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

Amina Benaceur, Virginie Ehrlacher, Alexandre Ern, Sébastien Meunier

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 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 full-order computations whenever available and RB computations otherwise. Another key feature of PREIM is to select twice the parameter in a greedy fashion, the second selection being made after computing the full-order solution 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 [11, 14] (see also the recent textbooks [8, 15]) 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 situations, where a problem needs to be solved very quickly under limited computational resources, or in multi-query scenarios, 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) (also called Full-Order) trajectories are computed for a training subset of parameters $\mathcal{P}^{\rm tr} \subsetneq \mathcal{P}$; (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 (typically a finite element space based on a fine

*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

mesh) 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.

In the present work, we are interested in nonlinear parabolic problems for which a RB method has been successfully developed in [5, 6]. A key ingredient to treat the nonlinearity so as to enable an online stage without HF computations is the Empirical Interpolation Method (EIM) [1, 12]. The EIM provides a separated 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 (λ, k) to build the EIM approximation (where λ is the parameter and k refers to the discrete time node) can lead to repeated values of λ for many different values of k. In other words, PREIM can select multiple space fields within the same 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.

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 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 [5, 6], and in Section 4, we briefly recall the EIM procedure in the standard offline stage as devised in [1, 12]. 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. 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 $\lambda \in \mathcal{P}$. Our goal is to solve the following nonlinear parabolic PDE for many values of the parameter $\lambda \in \mathcal{P}$: find $u_\lambda : I \times \Omega \to \mathbb{R}$ such that

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

where $\kappa_0 > 0$, $\Gamma : \mathcal{P} \times \mathbb{R} \to \mathbb{R}$ 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 source term at the boundary $\partial\Omega$, and $u_0 : \Omega \to \mathbb{R}$ is the initial condition. For simplicity, we assume for the time being that f, ϕ_e, and u_0 are parameter-independent. In our numerical experiments in Section 6, we will also consider an additional parameter-dependent reaction term. 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 $\epsilon_1 \leq \kappa_0 + \Gamma(\lambda, z) \leq \beta_1$ with $0 < \epsilon_1 < \beta_1 < \infty$, for all $(\lambda, z) \in \mathcal{P} \times \mathbb{R}$. We do not specify a functional setting for the nonlinear parabolic PDE (1); with the above assumptions, it is reasonable to look for a weak solution $u \in L^2(I; Y)$ with $Y = H^1(\Omega)$.

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 snapshots in time and for certain parameter values of u_λ. 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*

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

where $\alpha(u_\lambda)$ represents the product of the mass density times the heat capacity. Moreover, the quantity $(\kappa_0 + \Gamma(\lambda, u_\lambda))$ 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 [4] 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 $\Omega^{\mathrm{tr}} \subsetneq \Omega$, where $\mathrm{Card}(\Omega^{\mathrm{tr}}) = \mathcal{N}$. 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 $\mathbb{K}^{\mathrm{tr}} = \{1, \cdots, K\}$, $\overline{\mathbb{K}}^{\mathrm{tr}} = \{0\} \cup \mathbb{K}^{\mathrm{tr}}$, $I^{\mathrm{tr}} = \{t^k\}_{k \in \overline{\mathbb{K}}^{\mathrm{tr}}}$, and $\Delta t^k = t^k - t^{k-1}$ for all $k \in \mathbb{K}^{\mathrm{tr}}$. 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_λ for all $\lambda \in \mathcal{P}$. Henceforth, we use the convention that the superscript k always indicates a time index; thus, we write $u_\lambda^k(\cdot) = u_\lambda(t^k, \cdot) \in X$. Applying a semi-implicit Euler scheme, our goal is, given $u_\lambda^0 = u_0 \in X$, to find $(u_\lambda^k)_{k\in\mathbb{K}^{\rm tr}} \in X^K$ such that, for all $k \in \mathbb{K}^{\rm tr}$,

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

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

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

for all $\lambda \in \mathcal{P}$ and all $v, w \in Y$. In (2), the nonlinearity is treated explicitly, whereas the diffusive term is treated implicitly. This choice is made only for simplicity since it avoids dealing with a nonlinear solver at each time-step. The PREIM method developed in this work can be applied also if another time scheme is considered.

3 The Reduced-Basis method

In this section, we briefly recall the Reduced-Basis (RB) method for the nonlinear problem (2) [6, 5]. 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 $(\theta_n)_{1\le n\le N}$ be a Y-orthonormal basis of $\hat{X}_N$. For all $\lambda \in \mathcal{P}$ and $k \in \overline{\mathbb{K}}^{\rm tr}$, the RB solution $\hat{u}_\lambda^k \in \hat{X}_N$ that approximates the HF solution $u_\lambda^k \in X$ is decomposed as

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

with real numbers $\hat{u}_{\lambda,n}^k$ for all $n \in \{1, \dots, N\}$. Let us introduce the vector notation $\hat{\mathbf{u}}_\lambda^k := (\hat{u}_{\lambda,n}^k)_{1\le n\le N} \in \mathbb{R}^N$, for all $\lambda \in \mathcal{P}$ and $k \in \overline{\mathbb{K}}^{\rm tr}$. Let $\hat{u}^0$ be the Y-orthogonal projector 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 $u_\lambda^k \in X$ in the weak form (2) by the approximation $\hat{u}_\lambda^k \in \hat{X}_N$ with associated component vector $\hat{\mathbf{u}}_\lambda^k \in \mathbb{R}^N$, and using the test functions $(\theta_p)_{1\le p\le N}$, we obtain the following problem written in algebraic form: Given $\hat{\mathbf{u}}_\lambda^0 = \hat{\mathbf{u}}^0 \in \mathbb{R}^N$, find $(\hat{\mathbf{u}}_\lambda^k)_{k\in\mathbb{K}^{\rm tr}} \in (\mathbb{R}^N)^K$ such that, for all $k \in \mathbb{K}^{\rm tr}$,

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

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 $\mathbf{G}(\hat{\mathbf{u}}_\lambda^{k-1}) \in \mathbb{R}^N$ such that

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

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

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

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

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

where M is called the rank of the approximation, q_m are real-valued functions defined on Ω and $\varphi_{\lambda,j}^k$ are real numbers satisfying

$$\forall i \in \{1, \cdots, M\}\ , \qquad \sum_{j=1}^{M} \mathbf{B}_{ij} \varphi_{\lambda,j}^k = \gamma(\lambda, k, x_i), \tag{11}$$

where $\mathbf{B} \in \mathbb{R}^{M \times M}$ is an invertible lower-triangular matrix and $(x_i)_{1 \le i \le M}$ are points in the subset Ω^{tr} introduced in the Finite Element Method. The function γ_M can be computed by means of the EIM as detailed in Section 4 below. Using (9), (10) and the RB approximation, we obtain

$$\Gamma(\lambda, \hat{u}_\lambda^{k-1}(x)) \approx \Gamma\big(\lambda, u_\lambda^{k-1}(x)\big) = \gamma(\lambda, k-1, x) \approx \gamma_M(\lambda, k-1, x) = \sum_{j=1}^{M} \varphi_{\lambda,j}^{k-1} q_j(x).$$

Moreover, (9) and (11) yield

$$\varphi_{\lambda,j}^{k-1} = \sum_{i=1}^{M} (\mathbf{B}^{-1})_{ji}\ \gamma(\lambda, k-1, x_i) = \sum_{i=1}^{M} (\mathbf{B}^{-1})_{ji}\ \Gamma(\lambda, u_\lambda^{k-1}(x_i)) \approx \sum_{i=1}^{M} (\mathbf{B}^{-1})_{ji}\ \Gamma\Big(\lambda, \hat{u}_\lambda^{k-1}(x_i)\Big),$$

which can be recast as

$$\varphi_{\lambda,j}^{k-1} = (\mathbf{B}^{-1} \boldsymbol{\gamma}_\lambda^{k-1})_j, \tag{12}$$

with the vector $\boldsymbol{\gamma}_\lambda^{k-1} \in \mathbb{R}^M$ such that

$$\boldsymbol{\gamma}_\lambda^{k-1} := \Big(\Gamma\Big(\lambda, \hat{u}_\lambda^{k-1}(x_i)\Big)\Big)_{1 \le i \le M}. \tag{13}$$

