proper Orthogonal Decomposition

The goal of this appendix is to briefly describe the procedure associated with the notation

$$(\theta_1, \dots, \theta_N) = \mathrm{POD}(\mathcal{S}, \epsilon_{\mathrm{POD}}), \tag{27}$$

which is used in Algorithms 3, 4, and 6, where $\mathcal{S} = (v_1, \dots, v_R)$ is composed of $R \geq 1$ functions in the space X and ϵ_{POD} is a user-prescribed tolerance. For simplicity, we adopt an algebraic description, and we refer the reader to [8] for further insight. Let $(\varrho_1, \dots, \varrho_{\mathcal{N}})$ be a basis of X, where $\dim(X) = \mathcal{N}$. For a function $w \in X$, we denote by $\mathbf{w} := (w_j)_{1 \leq j \leq \mathcal{N}}$ its coordinate vector in $\mathbb{R}^{\mathcal{N}}$, so that $w = \sum_{j=1}^{\mathcal{N}} w_j \varrho_j$. The algebraic counterpart of (27) is that we are given R vectors forming the rectangular matrix $\mathbf{S} := (\mathbf{v}_1, \dots, \mathbf{v}_R) \in \mathbb{R}^{\mathcal{N} \times R}$, and we are looking for N vectors forming the rectangular matrix $\boldsymbol{\Theta} := (\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_N) \in \mathbb{R}^{\mathcal{N} \times N}$. The vectors $(\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_N)$ are to be orthonormal with respect to the Gram matrix of the inner product in X. In the present setting, we consider the Gram matrix $\mathbf{C}^{\mathcal{N}} \in \mathbb{R}^{\mathcal{N} \times \mathcal{N}}$ such that

$$\mathbf{C}^{\mathcal{N}} = \Big(m(\varrho_n, \varrho_p) + \eta a_0(\varrho_n, \varrho_p) \Big)_{1 \leq p, n \leq \mathcal{N}}, \tag{28}$$

where $\eta > 0$ is a user-prescribed weight and the bilinear forms m and a_0 are defined in (3). Thus, we want to have $\boldsymbol{\theta}_n^{\mathrm{T}} \mathbf{C}^{\mathcal{N}} \boldsymbol{\theta}_p = \delta_{n,p}$, the Kronecker delta, for all $n, p \in \{1, \dots, N\}$.

Let us set $\mathbf{T} := (\mathbf{C}^{\mathcal{N}})^{\frac{1}{2}} \mathbf{S} \in \mathbb{R}^{\mathcal{N} \times R}$ and consider the integer $D = \min(\mathcal{N}, R)$ (in most situations, we have $D = R$ and $D \ll \mathcal{N}$). Computing the Singular Value Decomposition [16] of the matrix $\mathbf{T}$, we obtain the real numbers $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_D \geq 0$, the orthonormal family of column vectors $(\boldsymbol{\xi}_n)_{1 \leq n \leq D} \in (\mathbb{R}^{\mathcal{N}})^D$ (so that $\boldsymbol{\xi}_n^{\mathrm{T}} \boldsymbol{\xi}_p = \delta_{p,n}$) and the orthonormal family of column vectors $(\hat{\boldsymbol{\psi}}_n)_{1 \leq n \leq D} \in (\mathbb{R}^R)^D$ (so that $\hat{\boldsymbol{\psi}}_n^{\mathrm{T}} \hat{\boldsymbol{\psi}}_p = \delta_{p,n}$), and we have

$$\mathbf{T} = \sum_{n=1}^{D} \sigma_n \boldsymbol{\xi}_n \hat{\boldsymbol{\psi}}_n^{\mathrm{T}}. \tag{29}$$

From (29), it follows that $\mathbf{T} \hat{\boldsymbol{\psi}}_n = \sigma_n \boldsymbol{\xi}_n$ and $\mathbf{T}^{\mathrm{T}} \boldsymbol{\xi}_n = \sigma_n \hat{\boldsymbol{\psi}}_n$ for all $n \in \{1, \dots, D\}$. The vectors we are looking for are then given by $\boldsymbol{\theta}_n := (\mathbf{C}^{\mathcal{N}})^{-\frac{1}{2}} \boldsymbol{\xi}_n$ for all $n \in \{1, \dots, N\}$ with $N := \max\{1 \leq n \leq D \mid \sigma_n \geq \epsilon_{\mathrm{POD}}\}$. It is well-known that the N-dimensional space spanned by the vectors $(\boldsymbol{\theta}_n)_{1 \leq n \leq N}$ minimizes the quantity $\inf_{\mathbf{z} \in \mathbf{Z}_N} \sum_{r=1}^{R} (\mathbf{v}_r - \mathbf{z})^{\mathrm{T}} \mathbf{C}^{\mathcal{N}} (\mathbf{v}_r - \mathbf{z})$ among all the N-dimensional subspaces $\mathbf{Z}_N$ of $\mathbb{R}^{\mathcal{N}}$. Moreover, we have $\|v - \Pi_{Z_N} v\|_X \leq \sigma_{N+1} \|v\|_X$, for all $v \in \mathcal{S}$.

