A reduced order model for the finite element approximation of eigenvalue problems

Fleurianne Bertrand\textsuperscript{a,}\textsuperscript{*}, Daniele Boffi\textsuperscript{b,c}, Abdul Halim\textsuperscript{a,d}

\textsuperscript{a} Department of Applied Mathematics, University of Twente, Netherlands
\textsuperscript{b} King Abdullah University of Science and Technology, Saudi Arabia
\textsuperscript{c} Dipartimento di Matematica “F. Casorati”, University of Pavia, Italy
\textsuperscript{d} Department of Mathematics, Hari Singh College, Munger University, India

Received 22 March 2022; received in revised form 1 August 2022; accepted 1 October 2022
Available online 28 November 2022

Abstract

In this paper we consider a reduced order method for the approximation of the eigensolutions of the Laplace problem with Dirichlet boundary condition. We use a time continuation technique that consists in the introduction of a fictitious time parameter. We use a POD approach and we present some theoretical results showing how to choose the optimal dimension of the POD basis. The results of our computations confirm the optimal behavior of our approximate solution. We compute the first eigenvalue and discuss how to approximate the next eigenmodes.

© 2022 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Keywords: Eigenvalue problem; FEM; POD; Reduced order modeling

1. Introduction

The mathematical analysis of FEM approximation to PDE eigenvalue problems is quite mature and its evolution has reached a deep development in most application areas. The a priori analysis of standard Galerkin approximations, after the pioneer works by Fix and Strang [1] is commonly referred to as the Babuška–Osborn theory [2]. In the case of standard Galerkin approximations, it is well understood that a finite element scheme can be used successfully for the discretization of compact eigenvalue problems, whenever it works well for the discretization of the corresponding source problem. This is not the case for other formulations, such as those arising from mixed methods, where the inf–sup conditions are neither sufficient nor necessary for the design of a spectrally correct approximation [3–6].

Reduced order models (ROM) are part of a consolidated technique that is commonly related to the pioneer works of [7–10]. It underwent then a successful development in the framework of PDEs from the late 90’s where different approaches have been explored [11–28]. A commonly used approach is the Proper Orthogonal Decomposition (POD) method, which is most often used in the one dimensional temporal discretization as opposed to the Reduced Basis (RB) approximation which is generally applied to multi-dimensional parameter domains.

Eigenvalue problems arising from partial differential equations are of great interest in multiple scientific and engineering case studies such as the modal analysis of structures in solid mechanics. Moreover, the stability of
Upper applications include fluid–structure interactions as in [30] or electromagnetic wave propagation [31]. Parametric eigenvalues occurs in many areas of science and engineering. The efficient computation of eigenvalues and eigenvectors is crucial for several problems, ranging from fluid dynamics, structural mechanics, electromagnetism and possibly involving various industrial and medical applications, such as cardiovascular surgery or aeronautical engineering. Other emerging fields where accurate computation of eigenmodes are crucial are related to nuclear reactor physics [32] and photonics [33]. The energetic diffusion of the neutron is represented as a parametric eigenvalue problem in nuclear physics, where the first eigenvalue represents the criticality of the reactor and the corresponding eigenvalue represent neutron flux.

Reduced order model for eigenvalue problems have been the object of a more limited but not empty investigation, starting from the pioneer works of [15,16,34,35], where only the first fundamental mode was considered. Pau in his PhD thesis [36] has the merit to start the investigation of higher-order modes, even if still in the setting of simple eigenvalues [37,38]. In [39] an a posteriori analysis for the first eigenvalue is presented and examples of applications to parametric problems are provided. Finally, [40] attacks the most complex situation of multiple eigenvalues by approximating at once a fixed number of eigenmodes, starting from the first one.

We start our investigation from the approach presented in [32] where the idea of adding a fictitious time variable for the solution of eigenvalue problems is introduced, see also [41]. We consider this research as a proof of concepts towards more general situations. In order to do so, let us compare this approach with some more conventional ones. In [34,35] a projection based method was used to solve the eigenvalue problem (EVP). In order to do so, we need to solve the high-fidelity problem several times for a reasonably large amount of parameters. On the other hand, with the introduction of a time continuation fictitious variable, as in [32] and our research, we can get more snapshots naturally, without employing too many parameters and high-fidelity solution of EVP. In turn, we can get a reasonable ROM formulation of parametric EVP with fewer samples of parameters. Since we are interested in understanding the mathematical properties and the numerical features of our approach, we start by considering a non-parametric EVP and only later we apply our technique to a parametric EVP. We have to point out some crucial differences with respect to the method of [32]. In particular, we used a different formula to calculate the eigenvalues, taking into account the Rayleigh quotient; this allows us to consider also the next eigenvalues while the formula presented in [32] will not work, for instance, in the case of the second eigenvalue of the Laplace operator. Moreover our work is investigative in nature for different domains and different meshes.

In this paper we investigate the use of reduced order models for the approximation of eigenvalue problem. We consider the model eigenvalue problem associated with the Laplace equation with Dirichlet boundary conditions and we introduce a fictitious time parameter in order to employ a time continuation technique. In this framework, we adopt a POD technique in order to build a ROM after the computation of some snapshots of the high fidelity solution. The problem we are considering can be thought as a proof of concept towards more general problems. We are dealing mainly with non parametric EVP and showing how our method can be applied to parameter dependent EVP. Moreover, we start our computations with the approximation of the first fundamental mode and discuss how this can be generalized to the approximation of other eigenvalues.

Section 2 introduces the problem and presents the time continuation technique approach. Section 3 describes the reduced order model framework that we will adopt, the construction of the POD basis, and the theoretical results about the choice of the optimal dimension for the POD space. Finally, Section 4 reports the results of several numerical tests, showing the good performance of the method.

2. Model problem and time continuation approach

Let us consider the eigenvalue problem related to the Laplace equation with Dirichlet boundary condition: find eigenvalues $\lambda$ and non vanishing eigenfunctions $u$ that solve the following problem

$$
\begin{align*}
- \Delta u(x) &= \lambda u(x) \quad \text{in } \Omega \\
 u(x) &= 0 \quad \text{on } \partial \Omega.
\end{align*}
$$

(1)

Our approach considers a reduced order model where a fictitious time parameter $t$ is introduced. In the spirit of [32] a time continuation approach, the problem reduces to finding the steady state solution to the following equation

$$
\frac{\partial u(x, t)}{\partial t} - \Delta u(x, t) = \lambda(t, u)u(x, t).
$$

(2)
This equation must be complemented by a relation linking \( u(x, t) \) and \( \lambda(t, u) \). In the steady state limit \( \lambda \) can be computed by the Rayleigh quotient derived as usual from the stationary equation

\[
-\Delta u(x) = \lambda u(x).
\]

Multiplying both sides by \( u(x) \), integrating over \( \Omega \), and using Green’s formula we get

\[
\lambda = \frac{\int_\Omega \nabla u(x) \cdot \nabla u(x) \, dx}{\int_\Omega u^2(x) \, dx} = \frac{\int_\Omega |\nabla u(x)|^2 \, dx}{\int_\Omega u^2(x) \, dx}. \tag{3}
\]

It should be noted that in [32] a different formula is adopted for calculating the eigenvalue. This formula involves a denominator which contains the term

\[
\int_\Omega u(x) \, dx.
\]

Clearly, this is not applicable to eigenfunctions with zero mean value, such as the second one in the case of the Laplace operator on a square or circular domain. On the contrary, our formula allows us to compute all eigenfunctions we are interested in.