Using the EIM approximation, the problem (6) becomes: Given $\hat{\mathbf{u}}_\lambda^0 = \hat{\mathbf{u}}^0 \in \mathbb{R}^N$, find $(\hat{\mathbf{u}}_\lambda^k)_{k \in \mathbb{K}^{\mathrm{tr}}} \in (\mathbb{R}^N)^K$ such that, for all $k \in \mathbb{K}^{\mathrm{tr}}$,

$$(\mathbf{M} + \Delta t^k \mathbf{A}_0) \hat{\mathbf{u}}_\lambda^k = \Delta t^k \mathbf{f}^k + \big(\mathbf{M} - \Delta t^k \mathbf{D}_\lambda^{k-1}\big) \hat{\mathbf{u}}_\lambda^{k-1}, \tag{14}$$

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

$$\mathbf{D}_\lambda^{k-1} = \sum_{m=1}^{M} \big(\mathbf{B}^{-1} \boldsymbol{\gamma}_\lambda^{k-1}\big)_m \mathbf{C}^m, \qquad \mathbf{C}^m = \Big(\int_\Omega q_m \nabla \theta_n \cdot \nabla \theta_p\Big)_{1 \le p,n \le N}. \tag{15}$$

As is standard in the RB method, the overall computational procedure consists of two stages: (i) an offline stage (to be 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 $\lambda \in \mathcal{P}$. During the offline stage, 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 approximation γ_M of the nonlinear function γ leading to the matrices $\mathbf{B} \in \mathbb{R}^{M\times M}$ and $\mathbf{C}^m \in \mathbb{R}^{N\times N}$, for all $m \in \{1,\dots,M\}$. Thus, all what remains to be performed during the online stage is to compute the vector $\boldsymbol{\gamma}_\lambda^{k-1} \in \mathbb{R}^M$ and the matrix $\mathbf{D}_\lambda^{k-1} \in \mathbb{R}^{N\times N}$ and to solve the N-dimensional linear problem (14) for all $k \in \mathbb{K}^{\rm tr}$. The online stage is summarized in Algorithm 1.

Algorithm 1 Online stage

INPUT : $(\theta_n)_{1\le n\le N}$, $\hat{\mathbf{u}}^0$, $(\mathbf{f}^k)_{k\in\mathbb{K}^{\rm tr}}$, $\mathbf{M}$, $\mathbf{A}_0$, $\mathbf{B}$ and $(\mathbf{C}^m)_{1\le m\le M}$
1: Choose $\lambda \in \mathcal{P}$; set $k = 1$ and $\hat{\mathbf{u}}_\lambda^0 = \hat{\mathbf{u}}^0$
2: **while** $k \in \mathbb{K}^{\rm tr}$ **do**
3: Compute $\boldsymbol{\gamma}_\lambda^{k-1}$ and $\mathbf{D}_\lambda^{k-1}$ from $\hat{\mathbf{u}}_\lambda^{k-1}$ using (13) and (15)
4: Solve the reduced system (14) to obtain $\hat{\mathbf{u}}_\lambda^k$
5: Set $k = k + 1$
6: **end while**
OUTPUT : $(\hat{\mathbf{u}}_\lambda^k)_{k\in\mathbb{K}^{\rm tr}}$

4 The standard offline stage

There are two tasks to be performed during the offline stage: (i) explore the solution manifold $\{u_\lambda^k\}_{\lambda\in\mathcal{P},k\in\overline{\mathbb{K}}^{\rm tr}} \subset X$ in order to construct a linear subspace $\hat{X}_N \subset X$ as small as possible and with the best possible approximation properties; (ii) build the EIM approximation (10) with rank M of the nonlinear function γ defined by (9). In the standard offline stage, these two tasks are performed independently.

Let us first discuss Task (ii), i.e., the construction of the rank-M EIM approximation. Recall from Section 3 that the goal is to find the functions $(q_m)_{1\le m\le M}$ where $q_m : \Omega \to \mathbb{R}$, the interpolation points $(x_m)_{1\le m\le M}$ in $\Omega^{\rm tr} \subsetneq \Omega$, and the matrix $\mathbf{B} \in \mathbb{R}^{M\times M}$. For a real-valued function v defined on $\Omega^{\rm tr}$, we define $\|v\|_{\ell^\infty(\Omega^{\rm tr})} := \max_{x_i\in\Omega^{\rm tr}} |v(x_i)|$. Given $m \ge 1$ and γ_{m-1}, with the convention that $\gamma_0 \equiv 0$, an EIM iteration consists of the following steps. First, one defines $(\lambda_m, k_m) \in \mathcal{P}^{\rm tr} \times \overline{\mathbb{K}}^{\rm tr}$ by

$$(\lambda_m, k_m) \in \operatorname*{argmax}_{(\lambda,k)\in\mathcal{P}^{\rm tr}\times\overline{\mathbb{K}}^{\rm tr}} \|\Gamma(\lambda, u_\lambda^k(\cdot)) - \gamma_{m-1}(\lambda, k, \cdot)\|_{\ell^\infty(\Omega^{\rm tr})}, \tag{16}$$

where we notice the use of the HF trajectories for all values of the parameter λ in the training set $\mathcal{P}^{\rm tr} \subset \mathcal{P}$ of cardinality P. Once (λ_m, k_m) is determined, one sets

$$q_m(\cdot) := \frac{r_m(\cdot)}{r_m(x_m)}, \tag{17}$$

where

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

and

$$\mathbf{B}_{mi} := (q_i(x_m)), \qquad \forall 1 \leq i \leq m. \tag{19}$$

The standard EIM procedure is presented in Algorithm 2 (for simplicity, we do not mention in the output the functions $(q_m)_{1\leq m\leq M}$ and the points $(x_m)_{1\leq m\leq M}$).

Algorithm 2 Standard EIM

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

1: Compute $\mathcal{S} = (u^k_\lambda)_{\lambda\in\mathcal{P}^{\rm tr}, k\in\overline{\mathbb{K}}^{\rm tr}}$ P `HF trajectories`

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

3: **do**

4: Search $(\lambda_m, k_m) \in \underset{(\lambda,k)\in\mathcal{P}^{\rm tr}\times\overline{\mathbb{K}}^{\rm tr}}{\rm argmax} \|\Gamma(\lambda, u^k_\lambda(\cdot)) - \gamma_{m-1}(\lambda, k, \cdot)\|_{\ell^\infty(\Omega^{\rm tr})}$

5: Set $r_m(\cdot) := \Gamma(\lambda_m, u^{k_m}_{\lambda_m}(\cdot)) - \gamma_{m-1}(\lambda_m, k_m, \cdot)$

6: Set $x_m := \underset{x\in\Omega^{\rm tr}}{\rm argmax} |r_m(x)|$, $q_m := r_m/r_m(x_m)$, and compute $(\mathbf{B}_{mi})_{1\leq i\leq m}$

7: Set $m = m + 1$

8: **while** ($\|r_m\|_{\ell^\infty(\Omega^{\rm tr})} > \epsilon_{\rm EIM}$)

OUTPUT : $\mathbf{B}$

Let us now briefly discuss Task (i) above, i.e., the construction of the set of RB functions with cardinality N. Combined with an a posteriori error estimator, the exploration of the solution manifold can be performed by evaluating only N HF trajectories from a set of selected parameters $\hat{\mathcal{P}}_N \subsetneq \mathcal{P}^{\rm tr}$ (see, e.g., [16]). However, in the present setting where HF trajectories are to be computed for all parameters in $\mathcal{P}^{\rm tr}$ to construct the EIM approximation, it is natural to exploit these computations by means of a Proper Orthogonal Decomposition (POD) [9, 10] to define the RB functions. This technique is often considered in the literature to build the RB functions, see, e.g., [7, 8, 15]. The construction of the RB functions is presented in Algorithm 3. Here, for a given collection of HF trajectories $\mathcal{S}$ and a user-defined tolerance $\epsilon_{\rm POD}$, $(\boldsymbol{\theta}_n)_{1\leq n\leq N} = {\rm POD}(\mathcal{S}, \epsilon_{\rm POD})$ denotes the selected POD modes (the procedure is briefly outlined for completeness in Appendix A). To sum up, the standard offline stage consists in performing Algorithms 2 and 3 in whatever order (if Algorithm 3 is performed first, the computation of the matrices $(\mathbf{C}^m)_{1\leq m\leq M}$ is postponed to the end of Algorithm 2). Obviously, the P HF trajectories $(u^k_\lambda)_{\lambda\in\mathcal{P}^{\rm tr}, k\in\overline{\mathbb{K}}^{\rm tr}}$ appearing in both algorithms are computed only once.