In practice, when $D = R$, we can avoid the computation of the matrix $(\mathbf{C}^{\mathcal{N}})^{\frac{1}{2}}$ and of its inverse by considering the matrix of smaller dimension $\mathbf{T}^{\mathrm{T}} \mathbf{T} = \mathbf{S}^{\mathrm{T}} \mathbf{C}^{\mathcal{N}} \mathbf{S} \in \mathbb{R}^{R \times R}$. Solving for the eigenvalues

of $\mathbf{T}^{\mathrm{T}}\mathbf{T}$, we obtain the vectors $\hat{\boldsymbol{\psi}}_n$ with associated eigenvalues σ_n^2 since we have $\mathbf{T}^{\mathrm{T}}\mathbf{T}\hat{\boldsymbol{\psi}}_n = \sigma_n \mathbf{T}^{\mathrm{T}}\boldsymbol{\xi}_n = \sigma_n^2 \hat{\boldsymbol{\psi}}_n$. Then, the vectors $(\boldsymbol{\theta}_n)_{1\leq n\leq N}$ are obtained as $\boldsymbol{\theta}_n = \left(\mathbf{C}^{\mathcal{N}}\right)^{-\frac{1}{2}} \boldsymbol{\xi}_n = \frac{1}{\sigma_n}\left(\mathbf{C}^{\mathcal{N}}\right)^{-\frac{1}{2}} \mathbf{T}\hat{\boldsymbol{\psi}}_n = \frac{1}{\sigma_n}\mathbf{S}\hat{\boldsymbol{\psi}}_n$.

References

[1] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera, *An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations*, C. R. Math. Acad. Sci. Paris, 339 (2004), pp. 667–672.

[2] C. Daversin and C. Prud'homme, *Simultaneous empirical interpolation and reduced basis method for non-linear problems*, C. R. Math. Acad. Sci. Paris, 353 (2015), pp. 1105–1109.

[3] M. Drohmann, B. Haasdonk, and M. Ohlberger, *Reduced basis approximation for nonlinear parametrized evolution equations based on empirical operator interpolation*, SIAM J. Sci. Comput., 34 (2012), pp. A937–A969.

[4] Electricité de France, *Finite element code_aster, analysis of structures and thermomechanics for studies and research.* Open source on www.code-aster.org, 1989–2017.

[5] A. Ern and J.-L. Guermond, *Theory and Practice of Finite Elements*, Applied Mathematical Sciences, Springer New York, 2004.

[6] M. A. Grepl, *Certified reduced basis methods for nonaffine linear time-varying and nonlinear parabolic partial differential equations*, Math. Models Methods Appl. Sci., 22 (2012), pp. 1150015, 40.

[7] M. A. Grepl, Y. Maday, N. C. Nguyen, and A. T. Patera, *Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations*, M2AN Math. Model. Numer. Anal., 41 (2007), pp. 575–605.

[8] M. Gubisch and S. Volkwein, *Chapter 1: Proper orthogonal decomposition for linear-quadratic optimal control*, in Model Reduction and Approximation, pp. 3–63.

[9] B. Haasdonk, *Convergence rates of the POD-greedy method*, ESAIM Math. Model. Numer. Anal., 47 (2013), pp. 859–873.

[10] F. Hecht, *FreeFem++, Third Edition, Version 3.0-1. User's Manual*, LJLL, University Paris VI.

[11] J. S. Hesthaven, G. Rozza, and B. Stamm, *Certified reduced basis methods for parametrized partial differential equations*, SpringerBriefs in Mathematics, Springer, Cham; BCAM Basque Center for Applied Mathematics, Bilbao, 2016. BCAM SpringerBriefs.

[12] M. Hinze and S. Volkwein, *Proper orthogonal decomposition surrogate models for nonlinear dynamical systems: error estimates and suboptimal control*, in Dimension reduction of large-scale systems, vol. 45 of Lect. Notes Comput. Sci. Eng., Springer, Berlin, 2005, pp. 261–306.

[13] K. Kunisch and S. Volkwein, *Galerkin proper orthogonal decomposition methods for parabolic problems*, Numer. Math., 90 (2001), pp. 117–148.

[14] L. Machiels, Y. Maday, I. B. Oliveira, A. T. Patera, and D. V. Rovas, *Output bounds for reduced-basis approximations of symmetric positive definite eigenvalue problems*, C. R. Acad. Sci. Paris Sér. I Math., 331 (2000), pp. 153–158.

[15] Y. Maday, N. C. Nguyen, A. T. Patera, and G. S. H. Pau, *A general multipurpose interpolation procedure: the magic points*, Commun. Pure Appl. Anal., 8 (2009), pp. 383–404.

[16] B. Noble and J. W. Daniel, *Applied Linear Algebra*, Prentice-Hall, 3rd ed., 1988.

[17] C. Prud'Homme, D. V. Rovas, K. Veroy, L. Machiels, Y. Maday, A. T. Patera, and G. Turinici, *Reliable Real-Time Solution of Parametrized Partial Differential Equations: Reduced-Basis Output Bound Methods*, Journal of Fluids Engineering, 124 (2001), pp. 70–80.

[18] A. Quarteroni, A. Manzoni, and F. Negri, *Reduced basis methods for partial differential equations*, La Matematica per il 3+2, Springer International Publishing, 2016.

[19] D. V. Rovas, L. Machiels, and Y. Maday, *Reduced-basis output bound methods for parabolic problems*, IMA J. Numer. Anal., 26 (2006), pp. 423–445.

[20] K. Urban and B. Wieland, *Affine decompositions of parametric stochastic processes for application within reduced basis methods*, IFAC Proceedings Volumes, 45 (2012), pp. 716–721.