Hence, the eigenvalues and eigenfunctions of (1) are given by the steady state solutions of the following coupled problem: given an initial guess \( u^0 : \Omega \to \mathbb{R} \), for all \( t > 0 \) find \( u(x, t) : \Omega \to \mathbb{R} \) and \( \lambda(t, u) \in \mathbb{R} \) such that

\[
\begin{aligned}
\lambda(t, u) &= \frac{\int_\Omega |\nabla u(x, t)|^2 \, dx}{\int_\Omega u^2(x, t) \, dx} \\
\frac{\partial u(x, t)}{\partial t} - \Delta u(x, t) &= \lambda(t, u) u(x, t) \quad \text{in } \Omega \\
u(x, 0) &= u^0(x) \quad \text{in } \Omega \\
u(x, t) &= 0 \quad \text{on } \partial \Omega, \ t > 0.
\end{aligned} \tag{4}
\]

We solve Problem (4) with finite elements in space and finite difference in time. The time derivative is discretized by Euler method, with time step \( \Delta t \), as \( u_t = \frac{u^{k+1} - u^k}{\Delta t} \), so that we get the following time semi-discretized model

\[
\begin{aligned}
\lambda^k &= \frac{\int_\Omega |\nabla u^k|^2 \, dx}{\int_\Omega |u^k|^2 \, dx} \\
\frac{1}{\Delta t} u^{k+1}(x) - \Delta u^{k+1}(x) &= \lambda^k u^k(x) + \frac{1}{\Delta t} u^k(x),
\end{aligned} \tag{5}
\]

where \( \lambda^k \) is calculated from the Rayleigh quotient (3) using the solution \( u^k \) at the previous time step.

The weak formulation of the second equation in (5) is obtained as usual after multiplication by a test function \( v \in V := H^1_0(\Omega) \) and integration over \( \Omega \)

\[
\frac{1}{\Delta t} \int_\Omega u^{k+1} v \, dx - \int_\Omega \Delta u^{k+1} v \, dx = \lambda^k \int_\Omega u^k v \, dx + \frac{1}{\Delta t} \int_\Omega u^k v \, dx
\]

Applying Green’s theorem in the second term we get

\[
\frac{1}{\Delta t} \int_\Omega u^{k+1} v \, dx + \int_\Omega \nabla u^{k+1} \cdot \nabla v \, dx = \lambda^k \int_\Omega u^k v \, dx + \frac{1}{\Delta t} \int_\Omega u^k v \, dx
\]

where we have used the fact that \( v = 0 \) on the boundary.

Thus the time discretized problem reads as: for given \( u^0 \in V \) and \( \Delta t > 0 \), for each \( k = 0, 1, \ldots \), find \( u^{k+1} \in V \) such that

\[
\begin{aligned}
\frac{1}{\Delta t} (u^{k+1}, v) + (\nabla u^{k+1}, \nabla v) &= \lambda^k (u^k, v) + \frac{1}{\Delta t} (u^k, v) \quad \forall v \in V
\end{aligned} \tag{6}
\]
2.1. Convergence analysis for the time continuation problem

The aim of this subsection is to show that the solution of (6) converges as $k \to \infty$ to the fundamental mode of the original problem (1), that is $u^k$ tends to the first eigenfunction and $\lambda^k$ to the first eigenvalue. Similar analysis will be applied also to the space discretized problem and to the reduced model.

We denote by $(\lambda_i, u_i) \in \mathbb{R} \times V$, $i = 1, 2, \ldots$, the eigenpairs of the Laplace operator with orthonormalized eigenfunctions, that is

$$\begin{align*}
(\nabla u_i, \nabla v) &= \lambda_i(u_i, v) \quad \forall v \in V \\
(u_i, u_j) &= \delta_{ij} \\
\lambda_1 &< \lambda_2 \leq \lambda_3 \leq \cdots
\end{align*}$$

We start by discussing some examples with increasing level of difficulty that will lead to the final result stated in Proposition 1.

Example 1. Let us take $u^0 = u_1$ so that $\lambda^0 = \lambda_1$, then it is clear that the solution of (6) is constant in $k$ with $u^k = u_1$ and $\lambda^k = \lambda_1$. Indeed, from (6) with $k = 0$ we have

$$\frac{1}{\Delta t}(u^1, v) + (\nabla u^1, \nabla v) = \lambda_i(u_i, v) + \frac{1}{\Delta t}(u_i, v)$$

which is satisfied by $u^k = u_1$ thanks to ($\nabla u_i, \nabla v = \lambda_i(u_i, v)$ valid for all $v \in V$.

Example 2. We now take $u^0 = \alpha_0 u_1 + \beta_0 u_2$ and we normalize it so that $\alpha_0^2 + \beta_0^2 = 1$. It follows that

$$\begin{align*}
\lambda^0 &= \alpha_0^2 \lambda_1 + \beta_0^2 \lambda_2 \\
&\in [\lambda_1, \lambda_2]
\end{align*}$$

Let us perform one step of the time continuation approach (6)

$$\frac{1}{\Delta t}(u^1, v) + (\nabla u^1, \nabla v) = \lambda^0(u^0, v) + \frac{1}{\Delta t}(u^0, v)$$

By using the relation ($\nabla u_i, \nabla v = \lambda_i(u_i, v)$ $(i = 1, 2$) it can be seen that the solution $u^1$ is given by

$$u^1 = \alpha_1 u_1 + \beta_1 u_2,$$

with

$$\begin{align*}
\alpha_1 &= \frac{\alpha_0}{\sqrt{\alpha_0^2 + \beta_0^2}} \\
\beta_1 &= \frac{\beta_0}{\sqrt{\alpha_0^2 + \beta_0^2}}
\end{align*}$$

and the corresponding Rayleigh quotient will be

$$\lambda^1 = \frac{\alpha_1^2}{\alpha_1^2 + \beta_1^2} \lambda_1 + \frac{\beta_1^2}{\alpha_1^2 + \beta_1^2} \lambda_2$$

The factors in front of the last equations can be made more explicit by using the expression of $\lambda^0$

$$\begin{align*}
1 + \frac{\Delta t \lambda_0}{1 + \Delta t \lambda_1} &= 1 + \frac{\beta_0^2 \Delta t (\lambda_2 - \lambda_1)}{1 + \Delta t \lambda_1} \\
1 + \frac{\Delta t \lambda_0}{1 + \Delta t \lambda_2} &= 1 - \frac{\alpha_0^2 \Delta t (\lambda_2 - \lambda_1)}{1 + \Delta t \lambda_2}
\end{align*}$$

It follows that if $\alpha_0 \neq 0$, that is if $u^0$ has a nonvanishing component in the direction of $u_1$, then this component becomes more dominant when moving from $u^0$ to $u^1$, while the component along $u_2$ is reduced.

In general if we denote $u^k = \alpha_k u_1 + \beta_k u_2$ we get,

$$u^{k+1} = \alpha_{k+1} u_1 + \beta_{k+1} u_2.$$
\[ \lambda^{k+1} = \frac{\alpha_{k+1}^2}{\alpha_{k+1}^2 + \beta_{k+1}^2} \lambda_1 + \frac{\beta_{k+1}^2}{\alpha_{k+1}^2 + \beta_{k+1}^2} \lambda_2 \]

with

\[ \alpha_{k+1} = \alpha_k \frac{1 + \Delta t \lambda^k}{1 + \Delta t \lambda_1} = \alpha_k \left(1 + \frac{\beta_k^2}{\alpha_k^2 + \beta_k^2} (\lambda_2 - \lambda_1) \right) \]

\[ \beta_{k+1} = \beta_k \frac{1 + \Delta t \lambda^k}{1 + \Delta t \lambda_2} = \beta_k \left(1 - \frac{\alpha_k^2}{\alpha_k^2 + \beta_k^2} (\lambda_2 - \lambda_1) \right) \]

By iterating this procedure for larger values of \( k \), the steady state will be reached when \( \beta_k \) is going to zero and, consequently, \( u^k \) tends to be aligned with \( u_1 \).