Algorithm 3 Construction of the reduced basis

INPUT : $\mathcal{P}^{\rm tr} \subset \mathcal{P}$, $\overline{\mathbb{K}}^{\rm tr} = \{0, \cdots, K\}$ and $\epsilon_{\rm POD} > 0$

1: Compute $\mathcal{S} = (u^k_\lambda)_{\lambda\in\mathcal{P}^{\rm tr}, k\in\overline{\mathbb{K}}^{\rm tr}}$ P `HF trajectories`

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

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

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

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

5 The Progressive RB-EIM method (PREIM)

PREIM consists in a progressive improvement of the EIM and RB approximations. The key idea is that, unlike the standard EIM for which HF trajectories are computed for all the parameter values in the training set $\mathcal{P}^{\rm tr}$ (Algorithm 2, line 1), PREIM works with a subset $\mathcal{P}_m^{\rm HF} \subset \mathcal{P}^{\rm tr}$ that is enriched progressively with the iteration index m. The role of the subset $\mathcal{P}_m^{\rm HF}$ is to collect the parameter values for which a HF trajectory has already been computed. PREIM is designed so that $\mathrm{Card}(\mathcal{P}_m^{\rm HF}) \leq m$ for all $m \in \{1, \ldots, M\}$. 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 $\lambda \in \mathcal{P}_m^{\rm HF}$ are HF trajectories, whereas they are approximated by RB trajectories for all $\lambda \in \mathcal{P}^{\rm tr} \setminus \mathcal{P}_m^{\rm HF}$. The RB functions 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. To reflect this, we add a superscript m to the RB trajectories which are now denoted $(\hat{u}_\lambda^{m,k})_{k \in \overline{\mathbb{K}}^{\rm tr}}$ for all $\lambda \in \mathcal{P}^{\rm tr} \setminus \mathcal{P}_m^{\rm HF}$. It is convenient to introduce the notation

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

and the nonlinear function

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

The goal of every iteration $m \geq 1$ of PREIM is twofold: (i) produce a set of RB basis functions $(\theta_n^m)_{1 \leq n \leq N^m}$ (note that 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

$$\bar{\gamma}_m^m(\lambda, k, x) := \sum_{j=1}^m (\bar{\varphi}^m)_{\lambda,j}^k \bar{q}_j(x). \tag{22}$$

More precisely, this construction uses the interpolation points $(\bar{x}_j)_{1 \leq j \leq m}$ in the subset $\Omega^{\rm tr}$ with $\bar{x}_m$ computed at iteration m, together with the functions $(\bar{q}_j)_{1 \leq j \leq m}$ such that $\bar{q}_j : \Omega \to \mathbb{R}$ with $\bar{q}_m$ computed at iteration m. Then, considering the (invertible) lower-triangular matrix $\bar{\mathbf{B}} \in \mathbb{R}^{m \times m}$ whose last row is calculated using $\bar{\mathbf{B}}_{mi} = \bar{q}_i(\bar{x}_m)$ for all $i \in \{1, \ldots, m\}$, we compute the real numbers $(\bar{\varphi}^m)_{\lambda,j}^k$ in (22) from the relations

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

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

Let us now describe in more detail the iteration $m \geq 2$ of PREIM (see Algorithm 4); the initialization step $m = 1$ is described afterwards. First, one selects both a new pair (λ, k) in a greedy fashion, i.e., one finds $(\lambda_m, k_m) \in \mathcal{P}^{\rm tr} \times \overline{\mathbb{K}}^{\rm tr}$ such that

$$(\lambda_m, k_m) \in \operatorname*{argmax}_{(\lambda,k) \in \mathcal{P}^{\rm tr} \times \overline{\mathbb{K}}^{\rm tr}} \| \Gamma\big(\lambda, \bar{u}_\lambda^{m-1,k}(\cdot)\big) - \bar{\gamma}_{m-1}^{m-1}(\lambda, k, \cdot) \|_{\ell^\infty(\Omega^{\rm tr})}, \tag{24}$$

Note that owing to the definition (20) of $\bar{u}_\lambda^{m-1,k}$, the selection criterion exploits the knowledge of the HF trajectory whenever available and otherwise uses the RB trajectory. This is an

important difference with respect to the standard offline stage. There are now two possibilities: either λ_m is already in the subset $\mathcal{P}^{\mathrm{HF}}_{m-1}$, and in this case we simply set $\mathcal{P}^{\mathrm{HF}}_m := \mathcal{P}^{\mathrm{HF}}_{m-1}$; or λ_m is not in this subset, then we compute a new HF trajectory for λ_m, and we set $\mathcal{P}^{\mathrm{HF}}_m := \mathcal{P}^{\mathrm{HF}}_{m-1} \cup \{\lambda_m\}$. Our numerical experiments reported in Section 6 below will show that at many iterations of PREIM, the pair (λ_m, k_m) selected in (24) differs from the previously selected pairs by the time index and not by the parameter value; this means that for many PREIM iterations, no additional HF computation is performed. An additional feature of PREIM is that, whenever a new HF trajectory is actually computed, one can either confirm or update the selected pair (λ_m, k_m) using the following HF-based re-selection criterion:

$$(\bar{\lambda}_m, \bar{k}_m) \in \underset{(\lambda,k)\in\mathcal{P}^{\mathrm{HF}}_m\times\overline{\mathbb{K}}^{\mathrm{tr}}}{\operatorname{argmax}} \|\Gamma\big(\lambda, u^k_\lambda(\cdot)\big) - \bar{\gamma}^{m-1}_{m-1}(\lambda, k, \cdot)\|_{\ell^\infty(\Omega^{\mathrm{tr}})}. \tag{25}$$

We notice that this re-selection criterion only handles HF trajectories since the parameter values are in $\mathcal{P}^{\mathrm{HF}}_m$; moreover, (25) only requires to probe the values for λ_m, since the values for the other parameters, which are in $\mathcal{P}^{\mathrm{HF}}_{m-1}$, have already been evaluated in (24). The pair $(\bar{\lambda}_m, \bar{k}_m)$ is then used to define the point $\bar{x}_m$ and the function $\bar{q}_m$. Regarding the RB functions, a new POD selection is made if $\lambda_m \notin \mathcal{P}^{\mathrm{HF}}_{m-1}$, based on the enriched set of snapshots of the available HF trajectories:

$$(\theta^m_n)_{1\le n\le N^m} = \mathrm{POD}\big((\mathcal{S}_1, \cdots, \mathcal{S}_m), \epsilon_{\mathrm{POD}}\big), \tag{26}$$

where $\mathcal{S}_m = (u^k_{\lambda_m})_{k\in\overline{\mathbb{K}}^{\mathrm{tr}}}$, thereby producing the new set of RB functions to be used at the next iteration of PREIM. Otherwise, the previous set of RB functions is re-used.

Let us now discuss the initialization of PREIM. Due to the lack of knowledge on the solution manifold when starting PREIM, the initialization step requires a first HF computation. Choosing $\lambda_1 \in \mathcal{P}^{\mathrm{tr}}$ (typically in a random fashion or exploiting some a priori knowledge on the problem at hand), one computes a HF trajectory $\mathcal{S}_1 = (u^k_{\lambda_1})_{k\in\overline{\mathbb{K}}^{\mathrm{tr}}}$. Using the POD with a user-defined tolerance value ϵ_{POD} leads to an initial reduced basis of size N^1:

$$(\theta^1_n)_{1\le n\le N^1} = \mathrm{POD}(\mathcal{S}_1, \epsilon_{\mathrm{POD}}), \tag{27}$$

and a first RB trajectory $(\hat{u}^{1,k}_\lambda)_{k\in\overline{\mathbb{K}}^{\mathrm{tr}}} \in (\hat{X}_{N^1})^{K+1}$, with $\hat{X}_{N^1} = \mathrm{Span}\{\theta^1_1, \cdots, \theta^1_{N^1}\}$. One can then compute the EIM quantities

$$k_1 = \underset{k\in\overline{\mathbb{K}}^{\mathrm{tr}}}{\operatorname{argmax}} \|\Gamma\big(\lambda_1, u^k_{\lambda_1}(\cdot)\big)\|_{\ell^\infty(\Omega^{\mathrm{tr}})}, \quad r_1(\cdot) = \Gamma\big(\lambda_1, u^{k_1}_{\lambda_1}(\cdot)\big), \quad \bar{x}_1 = \underset{x\in\Omega^{\mathrm{tr}}}{\operatorname{argmax}} |r_1(x)|, \quad \bar{q}_1 = \frac{r_1}{r_1(\bar{x}_1)},$$

leading to the initial EIM approximation

$$\bar{\gamma}^1_1(\lambda, k, x) = \bar{\gamma}^1(\lambda, k, \bar{x}_1)\bar{q}_1(x). \tag{28}$$

It is also possible to initialize the RB space with $J \ge 1$ HF trajectories to improve accuracy in the intial stage of PREIM.

Remark 3 (line 20) *As the RB functions can change at every new iteration, the quantities in line 20 of Algorithm 4 need to be updated at each iteration since they are reduced matrices and vectors that depend on the RB functions.*

Remark 4 (PREIM-NR variant) *One can consider a variant of PREIM where one skips the re-selection step in line 12 of Algorithm 4. This variant, which we call PREIM-NR (for 'no re-selection'), will be tested numerically in the next section so as to highlight the actual benefits brought by this re-selection step.*

Algorithm 4 PREIM

INPUT : $\mathcal{P}^{\mathrm{tr}} \subset \mathcal{P}$, $\Omega^{\mathrm{tr}} \subset \Omega$, $\overline{\mathbb{K}}^{\mathrm{tr}} = \{0, \cdots, K\}$, $\epsilon_{\mathrm{POD}} > 0$, and $\epsilon_{\mathrm{EIM}} > 0$

1: Choose $\mathcal{P}_1^{\mathrm{HF}} = \{\lambda_1^*, \cdots, \lambda_J^*\} \subsetneq \mathcal{P}^{\mathrm{tr}}$

2: Compute $\mathcal{S}_1 = (u_{\lambda^*}^k)_{\lambda^* \in \mathcal{P}_1^{\mathrm{HF}}, k \in \overline{\mathbb{K}}^{\mathrm{tr}}}$ and $(\theta_n^1)_{1 \le n \le N^1} = \mathrm{POD}(\mathcal{S}_1, \epsilon_{\mathrm{POD}})$. $J \ge 1$ `HF trajectories`

3: 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 $\mathbf{A}_0 \in \mathbb{R}^{N^1 \times N^1}$

4: Compute $\bar{x}_1$, $\bar{q}_1$, $\mathbf{B} \in \mathbb{R}^{1,1}$, and $\mathbf{C}^1 \in \mathbb{R}^{N^1 \times N^1}$

5: Set $m = 2$

6: **do**

7: Compute $(\bar{u}_\lambda^{m-1,k})_{\lambda \in \mathcal{P}^{\mathrm{tr}}, k \in \overline{\mathbb{K}}^{\mathrm{tr}}}$

8: Search $(\lambda_m, k_m) \in \underset{(\lambda,k) \in \mathcal{P}^{\mathrm{tr}} \times \overline{\mathbb{K}}^{\mathrm{tr}}}{\mathrm{argmax}} \|\Gamma\big(\lambda, \bar{u}_\lambda^{m-1,k}(\cdot)\big) - \bar{\gamma}_{m-1}^{m-1}(\lambda, k, \cdot)\|_{\ell^\infty(\Omega^{\mathrm{tr}})}$ `based on RB/HF`

9: **if** $\lambda_m \notin \mathcal{P}_{m-1}^{\mathrm{HF}}$ **then**

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

11: Set $\mathcal{P}_m^{\mathrm{HF}} = \mathcal{P}_{m-1}^{\mathrm{HF}} \cup \{\lambda_m\}$

12: Search $(\bar{\lambda}_m, \bar{k}_m) \in \underset{(\lambda,k) \in \mathcal{P}_m^{\mathrm{HF}} \times \overline{\mathbb{K}}^{\mathrm{tr}}}{\mathrm{argmax}} \|\Gamma\big(\lambda, u_\lambda^k(\cdot)\big) - \bar{\gamma}_{m-1}^{m-1}(\lambda, k, \cdot)\|_{\ell^\infty(\Omega^{\mathrm{tr}})}$

13: **else**

14: Set $\mathcal{S}_m = \emptyset$

15: $(\bar{\lambda}_m, \bar{k}_m) = (\lambda_m, k_m)$

16: **end if**

17: Define $r_m(\cdot) := \Gamma\big(\bar{\lambda}_m, u_{\bar{\lambda}_m}^{\bar{k}_m}(\cdot)\big) - \bar{\gamma}_{m-1}^{m-1}(\bar{\lambda}_m, \bar{k}_m, \cdot)$

18: Define $\bar{x}_m$, $\bar{q}_m$, and $\mathbf{B}$ as in Algorithm 2 (line 6)

19: Compute $(\theta_n^m)_{1 \le n \le N^m} = \mathrm{POD}\big((\mathcal{S}_1, \cdots, \mathcal{S}_m), \epsilon_{\mathrm{POD}}\big)$

20: 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 $\mathbf{M}$, $\mathbf{A}_0$, $(\mathbf{C}^i)_{1 \le i \le m}$ in $\mathbb{R}^{N^m \times N^m}$

21: Set $m = m + 1$

22: **while** ($\|r_m\|_{\ell^\infty(\Omega^{\mathrm{tr}})} > \epsilon_{\mathrm{EIM}}$)

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

Remark 5 (U-SER variant) *In the next section, we also test numerically a second variant of PREIM where one replaces $\bar{u}_\lambda^{m,k}$ with $\hat{u}_\lambda^{m,k}$ in lines 7 and 8 of Algorithm 4, and one skips the re-selection step in line 12. We call this variant U-SER since it is a relatively straightforward 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 by numerical examples related to transient heat transfer with a nonlinearity either in the volume or on the boundary of the computational domain. Our goal is to illustrate the computational performance of PREIM and compare it to the standard EIM approach described in Section 4 and to the variants PREIM-NR and U-SER described in Remarks 5 and 4. We consider a bi-dimensional setting based on the perforated plate illustrated in Figure 1 with $\Omega = (-2, 2)^2 \backslash (-1, 1)^2 \subset \mathbb{R}^2$. HF trajectories are computed

using a Finite Element subspace $X \subset Y = H^1(\Omega)$ of dimension $\mathcal{N} = 1489$ and consisting of continuous, piecewise affine functions. The time discretization depends on the test case. The HF computations use the industrial software `code_aster` [3] and Python, whereas the reduced-order modeling algorithms have been developed in Python.

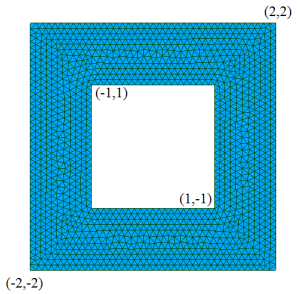

Figure 1 – Test cases (a) and (b): computational domain and mesh.

6.1 Test case a: Nonlinear thermal conductivity

We consider the nonlinear parabolic problem (1) with the nonlinear function $\Gamma(\lambda, z) := \sin\left(\frac{2\pi\lambda}{20}\left(\frac{z-u_0}{u_m-u_0}\right)^2\right)$, with $u_0 = 293K$ (20^oC) and $u_m = 323K$ (50^oC). We define $\kappa_0 = 1.05m^2.K^{-2}.s^{-1}$ and $\phi_e = 3K.m.s^{-1}$ (these units result from our normalization by the density times the heat capacity). Regarding the discretization scheme, we consider the time interval $I = [0, 5]$, the set of discrete times nodes $\mathbb{K}^{\text{tr}} = [\![1, 50]\!]$, and a constant time step $\Delta t^k = 0.1s$ for all $k \in \mathbb{K}^{\text{tr}}$. Finally, we consider the parameter interval $\mathcal{P} = [1, 20]$ and the training set $\mathcal{P}^{\text{tr}} = [\![1, 20]\!]$. In Figure 2, we show the temperature variation over the perforated plate at two different time nodes 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 λ than for smaller values. Also, for larger values of λ, the temperature variation tends to be less uniform over the plate than for small values of λ by the end of the simulation.

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 $\mathcal{N} = 1489$ nodal values. Applying the POD to $\mathcal{S}$ based on the H^1-norm ($\eta = 1$, see Appendix A) and a truncation threshold $\epsilon_{\text{POD}} = 10^{-3}$, we conserve $N = 15$ RB functions. The first six modes selected by the POD algorithm are shown in Figure 3. Additionally, Figure 4 (left panel) illustrates the rapid decrease of the singular values associated with the POD modes. Afterwards, we perform the standard EIM algorithm whose convergence rate is reported in Table 1. For $\epsilon_{\text{EIM}} = 5.10^{-3}$, the final rank of the EIM approximation is $M = 15$.