**Example 3.** Let us take \( u^0 = \alpha_0 u_1 + \beta_0 u_2 + \gamma_0 u_3 \) and normalize it so that \( \alpha_0^2 + \beta_0^2 + \gamma_0^2 = 1 \). It follows that

\[ \lambda^0 = \alpha_0^2 \lambda_1 + \beta_0^2 \lambda_2 + \gamma_0^2 \lambda_3 \in [\lambda_1, \lambda_3] \]

Let us perform one step of the time continuation approach (6)

\[ \frac{1}{\Delta t} (u^1, v) + (\nabla u^1, \nabla v) = \lambda^0 (u^0, v) + \frac{1}{\Delta t} (u^0, v) \]

By using the relation \( (\nabla u_i, \nabla v) = \lambda_i (u_i, v) \) \( (i = 1, 2, 3) \) it can be seen that the solution \( u^1 \) is given by

\[ u^1 = \alpha_1 u_1 + \beta_1 u_2 + \gamma_1 u_3 \]

with

\[ \alpha_1 = \alpha_0 \frac{1 + \Delta t \lambda^0}{1 + \Delta t \lambda_1} \]

\[ \beta_1 = \beta_0 \frac{1 + \Delta t \lambda^0}{1 + \Delta t \lambda_2} \]

\[ \gamma_1 = \gamma_0 \frac{1 + \Delta t \lambda^0}{1 + \Delta t \lambda_3} \]

The factors in front of the last equations can be made more explicit by using the expression of \( \lambda^0 \)

\[ \frac{1 + \Delta t \lambda^0}{1 + \Delta t \lambda_1} = 1 + \frac{\beta_0^2 \Delta t (\lambda_2 - \lambda_1)}{1 + \Delta t \lambda_1} + \frac{\gamma_0^2 \Delta t (\lambda_3 - \lambda_1)}{1 + \Delta t \lambda_1} \]

\[ \frac{1 + \Delta t \lambda^0}{1 + \Delta t \lambda_2} = 1 + \frac{\alpha_0^2 \Delta t (\lambda_2 - \lambda_1)}{1 + \Delta t \lambda_2} - \frac{\gamma_0^2 \Delta t (\lambda_3 - \lambda_2)}{1 + \Delta t \lambda_2} \]

\[ \frac{1 + \Delta t \lambda^0}{1 + \Delta t \lambda_3} = 1 - \frac{\alpha_0^2 \Delta t (\lambda_3 - \lambda_1)}{1 + \Delta t \lambda_3} - \frac{\beta_0^2 \Delta t (\lambda_3 - \lambda_2)}{1 + \Delta t \lambda_3} \]

Note that \( \alpha_1 \geq \alpha_0 \) and \( \gamma_1 \leq \gamma_0 \) but it is not clear a priori if \( \beta_1 \) is smaller or larger than \( \beta_0 \). In general if we denote \( u^k = \alpha_k u_1 + \beta_k u_2 + \gamma_k u_3 \) then the next iterate is obtained as:

\[ u^{k+1} = \alpha_{k+1} u_1 + \beta_{k+1} u_2 + \gamma_{k+1} u_3 \]

with

\[ \alpha_{k+1} = \alpha_k \frac{1 + \Delta t \lambda^k}{1 + \Delta t \lambda_1} \]

\[ \beta_{k+1} = \beta_k \frac{1 + \Delta t \lambda^k}{1 + \Delta t \lambda_2} \]

\[ \gamma_{k+1} = \gamma_k \frac{1 + \Delta t \lambda^k}{1 + \Delta t \lambda_3} \]
where \( \lambda^k = \frac{\alpha^2_k}{\alpha^2_k + \beta^2_k + \gamma^2_k} \lambda_1 + \frac{\beta^2_k}{\alpha^2_k + \beta^2_k + \gamma^2_k} \lambda_2 + \frac{\gamma^2_k}{\alpha^2_k + \beta^2_k + \gamma^2_k} \lambda_3 \). Using this expression we get,

\[
\alpha_{k+1} = \alpha_k \left( 1 + \frac{\beta^2_k}{\alpha^2_k + \beta^2_k + \gamma^2_k} \Delta t (\lambda_2 - \lambda_1) + \frac{\gamma^2_k}{\alpha^2_k + \beta^2_k + \gamma^2_k} \Delta t (\lambda_3 - \lambda_1) \right) \]
\[
\beta_{k+1} = \beta_k \left( 1 - \frac{\alpha^2_k}{\alpha^2_k + \beta^2_k + \gamma^2_k} \Delta t (\lambda_2 - \lambda_1) - \frac{\gamma^2_k}{\alpha^2_k + \beta^2_k + \gamma^2_k} \Delta t (\lambda_3 - \lambda_2) \right) \]
\[
\gamma_{k+1} = \gamma_k \left( 1 - \frac{\alpha^2_k}{\alpha^2_k + \beta^2_k + \gamma^2_k} \Delta t (\lambda_3 - \lambda_1) + \frac{\beta^2_k}{\alpha^2_k + \beta^2_k + \gamma^2_k} \Delta t (\lambda_3 - \lambda_2) \right) \tag{8}
\]

with \( S^2_k = \alpha^2_k + \beta^2_k + \gamma^2_k \). From (8) it turns out that \( \{ \alpha_k \} \) is increasing while \( \{ \gamma_k \} \) is decreasing. On the other hand, it is not immediately clear the behavior of \( \{ \beta_k \} \) which depends on the sign of the difference \( \frac{\alpha^2_k}{\alpha^2_k + \beta^2_k + \gamma^2_k} \Delta t (\lambda_2 - \lambda_1) - \frac{\gamma^2_k}{\alpha^2_k + \beta^2_k + \gamma^2_k} \Delta t (\lambda_3 - \lambda_2) \). More precisely, \( \{ \beta_k \} \) is decreasing if and only if

\[
\frac{\alpha^2_k}{\beta^2_k} > \frac{\lambda_3 - \lambda_2}{\lambda_2 - \lambda_1}
\]

On the other hand, from (7), we easily get

\[
\frac{\alpha_k}{\gamma_k} = \frac{\alpha_{k-1}}{\gamma_{k-1}} \frac{1 + \Delta t \lambda_1}{1 + \Delta t \lambda_3} \to \infty \quad \text{as } k \to \infty
\]

We can then conclude that the steady state solution satisfies \( \beta_\infty = \gamma_\infty = 0 \) so that \( \lambda_\infty = \lambda_1 \).

The situation presented in Examples 2 and 3 can be generalized to an initial guess

\[
u^0 = \sum_{i=1}^{\infty} \alpha^0_i u_i \tag{9}
\]
and the following proposition can be obtained.

**Proposition 1.** If \( u^0 \) is given by (9) and \( \alpha^0_i \neq 0 \) then the solution of (6) converges to \( \lambda_\infty = \lambda_1 \).

### 2.2. Galerkin finite element discretization

The Galerkin method considers a finite dimensional subspace \( V_h \subset V \) of dimension \( N_h \) so that the fully discretized problem reads: given an approximation \( u_h^0 \) of \( u^0 \), for all \( k \geq 0 \) find \( \lambda_h^k \in \mathbb{R} \) and \( u_h^{k+1} \in V_h \) such that

\[
\lambda_h^k = \frac{\int_{\Omega} |\nabla u_h^k|^2}{\int_{\Omega} |u_h^k|^2} \quad \text{and} \quad \frac{1}{\Delta t} \int_{\Omega} u_h^{k+1} v \, dx + \int_{\Omega} \nabla u_h^k \cdot \nabla v \, dx = \lambda_h^k \int_{\Omega} u_h^k v \, dx + \frac{1}{\Delta t} \int_{\Omega} u_h^{k} v \, dx \quad \forall v \in V_h.
\]

The fully discretized problem reads: given \( u_h^0 \in V_h \) and \( \Delta t > 0 \), for each \( k = 0, 1, \ldots \), find \( u_h^{k+1} \in V_h \) such that

\[
\lambda_h^k \frac{\| \nabla u_h^k \|^2}{\| u_h^k \|^2} \quad \text{and} \quad \frac{1}{\Delta t} (u_h^{k+1}, v) + (\nabla u_h^k, \nabla v) = \lambda_h^k (u_h^k, v) + \frac{1}{\Delta t} (u_h^k, v) \quad \forall v \in V_h.
\tag{10}
\]
Let \( \varphi_i(x) \), \( i = 1, 2, \ldots, N_h \), be a basis of \( V_h \) and consider the representation \( u_h^k(x) = \sum_j u_j \varphi_j(x) \). Choosing \( v = \varphi_i \) for \( i = 1, 2, \ldots, N_h \), we finally get the following discrete form of the evolution equation

\[
\frac{1}{\Delta t} \sum_j u_j^{k+1} \int_{\Omega} \varphi_j(x) \varphi_i(x) \, dx + \sum_j u_j^{k+1} \int_{\Omega} \nabla \varphi_j(x) \cdot \nabla \varphi_i(x) = \left( \lambda_h^k + \frac{1}{\Delta t} \right) \sum_j u_j^k \int_{\Omega} \varphi_j(x) \varphi_i(x). \tag{11}
\]

The matrix form of the evolution problem is then

\[
AU^{k+1} + \frac{1}{\Delta t} MU^{k+1} = \left( \lambda_h^k + \frac{1}{\Delta t} \right) MU^k \tag{11}
\]

where

\[
M_{i,j} = \int_{\Omega} \varphi_i(x) \varphi_j(x) \, dx \quad \text{and} \quad A_{i,j} = \int_{\Omega} \nabla \varphi_i(x) \cdot \nabla \varphi_j(x) \, dx. \tag{12}
\]

We refer to this problem as high fidelity (HF) problem or as full order model (FOM). Here \( U^k \) denotes the vector of coefficients at the time step \( k \). We can rewrite the formula for calculating eigenvalues using these two matrices in discrete form as follows:

\[
\frac{\lambda_h^k}{\left( U_h^k \right)^T A U_h^k} = \left( \frac{U_h^k}{\left( U_h^k \right)^T M U_h^k} \right) \tag{13}
\]

where \( U^T \) is the transpose of \( U \). Starting from some initial guess \( U^0 \) and using the equation above we can evaluate the solution until the steady state is reached. The steady state solution will converge to the first eigenvector; putting this eigenvector into (13) we will get the corresponding eigenvalue. We explicitly observe that any random vector can be used as initial guess \( U^0 \) for calculating the first eigenvalue [see Table 2]. Even if we start with a small perturbation of the second eigenvector then the solution still converges to the first eigenvector [see Table 3].

This result can be made more precise by repeating the same analysis performed in Proposition 1. We denote by \( (\lambda_{i,h}, u_{i,h}) \) \( i = 1, 2, \ldots, \dim(V_h) \), the discrete eigenpairs of the Laplace operator with orthonormalized eigenfunctions, that is

\[
(\nabla u_{i,h}, \nabla v) = \lambda_{i,h}(u_{i,h}, v) \quad \forall v \in V_h \nabla
(\nabla u_{i,h}, \nabla v) = \lambda_{i,h}(u_{i,h}, v) \quad \forall v \in V_h
\]

\[
\lambda_{1,h} \leq \lambda_{2,h} \leq \lambda_{3,h} \leq \cdots \leq \lambda_{\dim(V_h),h} \nabla
\lambda_{1,h} \leq \lambda_{2,h} \leq \lambda_{3,h} \leq \cdots \leq \lambda_{\dim(V_h),h} \nabla
\]

The same discussion presented in Examples 1, 2, and 3 can be repeated in the discrete setting, leading to the following result.

**Proposition 2.** If \( u_0^0 \) is given by

\[
u_0^0 = \sum_{i=1}^{\dim(V_h)} \alpha_{0,h}^i u_{i,h}\nabla
\]

and \( \alpha_{0,h}^i \neq 0 \) then the solution of (10) converges to \( \lambda_{h}^\infty = \lambda_{1,h} \).

2.3. Time continuation strategy for higher eigenmodes

We know that the \( j \)th eigenvalue of the EVP \( a(u, v) = \lambda b(u, v) \) is characterized by

\[
\lambda^{(j)} = \min_{v \in \bigoplus_{\lambda^{(i)}} E^{(i)}} \frac{a(v, v)}{b(v, v)}, \nabla
\]

where \( E^{(i)} \) are the eigen spaces corresponding to the eigenvalue \( \lambda^{(i)} \). So, to get the next eigenmodes we need to subtract the components in the directions of all the previous eigenvectors in each time iterations starting from the initial guess. That is we need to update the iterates as follows:

\[
U_j^k = U_j^{k-1} - \sum_{i=1}^{j-1} \frac{(U_j^k, u_i)_L^2}{\| u_i \|_L^2} u_i, \quad k = 0, 1, \ldots, \quad j = 2, 3, \ldots \tag{14}
\]

F. Bertrand, D. Boffi and A. Halim Computer Methods in Applied Mechanics and Engineering 404 (2023) 115696
where \( U^j \) denotes the \( k \)th iterate of the \( j \)th eigenvector and \( u_1, u_2, \ldots u_{j-1} \) are the final iterates of previous eigenvectors that is \( U_{j}^{\text{final}} = u_i \). We will refer this updating process as orthogonalization.

### 3. Reduced order model

With a reduced order model we aim at calculating an approximate solution to the discrete problem (11) in a subspace \( V_N \subset V_h \) of dimension \( N \), where \( N \ll N_h \). The definition of the subspace \( V_N \) is related to the choice of a suitable basis which stems from the high fidelity solution corresponding to an appropriate choice of the parameters. Possible approaches for calculating the basis of \( V_N \) include the use of a greedy approach or of a Proper Orthogonal Decomposition (POD). In the greedy approach only \( N \) snapshots are selected based on some optimal criterion. On the other hand, in the POD approach \( n_s \) snapshots are taken at \( n_s \) different predefined values of the parameters. Different sampling techniques like tensorial sampling, Monte Carlo sampling, Latin hypercube sampling, Clenshaw–Curtis points may be used to select the predefined parameters. Here we shall adopt a POD approach and use tensorial sampling for choosing the parameters. Actually, we will choose uniformly distributed points in the parameter (time) space. We need to choose the parameters in such a way that it represents the parameter space. Let \( S \) be the snapshot matrix of size \( N_h \times n_s \), that is the \( n_s \) columns of the \( S \) are formed by the snapshots computed with the high fidelity model. The singular values of \( S \) are computed and sorted in decreasing order. Then the first \( N \) left singular vectors are taken as the basis of the subspace \( V_N \). In the next subsection we are discussing how to calculate the first \( N \) left singular vectors of \( S \).