We now investigate PREIM and its variants PREIM-NR and U-SER. The first chosen parameter is identical for PREIM, PREM-NR and U-SER. Table 2 shows the selected parameters at each stage of PREIM and its variant PREM-NR. We can make several 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, whereas for the remaining 9 iterations, a different time snapshot of an already existing HF trajectory has been selected. The situation is similar with PREM-NR, whereby five HF trajectories are needed in the first 13 iterations. Another interesting observation is that PREM-NR actually selects the same couple (λ, k) twice (see Table 2, columns 4 and 8 for PREM-NR). This situation occurs because the

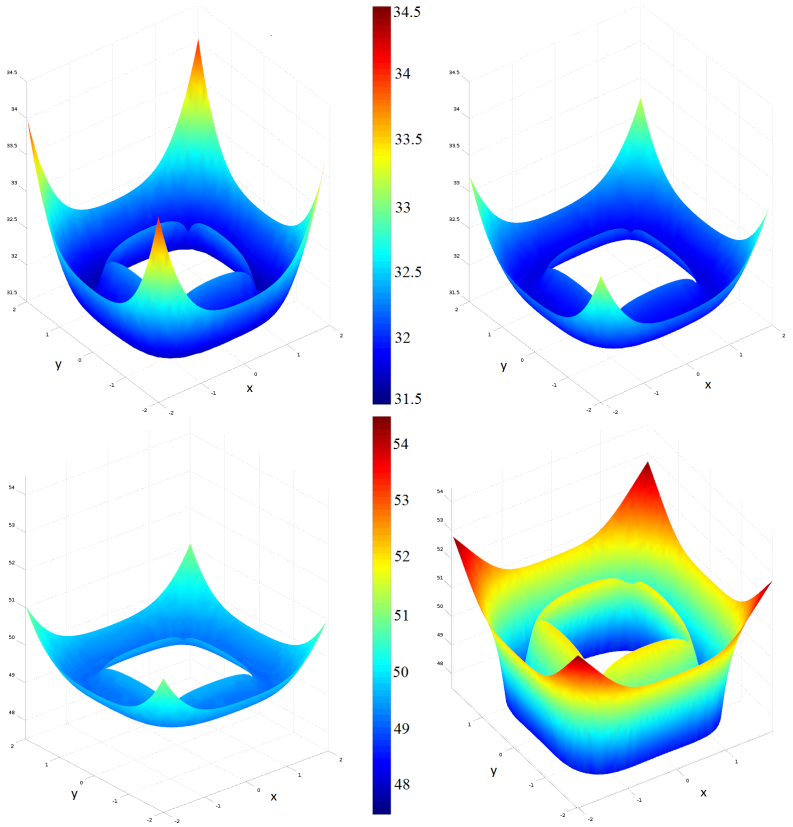

Figure 2 – Test case (a): HF solutions for the parameter values $\lambda = 1$ (left) and $\lambda = 18$ (right) at $t = 2s$ (top) and $t = 5s$ (bottom).

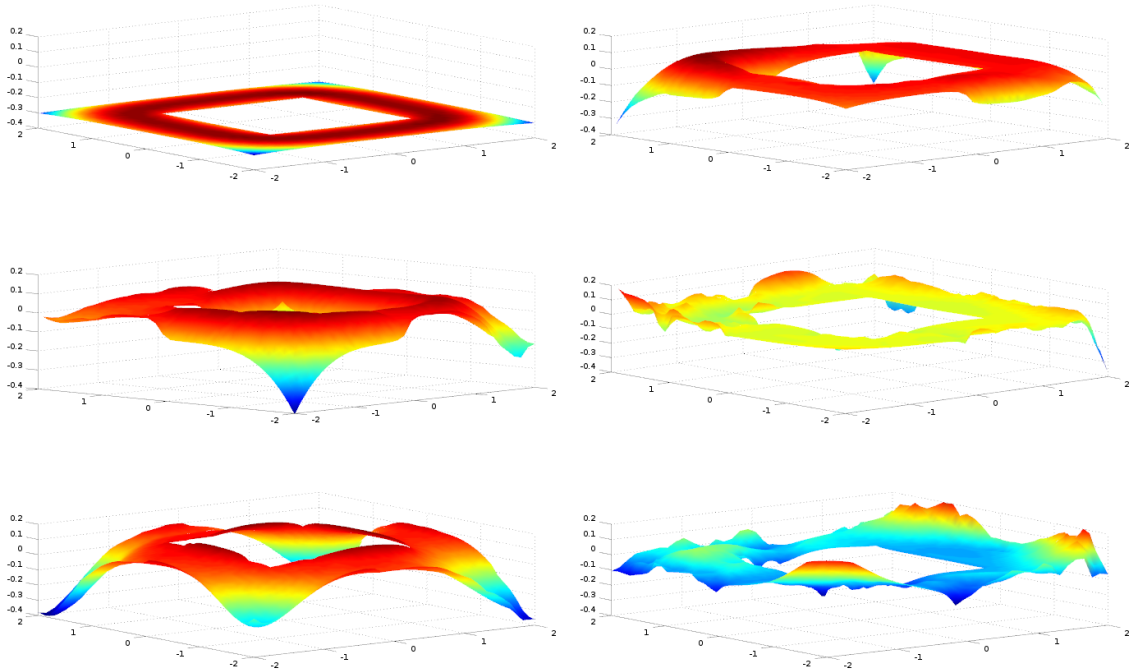

Figure 3 – Test case (a): Six first POD modes.

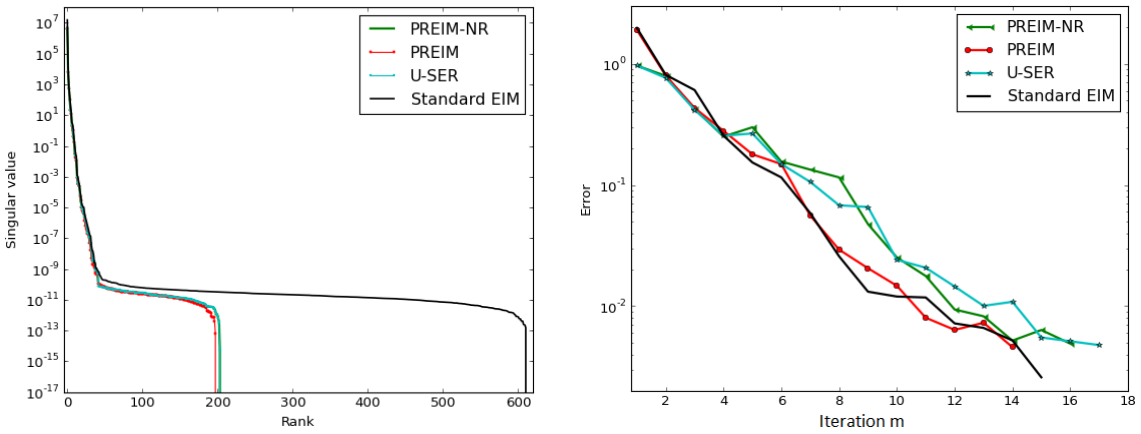

Figure 4 – Test case (a). Left: Singular values resulting from the POD; Right: EIM approximation error $\|r_m\|_{\ell^\infty(\Omega^{\mathrm{tr}} \times I^{\mathrm{tr}} \times \mathcal{P}^{\mathrm{tr}})}$ as a function of m.

m	$\|r_m\|_{\ell^\infty(\Omega^{\mathrm{tr}})}$
1	1.97
2	8.11E−01
6	1.12E−01
14	5.22E−03
15	2.59E−03
20	1.08E−03
25	1.56E−04

Table 1 – Test case (a): Evolution of the standard EIM error. m is the rank of the EIM approximation and $\|r_m\|_{\ell^\infty(\Omega^{\mathrm{tr}})}$ is the residual norm in (18).

m		1	2	3	4	5	6	7	8	9	10	11	12	13
PREIM	λ	1	20	20	20	20	20	20	17	16	20	20	20	20
	k	50	45	48	50	43	42	39	50	50	49	33	47	44
PREIM-NR	λ	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 2 – Test case (a): Selected parameter and time node values in PREIM and PREM-NR. The gray cells correspond to a new parameter selection and, therefore, to a new HF computation.