#### 3.1. Construction of the POD basis

Let \( S \) be the matrix of the \( n_s \) snapshots. The SVD gives

\[
S = U \Sigma Z^T
\]

with

\[
U = [\zeta_1, \ldots, \zeta_{N_h}] \in \mathbb{R}^{N_h \times N_h} \quad \text{and} \quad Z = [\psi_1, \ldots, \psi_{n_s}] \in \mathbb{R}^{n_s \times n_s}
\]

orthogonal matrices and \( \Sigma = \text{diag}(\sigma_1, \ldots, \sigma_r, 0, \ldots, 0) \in \mathbb{R}^{N_h \times n_s} \) with \( \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0 \), where \( r \) is the rank of the matrix \( S \). As we mentioned earlier, \( \{\zeta_1, \ldots, \zeta_{N_h}\} \) will be the basis of the low-dimensional subspace \( V_N \). We will not use the SVD directly to find the left singular vectors unless \( n_s \geq N_h \). Next, we describe the process of calculating the POD basis. We can write

\[
S \psi_i = \sigma_i \zeta_i \quad \text{and} \quad S^T \zeta_i = \sigma_i \psi_i, \quad i = 1, \ldots, r
\]

or, equivalently,

\[
S^T S \psi_i = \sigma_i^2 \psi_i \quad \text{and} \quad S^T \zeta_i = \sigma_i^2 \zeta_i, \quad i = 1, \ldots, r.
\]

The matrix \( C = S^T S \) is called correlation matrix. The POD basis \( \mathcal{V} \) of dimension \( N \leq n_s \) is defined as the set of the first \( N \) left singular vectors \( \{\zeta_1, \ldots, \zeta_N\} \) of \( S \) or, equivalently, the set of vectors

\[
\zeta_j = \frac{1}{\sigma_j} S \psi_j, \quad 1 \leq j \leq N
\]

obtained from the first \( N \) eigenvectors \( \{\psi_1, \ldots, \psi_{N_s}\} \) of the correlation matrix \( C \). So, instead of using the SVD of the matrix \( S \) we solve the eigenvalue problem \( C \psi = \sigma^2 \psi \) which is of size \( n_s \); notice that \( n_s \) is usually much smaller than \( N_h \).

Let us now discuss the relation between the solution of the problem in \( V_h \) and the approximate solution in \( V_N \). Since the basis functions \( \zeta_m, m = 1, 2, \ldots \) belong to \( V_h \), they can be expressed as linear combinations of basis of \( V_h \), namely

\[
\zeta_m = \sum_{i=1}^{N_h} \zeta_m^i \varphi_i \quad 1 \leq m \leq N.
\]

The transformation matrix \( \mathcal{V} \) is actually the matrix containing all the coefficients \( \zeta_m^i \), that is

\[
(\mathcal{V})_{i,m} = \zeta_m^i, \quad 1 \leq m \leq N, \quad 1 \leq i \leq N_h.
\]
Hence, for any linear function $f$, we have $f(\zeta_m) = \sum_{i=1}^{N_h} \zeta_i f(\varphi_i)$, $1 \leq m \leq N$ or, in matrix form, $f_N = \mathsf{V}^T f_h$ where $(f_h)_i = f(\varphi_i)$ and $(f_N)_m = f(\zeta_m)$. Since $\mathsf{V}$ is orthogonal we have $f_h = \mathsf{V} f_N$. The solution $U^k$ of Eq. (11) is approximated by $U^k = \mathsf{V}_N^k$, where $U_N^k$ is the vector containing the coefficients of the approximate solution at time level $k$ in the subspace $V_N$. Finally, premultiplying Eq. (11) by $\mathsf{V}^T$ and using the relation $U^k = \mathsf{V} U_N^k$, we get the formula for reduced order solution at time level $k + 1$ as:

$$\mathsf{V}^T A \mathsf{V} U_N^{k+1} + \frac{1}{\Delta t} \mathsf{V}^T M \mathsf{V} U_N^k = \left( \lambda^k + \frac{1}{\Delta t} \right) \mathsf{V}^T M \mathsf{V} U_N^k. \tag{18}$$

The approximate solution in $V_h$ is given by $U^{k+1} = \mathsf{V}_N^{k+1}$ and the corresponding eigenvalue can be calculated using (13). As these iterates converges to the first eigenvector we will get the first eigenpair.

3.2. Reduced problem for higher eigenmodes

There are two ways we can compute the later eigenvalues for the reduced problem (i) we can calculate separately by imposing orthogonality and (ii) multiple eigenvalues simultaneously.

3.2.1. Technique I: separate computation of different eigenvalues

Suppose we are interested to find the $p$th eigenvalue. In this technique, we calculate the iterates of the $p$th eigenvector using the time continuation problem and imposing orthogonality and put them together to form the snapshot matrix. Then applying SVD/POD to the snapshot matrix we will get the transform/ basis matrix $\mathsf{V}_p$. Once we get the matrix $\mathsf{V}_p$, we will get the corresponding reduced system (18) and solving steady state we will get the required eigenvector in reduced space. Pre-multiplying the reduced eigenvector by $\mathsf{V}_p$ and using the resulting vector in Rayleigh quotient (13) we get the corresponding reduced eigenvalue.

Let us describe the technique for the second eigenmode. Suppose, $u_1$ be the first eigenvector EVP. Then to get the iterates of the second eigenvector, we need to solve the time continuation problem with following modification in each iterations:

$$U_2^k = U_2^k - \frac{(U_2^k, u_1)}{\|u_1\|_2^2} u_1, \quad k = 0, 1, \ldots$$

where $U_2^k$ denotes the $k$th iterate for second eigenvector and the corresponding eigenvalue ($\lambda_2^k$) can be calculated using (13). In this case also we choose $U_2^0$ randomly and the modified according to the above formula. Then it converges to the second eigenvector $u_2$. Applying SVD or the procedure described in Section 3.1 to the snapshot matrix and we will get the transform matrix $\mathsf{V}_2$. Using this matrix form the reduced system (18) and solving it we will get the required second eigenvector and using that vector in Rayleigh quotient we get the second eigenvalue.

3.2.2. Technique II: simultaneous computation of different eigenvalues

In this case, let us assume we are interested in finding the first $p$ eigenvalues of the problem. Then we need to put the snapshots corresponding to all the first $p$-eigenvectors. The iterates of first eigenvector is obtained by solving the time continuation problem and the others are obtained by solving the time continuation problem with the update (14). Let $N_i$ be the final iterates for the $i$th eigenvector of the time continuation problem. Then we calculate all the vectors $\{U_1^1, \ldots, U_1^{N_1}, U_2^1, \ldots, U_2^{N_2}, \ldots, U_p^1, \ldots, U_p^{N_p}\}$. We have not include all the these vectors in the snapshot matrix but include the solution 4th time steps for each eigenvectors. Then employing the POD technique to the snapshot matrix we will get the transform reduced matrix $\mathsf{V}_{1:p}$. Then we form the following reduced matrix:

$$\mathsf{V}_{1:p}^T A \mathsf{V}_{1:p} = \lambda \mathsf{V}_{1:p}^T M \mathsf{V}_{1:p}. \tag{19}$$

Then the first $p$ eigenvalues of the above reduced system will be the first $p$-eigenvalues approximate of the original problem. By pre-multiplying the first $p$ eigenvalues of the reduced system (19) by the transform matrix $\mathsf{V}_{1:p}$, we will get the first $p$ approximate eigenvectors of the original eigenvalue problem. Note that this approach is applicable for the first eigenvalue also, with the choice $p = 1$.

Remark 1. We can also use the reduced time continuation system (18) by replacing $\mathsf{V}$ with $\mathsf{V}_{1:p}$ to get the later eigenvectors by imposing orthogonality in reduced problem. Once we get the eigenvector, we will get the corresponding eigenvalues using the Rayleigh quotient.
3.3. POD-based RB methods and selection of the dimension of the basis

The POD basis is orthonormal and it minimizes the sum of squares of errors between each snapshot vector $u_i$ $(u_i$ is the $i$th column vector of the snapshot matrix) and its projection onto any $N$-dimensional subspace $W$. More precisely it satisfies the following proposition which is a consequence of Schmidt–Eckart–Young theorem.