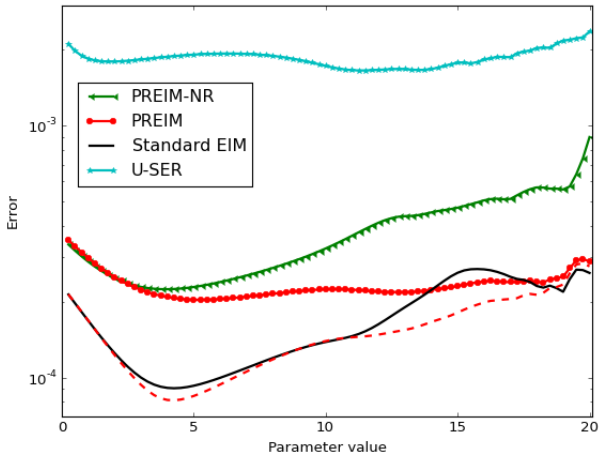

Figure 5 – Test case (a): RB approximation error $\|u_\lambda - \hat{u}_\lambda\|_{\ell^1(I^{\mathrm{tr}};\ell^2(\Omega^{\mathrm{tr}}))}$.

greedy selection in PREM-NR is not based on HF trajectories but on RB trajectories, and the latter vary with the iteration number since the approximation of the nonlinearity is improved. Therefore, there is nothing preventing a pair (λ, k) from being selected twice in PREM-NR; this is not possible in PREIM since the selection is based on the HF trajectories which are independent from the iteration index. Incidentally, we point out that a similar situation may also happen in U-SER (and in the standard stationary SER). In Figure 4 (left panel), we present the singular values associated with the POD modes at the last iteration for PREIM, PREIM-NR and U-SER; we observe that all the significant singular values coincide with those detected in the standard offline stage. Figure 4 (right panel) displays the error on the approximation of the nonlinear function Γ for PREIM, PREM-NR and U-SER as a function of the iteration number m (i.e., $\|\Gamma(\lambda, u_\lambda^k(\cdot)) - \overline{\gamma}_m^m(\lambda, k, \cdot)\|_{\ell^\infty(\Omega^{\mathrm{tr}}\times\overline{\mathbb{K}}^{\mathrm{tr}}\times\mathcal{P}_m^{\mathrm{HF}})}$, $\|\Gamma(\lambda, \bar{u}_\lambda^k(\cdot)) - \overline{\gamma}_m^m(\lambda, k, \cdot)\|_{\ell^\infty(\Omega^{\mathrm{tr}}\times\overline{\mathbb{K}}^{\mathrm{tr}}\times\mathcal{P}^{\mathrm{tr}})}$, and $\|\Gamma(\lambda, \hat{u}_\lambda^k(\cdot)) - \overline{\gamma}_m^m(\lambda, k, \cdot)\|_{\ell^\infty(\Omega^{\mathrm{tr}}\times\overline{\mathbb{K}}^{\mathrm{tr}}\times\mathcal{P}^{\mathrm{tr}})}$ respectively) and compares these errors to those obtained with the standard approach. We observe that the PREIM approximation of Γ is practically as good as that of the standard EIM overall the whole iteration range. However, PREIM achieves this by using much less HF trajectories. For an error tolerance $\epsilon_{\mathrm{EIM}} = 5.10^{-3}$, PREIM only requires four HF computations (and PREM-NR only requires five); this is five times (four for PREM-NR) less HF trajectories than the 20 HF trajectories computed in the standard EIM. Finally, Figure 5 compares (using the ℓ^1-norm in time and the Euclidean ℓ^2-norm in space) the space-time errors on the actual trajectories for the whole parameter range, which we denote $\|u_\lambda - \hat{u}_\lambda\|_{\ell^1(I^{\mathrm{tr}};\ell^2(\Omega^{\mathrm{tr}}))}$. Considering all the possible parameter values, the PREIM error is of the order of 5.10^{-4} at most. Moreover, for the same value of ϵ_{EIM}, PREIM achieves an accuracy that is even better than that of the standard EIM for values of the parameter in $[14, 18]$ (compare the red and the black curves). We also observe that PREIM approximates the RB solution better than PREIM-NR, with the most significant improvements observed for the larger values of the parameter (compare the red and the green curves). In a worst-case scenario with respect to all parameter values, PREIM achieves a similar error to that of the standard EIM, which achieves a smaller error than PREIM-NR. Finally, we combine the EIM approximation produced by PREIM with the RB space produced by the standard nonlinear RB method, and we display the corresponding error curve (referred to as 'PREIM+RB' in Figure 5). We observe that this combination produces an error curve that behaves similarly to that of the standard EIM with $M = 25$. Hence, this indicates that the dominant error source in PREIM is due to the RB

space rather than to the approximation of the nonlinearity.

6.2 Test case b: Nonlinear reaction coefficient

Here, the nonlinear behavior of the problem arises in a reaction coefficient over Ω. We consider the following nonlinear parabolic PDE: For many values of the parameter $\lambda \in \mathcal{P}$, find $u_\lambda : I \times \Omega \to \mathbb{R}$ such that

$$\begin{cases} \dfrac{\partial u_\lambda}{\partial t} - \nabla \cdot (\kappa_0(1+\lambda)\nabla u_\lambda) + \Upsilon(\lambda, u_\lambda) = f, & \text{in } I \times \Omega, \\ -\kappa_0 \dfrac{\partial u_\lambda}{\partial n} = \phi_e, & \text{on } I \times \partial\Omega, \\ u_\lambda(t=0,\cdot) = u_0(\cdot) & \text{in } \Omega, \end{cases} \tag{29}$$

where $\kappa_0 > 0$, $u_0 \in H^1(\Omega)$ is the initial condition, $f \in L^2(I; L^2(\Omega))$ is the given source term, and $\Upsilon : \mathcal{P} \times \mathbb{R} \to \mathbb{R}$ accounts for the nonlinear reaction term in Ω. For the sake of numerical illustration, we consider the function $\Upsilon(\lambda, z) = \exp(-\lambda \frac{z}{u_0})z$.

We consider again a Y-conforming Finite Element Method and a time-marching scheme, leading to HF trajectories that we still denote by u_λ. Using as above a time semi-implicit scheme for simplicity, our goal is, given $u_\lambda^0 = u_0 \in X$, to find $(u_\lambda^k)_{k \in \mathbb{K}^{\mathrm{tr}}} \in X^K$ such that, for all $k \in \mathbb{K}^{\mathrm{tr}}$,

$$\forall v \in X, \quad m(u_\lambda^k, v) + \Delta t^k a_0\big(u_\lambda^k, v\big) + \Delta t^k n_\Upsilon\big(\lambda, u_\lambda^{k-1}, v\big) = m(u_\lambda^{k-1}, v) + \Delta t^k g^k(v), \tag{30}$$

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

$$n_\Upsilon(\lambda, v, w) = \int_\Omega \Upsilon(\lambda, v) w, \tag{31}$$

and the linear forms $g^k : Y \to \mathbb{R}$ such that $g^k(v) = \int_\Omega f^k v$.

Similarly to Section 3, we introduce the nonlinear function $\upsilon : \mathcal{P} \times \overline{\mathbb{K}}^{\mathrm{tr}} \times \Omega \to \mathbb{R}$ such that

$$\upsilon(\lambda, k, x) := \Upsilon(\lambda, u_\lambda^k(x)), \tag{32}$$

and we use the EIM to compute a rank-M approximation of this function in the form

$$\upsilon_M(\lambda, k, x) := \sum_{m=1}^{M} \varphi_{\lambda,m}^k q_m(x), \tag{33}$$

where q_m are real-valued functions defined on Ω and $\varphi_{\lambda,m}^k$ are real numbers such that

$$\forall i \in \{1, \dots, M\}, \quad \sum_{m=1}^{M} \mathbf{B}_{im} \varphi_{\lambda,m}^k = \upsilon(\lambda, k, x_i), \tag{34}$$

where $\mathbf{B} \in \mathbb{R}^{M \times M}$ is a lower-triangular invertible matrix and $(x_i)_{1 \le i \le M}$ are points in the subset Ω^{tr}, which is typically the collection of the nodes of the finite element mesh lying at the boundary. 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 $(\theta_n)_{1 \le n \le N}$ be a Y-orthonormal basis of $\hat{X}_N$. Then, the matrix formulation

m	$\|r_m\|_{\ell^\infty(\Omega^{\mathrm{tr}})}$
1	14.81
2	2.69E−01
7	8.54E−02
10	9.63E−03
20	1.93E−04

Table 3 – Test case (b): Evolution of the EIM error. m is the rank of the EIM approximation and $\|r_m\|_{\ell^\infty(\Omega^{\mathrm{tr}})}$ is the residual norm in (18).