**Theorem 1 (Schmidt–Eckart–Young, see [42]).** Given a matrix $A \in \mathbb{R}^{m \times n}$ of rank $r$, the matrix

$$A_k = \sum_{i=1}^{k} \sigma_i \xi_i \psi_i^T, \quad 1 \leq k \leq r$$

satisfies the optimality property

$$\|A - A_k\|_F = \min_{B \in \mathbb{R}^{m \times k}} \|A - B\|_F = \sqrt{\sum_{i=k+1}^{r} \sigma_i^2},$$

(20)

where $\| \cdot \|_F$ is the Frobenius matrix norm.

**Proposition 3 (see [42]).** Let $\mathcal{W} = \{W \in \mathbb{R}^{N \times N} : W^T W = I_N\}$ be the set of all $N$-dimensional bases. Then

$$\sum_{i=1}^{n_t} \|u_i - W^T W u_i\|^2 = \min_{W \in \mathcal{W}} \sum_{i=1}^{n_t} \|u_i - W^T W u_i\|^2 = \sum_{i=N+1}^{r} \sigma_i^2.$$  

(21)

This proposition implies that the error in the POD basis is the sum of the squares of the singular values corresponding to the neglected POD modes. This result gives rise to a criterion for selecting the optimal dimension $N$ of the POD basis. For a given tolerance $\epsilon$, $N$ will be the smallest integer such that the percentage of energy $I(N)$ of the first $N$ POD modes is less than or equal to $\epsilon$, that is

$$I(N) = \frac{\sum_{i=1}^{N} \sigma_i^2}{\sum_{i=1}^{r} \sigma_i^2} \geq 1 - \epsilon^2.$$  

(22)

The convergence analysis presented for Galerkin FEM problem, see Proposition 2, will also be applicable to the reduced order model after replacing $V_h$ by $V_N$. This will show in particular that if $V_N$ contains a good approximation of the first eigenfunction then the reduced model will provide a good approximation of the first discrete eigenvalue.

As a typical example of convergence result, we recall the a priori estimate of [43,44]. It refers to the case of elliptic PDEs with single parameter where the associated bilinear form is

$$a(u, v; \mu) = a_0(u, v) + \mu a_1(u, v) \quad \forall u, v \in V$$  

(23)

with $a_0 : V \times V \to \mathbb{R}$ and $a_1 : V \times V \to \mathbb{R}$ continuous, symmetric, positive semi-definite, and $a_0$ coercive. Then there exists a positive constant $\gamma_1$ such that

$$0 \leq \frac{a_1(v, v)}{a_0(v, v)} \leq \gamma_1 \quad \forall v \in V.$$  

Let us choose the parameter sample points $\mu_i, i = 1, 2, \ldots, N$ log-equidistributed in the parameter space $\mathcal{D}$. With this choice of parameters the solution of the reduced order model $u_N(\mu)$ in the space spanned by the corresponding solutions satisfies the following theorem which gives the a priori exponential convergence, where $\| \cdot \|^2 = a_0(\cdot, \cdot)$.

**Proposition 4.** For $N \geq N_{crit}$

$$\|u_N(\mu) - u_N(\mu)\| \leq (1 + \mu_{max}\gamma_1)^{\frac{1}{2}}\|u_0(0)\|e^{-N/N_{crit}}, \quad \forall \mu \in \mathcal{D}$$  

(24)

where $N_{crit} = c^* e \cdot \log(\gamma_0\mu_{max} + 1)$ with $\gamma$ is an upper bound of $\gamma_1$.

In [23] the sampling parameters are chosen as

$$\mu_i = \mu_{min} \exp \left\{ \frac{i - 1}{N - 1} \ln \left( \frac{\mu_{max}}{\mu_{min}} \right) \right\}, \quad 1 \leq i \leq N,$$  

(25)
where the parameter space is $\mathcal{D} = [\mu_{\text{min}}, \mu_{\text{max}}]$ and the corresponding low-dimensional space is
\[
W_N = \text{span}\{u_h(\mu_i), 1 \leq i \leq N\}.
\]

With this choice, the following results can be obtained.

**Proposition 5.** For any $N \geq N_{\text{crit}}$ and for all $\mu \in \mathcal{D}$
\[
\frac{\|u_h(\mu) - u_N(\mu)\|}{\|u_h(\mu)\|} \leq \exp\left\{-\frac{N - 1}{N_{\text{crit}} - 1}\right\}
\]
where $N_{\text{crit}} = 1 + [2e \ln \mu_r]$ and $u_h(\mu), u_N(\mu)$ are the FE solution and reduced-order solution in low-dimensional space $W_N$, respectively.

**Remark 2.** The error decays exponentially fast with respect to $N$ under the assumption that the solution manifold is analytic [42].

4. Numerical results

In this section we show the results of our computations with the approach described above. We point out that the present research should be considered as a proof of concept towards more general problems. In this context, we are using the terminologies “offline” and “online” phases with a meaning that might seem not perfectly matching the usual one. However, this makes sense if we consider that the time continuation approach can be used also for parametric eigenvalue problems (see Section 4.4) and that we are planning to extend our approach to more general problems.

We are mainly approximating the first eigenvalue of the model problem but include few results for later eigenvalues also. We consider different domains with different types of triangular meshes and use a proper orthogonal decomposition (POD) based reduced order model (ROM). We consider the square domain $(0, \pi)^2$ and the L-shaped domain $(-1, 1)^2 \setminus \{(0, 1) \times [0, -1]\}$ with various mesh sequences: uniform meshes of type crisscross, right, and left, and non uniform meshes of Delaunay type. In our case, the parameter is the time $t$. We consider the time step size $\Delta t = 0.1$ and set the stopping criterion as $\frac{\|u^{k+1} - u^k\|}{\|u^{k+1}\|} \leq 10^{-8}$ for both ROM and FOM. We could start with any random guess; here we take a vector $U^0$ with all elements zero except the first fifteen ones which are taken equal to 1; then we take a vector $W^0$ with all elements equal to 1 except the first fifteen which are taken equal to 0 and we use $U^0_N = \Psi^T W^0$ as initial guess in the ROM system (18), where $\Psi$ is the basis matrix.

4.1. Results for first eigenvalues

We use three sets of equally spaced parameters to form snapshot matrices. The corresponding snapshot matrices are denoted by $S_2, S_4, S_8$, respectively, as they consists of solutions of every 2, 4 and 8 time steps. We developed our FEM code based on the code provided in [45,46].