for computing a RB trajectory reads as follows: Given $\hat{\mathbf{u}}_\lambda^0 = \hat{\mathbf{u}}^0 \in \hat{X}_N$, find $(\hat{\mathbf{u}}_\lambda^k)_{k\in\mathbb{K}^{\mathrm{tr}}} \in (\mathbb{R}^N)^K$ such that, for all $\mathbb{K}^{\mathrm{tr}}$,

$$(\mathbf{M} + \Delta t^k \mathbf{A}_0)\hat{\mathbf{u}}_\lambda^k = \mathbf{M}\hat{\mathbf{u}}_\lambda^{k-1} + \Delta t^k(\mathbf{g}^k - \mathbf{D}\boldsymbol{\upsilon}_\lambda^{k-1}), \tag{35}$$

with the rectangular matrices $\mathbf{D} = \mathbf{C}(\mathbf{B})^{-1} \in \mathbb{R}^{N\times M}$ and $\mathbf{C} \in \mathbb{R}^{N\times M}$, the vector $\boldsymbol{\upsilon}_\lambda^{k-1} \in \mathbb{R}^M$ such that

$$\mathbf{C} = \left(\int_\Omega q_m \theta_p\right)_{1\le p\le N, 1\le m\le M}, \qquad \boldsymbol{\upsilon}_\lambda^{k-1} = \left(\Upsilon(\lambda, \hat{u}_\lambda^{k-1}(x_m))\right)_{1\le m\le M}, \tag{36}$$

and the vector $\mathbf{g}^k \in \mathbb{R}^N$ such that $\mathbf{g}^k = \left(g^k(\theta_p)\right)_{1\le p\le N}$. Therefore, there is only one difference with the computational procedure outlined in Section 3, namely the collection of matrices $(\mathbf{C}^m)_{1\le m\le M}$ in $\mathbb{R}^{N\times N}$ is replaced with a single matrix $\mathbf{C} \in \mathbb{R}^{N\times M}$. One advantage is that it is no longer necessary to perform reduced matrix computations during the online stage.

Remark 6 (Alternative choice) *It is also possible to define* $\Upsilon'(\lambda, z) := \exp(-\lambda\frac{z}{u_0})$*, leading to the nonlinear form* $n_\Upsilon(\lambda, v, w) = \int_\Omega \Upsilon'(\lambda, v)vw$*. Then, one recovers a similar formulation to that of Section 3 requiring some additional computations during the online stage. One motivation for this choice* (36) *is that the structure of the reaction term with a scaling proportional to the solution is somehow better preserved.*

We set $\kappa_0 = 0.01m^2.K^{-2}s^{-1}$, $u_0 = 293K$ (20^oC), $\eta = 0.5$, and we introduce the time interval $I = [0, 1]$, a set of discrete times nodes $\mathbb{K}^{\mathrm{tr}} = [\![1, 20]\!]$ and a uniform time step $\Delta t = 0.2s$ (so that $K = 5$). Finally, we consider a parameter interval $\mathcal{P} = [0, 10]$ and a training set $\mathcal{P}^{\mathrm{tr}} = \{0.5i \mid 0 \le i \le 20\}$, and we take $\epsilon_{\mathrm{POD}} = 10^{-1}$. In Figure 6, we show the temperature variation over the perforated plate at two different time nodes for two different parameter values. The temperature increases as the simulation evolves in time. Also, for larger values of the parameter λ, the overall temperature increases faster than for smaller values. We perform $P = 21$ HF computations in the standard offline stage. Since $K = 5$, the set $\mathcal{S}$ (Algorithm 2, line 1) contains 105 snapshots. The convergence of the standard EIM approximation is illustrated in Table 3. Regarding PREIM, Table 4 shows the first selected parameters. We notice that only seven HF trajectories are computed in PREIM which is four times less than the number of HF trajectories computed in the standard EIM. The approximation error on the reaction term for the the standard offline stage, PREIM, and its variants PREIM-NR and U-SER are compared in Figure 7. We can draw similar conclusions to those from the first test case. Finally, Figure 8 reports the space-time errors on the actual trajectories for the

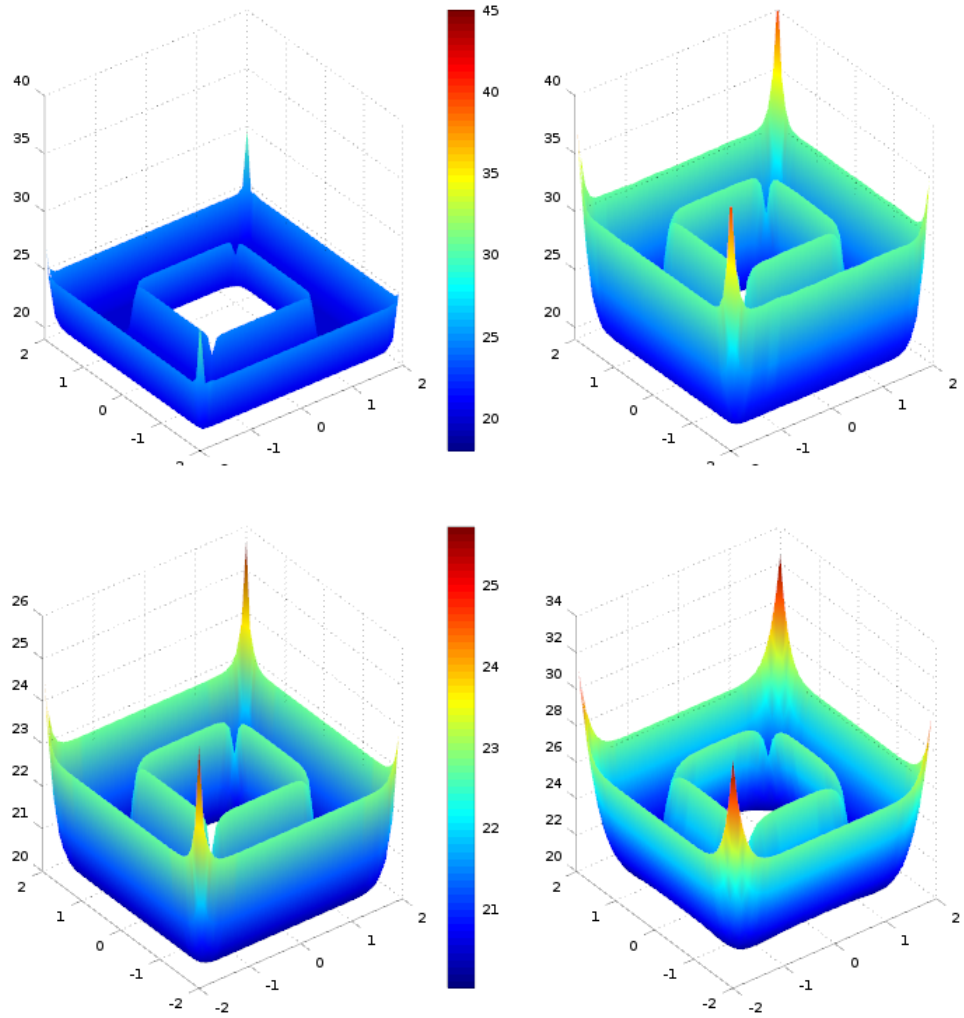

Figure 6 – Test case (b): Numerical HF solutions for the parameter values $\lambda = 0.5$ (top) and $\lambda = 9.5$ (bottom) at $t = 0.2s$ (left) and $t = 1s$ (right).

m		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
PREIM	λ	0	0	1	2	4	1	2	0	2	7	1	0	–	–	–	–
	k	4	1	4	0	4	4	4	2	3	4	1	3	–	–	–	–
PREIM-NR	λ	0	0	0	1	1	3	5	2	0	1	4	2	5	1	2	0
	k	4	0	1	4	2	4	4	4	2	4	4	4	4	1	3	3

Table 4 – Test case (b): First selected parameter values in PREIM. The gray cells correspond to a new parameter selection and, therefore, to a new HF computation.