4.1.1. Results on the square domain

First, we compare the different choices of snapshot matrices mentioned above. Table 1, shows the first eigenvalue corresponding to the different snapshot matrices on the crisscross mesh sequence. We have used P1 FEM for solving the high-fidelity problem. In the table, $n$ denotes the number of sub-intervals on each side of the square domain. The second, third, and fourth columns contain the eigenvalues corresponding to the solution of reduced order model with snapshot matrices $S_2, S_4, S_8$, respectively. In the last column we have reported the minimum number $N$ of POD basis functions satisfying the criterion coming from (22), where the tolerance $\varepsilon$ is set as the norm of the difference between the exact and approximate first eigenvectors in normal form. We can see that if the mesh is fine enough then all results corresponding to different snapshot matrices are similar to each other. Also, notice that the results corresponding to the snapshot matrices $S_4$ and $S_8$ are similar except the first two cases. From the last column one can see that the results corresponding to $S_4$ and $S_8$ are obtained with a smaller number of POD basis functions with respect to $S_2$ and that for these two cases the number of POD basis functions is comparable. For this reason, we will use $S_4$ snapshot matrix in the next results. In Fig. 1 we have shown the first few singular values for the crisscross mesh with number of sub-intervals $n = 16$ on each side of the square domain. We can see that the
Table 1
Comparison of first eigenvalue using ROM for three different snapshot matrices on crisscross mesh. For high-fidelity problem we have used P1-FEM.

| n  | $S_2$                  | $S_4$                  | $S_8$                  | $N$  |
|----|------------------------|------------------------|------------------------|------|
| 8  | 2.0215691669           | 2.0215680613           | 2.0215680546           | 4,4,4|
| 16 | 2.0053640028(2.1)      | 2.0053639952(2.1)      | 2.0053639952(2.1)      | 6,5,4|
| 32 | 2.0013392391(2.0)      | 2.0013392384(2.0)      | 2.0013392384(2.0)      | 6,5,4|
| 64 | 2.0003346995(2.0)      | 2.0003346995(2.0)      | 2.0003346994(2.0)      | 7,5,5|
| 128| 2.0000836680(2.0)      | 2.0000836680(2.0)      | 2.0000836680(2.0)      | 8,6,5|
| 256| 2.0000209166(2.0)      | 2.0000209166(2.0)      | 2.0000209166(2.0)      | 8,7,5|
| 512| 2.0000052291(2.0)      | 2.0000052291(2.0)      | 2.0000052291(2.0)      | 9,7,6|

Fig. 1. Singular values corresponding to the snapshot matrices $S_2$, $S_4$ and $S_8$ for crisscross mesh with $n = 16$. The singular values corresponding to $S_8$ decay much faster than the other two cases. From the theory it is then expected that this snapshot matrix can give better results; this is confirmed by the numbers shown in Table 1.

It is interesting to investigate the impact of the choice of the initial guess $U^0$ on the first eigenvalue. In Table 2 we report the results of the computation of the first eigenvalue when the initial guess $U^0$ is generated using the MATLAB function ‘rand’ and the crisscross mesh is used. Note that with the random choice of the initial vector we get as before the convergence towards the first eigenvector with rate 2.

If we take as initial guess $U^0$ the second eigenvector with a small random perturbation of order $10^{-8}$, we observe that the time continuation problem still converges to the first eigenvector. The results are shown in Table 3. In this case, we can see that the dimension of the reduced basis is 3 for all meshes.

Another interesting subject of investigation is related to the dependence of the results on the different types of triangular meshes. Here we consider our four types of triangular meshes: crisscross, right, left, and Delaunay mesh. A sample mesh on unit square domain has been shown in Fig. 2. As we have already mentioned, we take $S_4$ as the snapshot matrix also for this test. In Table 4 we have reported the first eigenvalue on four different kinds of triangular meshes with different degrees of freedom. In the last column of the table we have indicated the number of POD basis functions which is obtained by utilizing the criterion coming from (22). The rate of convergence of the eigenvalues is written within parentheses. Observe that the rate of convergence is 2 for all the meshes as expected.
Fig. 2. Different type of triangulation on unit square mesh, with n = 4 sub-interval on each side. Crisscross, Right, Left and Delaunay triangulation.

Table 2
Comparison of first eigenvalue using random initial guess on crisscross mesh.

| n   | FOM                  | ROM                  | DOF | N  |
|-----|----------------------|----------------------|-----|----|
| 8   | 2.077646080267       | 2.077646080280       | 81  | 4  |
| 16  | 2.019309896556(2.2)  | 2.019309896556(2.2)  | 289 | 5  |
| 32  | 2.004821215327(2.1)  | 2.004821215327(2.1)  | 1089| 6  |
| 64  | 2.001204915048(2.1)  | 2.001204915048(2.1)  | 4225| 6  |
| 128 | 2.000301204505(2.0)  | 2.000301204505(2.0)  | 16641| 7  |
| 256 | 2.000075299611(2.0)  | 2.000075299611(2.0)  | 66049| 7  |

because we are using P1 finite elements and the eigenspace is smooth. The CPU time taken by the full order model and the CPU time for the reduced order model are reported in Table 5. From the CPU time reported in Table 5 one can see that in the ROM the online phase is from 10 to 30 times faster than the full order model. Notice that the results on the right and left uniform meshes are exactly the same. These two types of meshes are the same with a different orientation. Hence, from now on we will use only the right uniform mesh for our tests. In all cases, when the mesh is fine enough, then the results obtained by the ROM are exactly same as the ones obtained by the FOM.

Next, we have used P2 FEM for the high-fidelity problem. The results corresponding to P2 FEM on the square meshes are reported in Table 6. In this case we have considered triangular meshes of type crisscross, right, and
Table 3
First eigenvalue using perturbed second eigenvector as initial guess on crisscross mesh.

| n  | FOM                      | ROM                      | DOF | N         |
|----|--------------------------|--------------------------|-----|-----------|
| 8  | 2.077646080267           | 2.077646080267           | 81  | 3         |
| 16 | 2.019309896556(2.2)      | 2.019309896556(2.2)      | 289 | 3         |
| 32 | 2.004821215327(2.1)      | 2.004821215327(2.1)      | 1089| 3         |
| 64 | 2.001204915048(2.1)      | 2.001204915048(2.1)      | 4225| 3         |
| 128| 2.000301204505(2.0)      | 2.000301204505(2.0)      | 16641| 3         |
| 256| 2.00075299611(2.0)       | 2.00075299611(2.0)       | 66049| 3         |

Table 4
Comparison of first eigenvalue on different meshes using P1 FEM on square domain.

| Mesh     | n   | FOM                      | ROM                      | DOF | N         |
|----------|-----|--------------------------|--------------------------|-----|-----------|
| Crisscross| 16  | 2.005363995049           | 2.005363995229           | 545 | 5         |
|          | 32  | 2.001339238351(2.1)      | 2.001339238375(2.1)      | 2113| 5         |
|          | 64  | 2.000334699425(2.0)      | 2.000334699426(2.0)      | 8321| 5         |
|          | 128 | 2.000083667969(2.0)      | 2.000083667969(2.0)      | 33025| 6         |
|          | 256 | 2.000020916562(2.0)      | 2.000020916562(2.0)      | 131585| 7        |
|          | 512 | 2.000005229114(2.0)      | 2.000005229114(2.0)      | 525313| 7        |
| Right    | 16  | 2.004821215327(2.1)      | 2.004821215369(2.1)      | 1089| 5         |
|          | 32  | 2.001204915048(2.1)      | 2.001204915048(2.1)      | 4225| 6         |
|          | 64  | 2.000301204505(2.0)      | 2.000301204506(2.0)      | 16641| 6         |
|          | 128 | 2.000075299611(2.0)      | 2.000075299611(2.0)      | 66049| 7         |
|          | 256 | 2.000018824887(2.0)      | 2.000018824807(2.0)      | 263169| 7        |
|          | 512 | 2.000005229114(2.0)      | 2.000005229114(2.0)      | 525313| 7        |
| Left     | 16  | 2.001339238351(2.1)      | 2.001339238375(2.1)      | 1089| 5         |
|          | 32  | 2.001204915048(2.1)      | 2.001204915048(2.1)      | 4225| 6         |
|          | 64  | 2.000301204505(2.0)      | 2.000301204506(2.0)      | 16641| 6         |
|          | 128 | 2.000075299611(2.0)      | 2.000075299611(2.0)      | 66049| 7         |
|          | 256 | 2.000018824887(2.0)      | 2.000018824807(2.0)      | 263169| 7        |
|          | 512 | 2.000005229114(2.0)      | 2.000005229114(2.0)      | 525313| 7        |
| Delaunay | 16  | 2.000363995049           | 2.000363995229           | 545 | 5         |
|          | 32  | 2.000334699425(2.0)      | 2.000334699426(2.0)      | 8321| 5         |
|          | 64  | 2.000083667969(2.0)      | 2.000083667969(2.0)      | 33025| 6         |
|          | 128 | 2.000020916562(2.0)      | 2.000020916562(2.0)      | 131585| 7        |
|          | 256 | 2.000005229114(2.0)      | 2.000005229114(2.0)      | 525313| 7        |