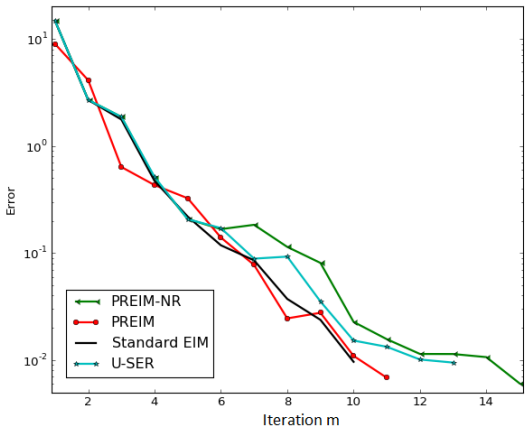

Figure 7 – Test case (b). EIM approximation error $\|r_m\|_{\ell^\infty(\Omega^{\mathrm{tr}}\times I^{\mathrm{tr}}\times\mathcal{P}^{\mathrm{tr}})}$ as a function of m.

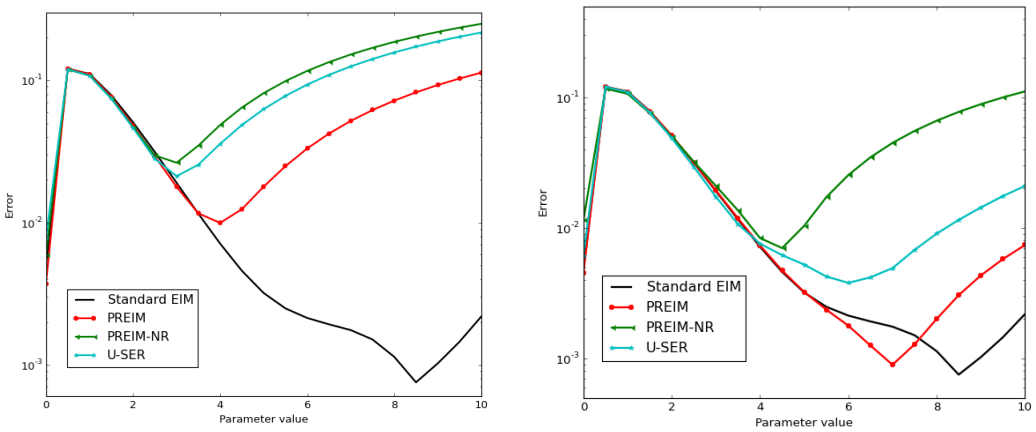

Figure 8 – Test case (b). RB approximation error $\|u_\lambda - \hat{u}_\lambda\|_{\ell^1(I^{\mathrm{tr}};\ell^2(\Omega^{\mathrm{tr}}))}$ with $\epsilon_{\mathrm{POD}} = 10^{-1}$. Left: $\epsilon_{\mathrm{EIM}} = 10^{-2}$; Right: $\epsilon_{\mathrm{EIM}} = 10^{-5}$.

whole parameter range for $\epsilon_{\mathrm{EIM}} = 10^{-2}$ (left panel) and $\epsilon_{\mathrm{EIM}} = 10^{-5}$ (right panel). We observe that tightening the tolerance on the approximation of the nonlinearity allows the different algorithms to produce accurate solutions for a larger set of the parameter values. Moreover, as before, PREIM performs a reduced number of HF computations compared to the standard EIM while obtaining a similar overall accuracy.

7 Conclusion and perspectives

In this work, we have devised a new methodology, called PREIM, that diminishes the offline expenses incurred in the nonlinear RB method applied to unsteady nonlinear PDEs. Numerical tests on two-dimensional nonlinear heat transfer problems with either nonlinear conductivity or nonlinear Robin boundary condition have shown the computational efficiency and the accuracy of the algorithm. In addition, the application of PREIM to an industrial test case of a three-dimensional flow-regulation valve is ongoing. One possible perspective to improve further the accuracy of the final RB approximation resulting from PREIM is to consider a RB error threshold in addition to the existing EIM threshold. For instance, at each iteration of PREIM,

the RB error can be approximated by its value on the set of the HF trajectories that are already available.

Acknowledgments

This work is 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`.

A Proper Orthogonal Decomposition

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

$$(\theta_1, \dots, \theta_N) = \mathrm{POD}(\mathcal{S}, \epsilon_{\mathrm{POD}}), \tag{37}$$

which is used in Algorithm 3 and in Algorithm 4, where $\mathcal{S} = (v_1, \cdots, v_R)$ is composed of $R \geq 1$ elements in X and ϵ_{POD} is a user-prescribed tolerance. For simplicity, we adopt an algebraic description. Let $(\varrho_1, \dots, \varrho_{\mathcal{N}})$ be a basis of X where $\dim(X) = \mathcal{N}$; then, 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 $w = \sum_{j=1}^{\mathcal{N}} w_j \varrho_j$. Thus, the algebraic counterpart of (37) 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 the N vectors forming the rectangular matrix $\mathbf{\Theta} := (\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_N) \in \mathbb{R}^{\mathcal{N} \times N}$. The vectors $\boldsymbol{\theta}_n$ are to be orthonormal with respect to the Gram matrix of an inner product in X. We consider the Gram matrix $\mathbf{C}^{\mathcal{N}} \in \mathbb{R}^{\mathcal{N} \times \mathcal{N}}$ such that

$$\mathbf{C}^{\mathcal{N}} = \Big(m(\varrho_n, \varrho_p) + \eta a_0(\varrho_n, \varrho_p) \Big)_{1 \leq p,n \leq \mathcal{N}}, \tag{38}$$

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 general, we have $D = R$). Computing the Singular Value Decomposition [13] 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

$$\mathbf{T} = \sum_{n=1}^{D} \sigma_n \boldsymbol{\xi}_n \hat{\boldsymbol{\psi}}_n^{\mathrm{T}}. \tag{39}$$

From (39), 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 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 $\sum_{r=1}^{R} \inf_{\mathbf{z} \in \mathbf{Z}_N} (\mathbf{v}_r - \mathbf{z})^{\mathrm{T}} \mathbf{C}^{\mathcal{N}} (\mathbf{v}_r - \mathbf{z})$ among all the N-dimensional subspaces $\mathbf{Z}_N$ of $\mathbb{R}^{\mathcal{N}}$. Moreover, we have $\|u_\lambda^k - \Pi_{Z_N} u_\lambda^k\|_X \leq \sigma_{N+1} \|u_\lambda^k\|_X$, for all $\lambda \in \mathcal{P}^{\mathrm{tr}}$ and for all $k \in \overline{\mathbb{K}}^{\mathrm{tr}}$.

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 follows:

$$\boldsymbol{\theta}_n = \left(\mathbf{C}^{\mathcal{N}}\right)^{-\frac{1}{2}} \boldsymbol{\xi}_n = \frac{1}{\sigma_n}\left(\mathbf{C}^{\mathcal{N}}\right)^{-\frac{1}{2}} \mathbf{T}\hat{\boldsymbol{\psi}}_n = \frac{1}{\sigma_n}\mathbf{S}\hat{\boldsymbol{\psi}}_n. \tag{40}$$

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(9):667–672, 2004.

[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(12):1105–1109, 2015.

[3] 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.

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

[5] M. A. Grepl. Certified reduced basis methods for nonaffine linear time-varying and nonlinear parabolic partial differential equations. Math. Models Methods Appl. Sci., 22(3):1150015, 40, 2012.

[6] 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(3):575–605, 2007.

[7] B. Haasdonk. Convergence rates of the POD-greedy method. ESAIM Math. Model. Numer. Anal., 47(3):859–873, 2013.

[8] 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.

[9] 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, volume 45 of Lect. Notes Comput. Sci. Eng., pages 261–306. Springer, Berlin, 2005.

[10] K. Kunisch and S. Volkwein. Galerkin proper orthogonal decomposition methods for parabolic problems. Numer. Math., 90(1):117–148, 2001.

[11] Luc Machiels, Yvon Maday, Ivan B. Oliveira, Anthony T. Patera, and Dimitrios V. Rovas. Output bounds for reduced-basis approximations of symmetric positive definite eigenvalue problems. C. R. Acad. Sci. Paris Sér. I Math., 331(2):153–158, 2000.

[12] 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(1):383–404, 2009.

[13] B. Noble and J. W. Daniel. Applied Linear Algebra. Prentice-Hall, 3rd edition, 1988.

[14] Christophe Prud'Homme, Dimitrios V. Rovas, Karen Veroy, Luc Machiels, Yvon Maday, Anthony T. Patera, and Gabriel Turinici. Reliable Real-Time Solution of Parametrized Partial Differential Equations: Reduced-Basis Output Bound Methods. Journal of Fluids Engineering, 124(1):70–80, November 2001.

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

[16] D. V. Rovas, L. Machiels, and Y. Maday. Reduced-basis output bound methods for parabolic problems. IMA J. Numer. Anal., 26(3):423–445, 2006.