Table 5
CPU time for FOM and ROM corresponding to the results of P1 FEM on square domain.

| Mesh     | Method | n = 16 | n = 32 | n = 64 | n = 128 | n = 256 | n = 512 |
|----------|--------|--------|--------|--------|---------|---------|---------|
| Crisscross| FOM    | 0.04   | 0.14   | 0.85   | 3.57    | 16.38   | 83.61   |
|          | ROM    | 0.008  | 0.01   | 0.03   | 0.13    | 0.60    | 2.79    |
| Right    | FOM    | 0.02   | 0.08   | 0.39   | 1.66    | 7.56    | 42.53   |
|          | ROM    | 0.006  | 0.008  | 0.02   | 0.06    | 0.24    | 1.25    |
| Left     | FOM    | 0.02   | 0.08   | 0.39   | 1.58    | 7.26    | 41.15   |
|          | ROM    | 0.006  | 0.008  | 0.02   | 0.06    | 0.25    | 1.23    |
| Delaunay | FOM    | 0.04   | 0.18   | 0.90   | 4.10    | 22.34   | 185.56  |
|          | ROM    | 0.005  | 0.01   | 0.04   | 0.15    | 0.76    | 5.52    |

Delaunay. For crisscross mesh we took a number of sub-intervals equal to 8, 16, . . . , 512, that is we have done uniform refinement in each step. The results are converging to the exact solution with rate of convergence equal to 4. In the last case the result is correct up to six decimal place. Like in the P1 case, here also we have reported the CPU time for both the FOM and the ROM computations in Table 7. From the table it is clear that the ROM is 10−15 times
Table 6
Comparison of first eigenvalue on three different mesh with P2 FEM on square domain.

| Mesh     | n  | FOM         | ROM         | DOF   | N   |
|----------|----|-------------|-------------|-------|-----|
| Crisscross | 16 | 2.0000034983508(4.0) | 2.0000034983774(4.1) | 2113  | 6   |
|          | 32 | 2.000002191655(4.0)  | 2.000002191679(4.0)  | 8321  | 6   |
|          | 64 | 2.000000137061(4.0)  | 2.000000137061(4.0)  | 33025 | 7   |
|          | 128| 2.000000008569(4.0)  | 2.000000008569(4.0)  | 131585| 8   |
|          | 256| 2.000000000535(4.0)  | 2.000000000535(4.0)  | 525313| 8   |
|          | 512| 2.000000000007(6.0)  | 2.0000000000006(6.0) | 2099201| 8  |
| Right    | 16 | 2.0000286902960(4.0) | 2.0000286902960(4.1) | 1089  | 6   |
|          | 32 | 2.0000018029662(4.0) | 2.0000018029662(4.1) | 4225  | 7   |
|          | 64 | 2.0000001128424(4.0) | 2.0000001128424(4.0) | 16641 | 7   |
|          | 128| 2.0000000070555(4.0) | 2.0000000070555(4.0) | 66049 | 7   |
|          | 256| 2.0000000004417(4.0) | 2.0000000004417(4.0) | 263169| 7   |
|          | 512| 2.000000000000333(3.7)| 2.000000000000333(3.7) | 1050625 | 8  |
| Delaunay | 16 | 2.0000058772706(4.1) | 2.0000058772719(4.1) | 1799  | 6   |
|          | 32 | 2.0000003559407(4.1) | 2.0000003559447(4.1) | 47003 | 6   |
|          | 64 | 2.0000000216176(4.0) | 2.0000000216177(4.0) | 28299 | 7   |
|          | 128| 2.0000000013122(4.0) | 2.0000000013121(4.0) | 113575| 7   |
|          | 256| 2.0000000000813(4.0) | 2.0000000000813(4.0) | 454355| 8   |
|          | 512| 2.0000000000000051(4.0)| 2.0000000000000051(4.0) | 1816879 | 8  |

Table 7
CPU time for FOM and ROM corresponding to the results of P2-FEM on square domain.

| Mesh     | Method | n = 16  | n = 32  | n = 64  | n = 128 | n = 256 | n = 512 |
|----------|--------|---------|---------|---------|---------|---------|---------|
| Crisscross | FOM    | 0.16    | 0.78    | 3.40    | 17.53   | 86.37   | 414.58  |
|          | ROM    | 0.02    | 0.04    | 0.17    | 0.89    | 3.80    | 22.12   |
| Right    | FOM    | 0.10    | 0.47    | 1.80    | 8.80    | 46.65   | 222.04  |
|          | ROM    | 0.01    | 0.02    | 0.08    | 0.39    | 1.74    | 15.52   |
| Delaunay | FOM    | 0.20    | 0.05    | 4.60    | 24.30   | 145.71  | 847.67  |
|          | ROM    | 0.02    | 0.04    | 0.19    | 1.29    | 5.12    | 38.40   |

Fig. 3. Different type of Triangular mesh of L-shaped domain with n = 4 sub-interval on each side of the squares. Left to right: Crisscross, Delaunay, Mixed triangulation.

faster than the corresponding FOM. Similarly, for the right uniform mesh also the order of convergence is equal to 4 except for the last case. In this case also the ROM is giving the solution 10—20 times faster than the offline stage. In the finest mesh the eigenvalue is matching the exact one up to 10 decimal place and the corresponding number of degrees of freedom is 1,050,625. Similar conclusions can be drawn for the unstructured Delaunay mesh.
| Mesh     | n    | FOM       | ROM       | DOF | N   |
|----------|------|-----------|-----------|-----|-----|
| Crisscross | 16   | 9.689550909492 | 9.689550909499 | 1601 | 5   |
|          | 32   | 9.65719368400(1.53) | 9.65719368400(1.53) | 6273 | 6   |
|          | 64   | 9.646089739637(1.47) | 9.646089739637(1.47) | 24833 | 6   |
|          | 128  | 9.642109199667(1.42) | 9.642109199667(1.42) | 98817 | 6   |
|          | 256  | 9.64065313503(1.39) | 9.64065313503(1.39) | 394241 | 7   |
|          | 512  | 9.640076778359(1.37) | 9.640076778359(1.37) | 1574913 | 7   |
| Mixed    | 16   | 9.737622827078 | 9.737622827084 | 833  | 5   |
|          | 32   | 9.672057256699(1.65) | 9.672057256699(1.65) | 3201  | 6   |
|          | 64   | 9.650967866448(1.55) | 9.650967866448(1.55) | 12545 | 6   |
|          | 128  | 9.643793757105(1.48) | 9.643793757105(1.48) | 49665 | 6   |
|          | 256  | 9.64124147249(1.43) | 9.64124147249(1.43) | 197633 | 7   |
|          | 512  | 9.640301753187(1.40) | 9.640301753187(1.40) | 788481 | 7   |
| Delaunay | 16   | 9.691713765278 | 9.691713765278 | 1356 | 5   |
|          | 32   | 9.65920378910(1.43) | 9.65920378910(1.43) | 5354 | 6   |
|          | 64   | 9.645755044694(1.69) | 9.645755044694(1.69) | 21406 | 6   |
|          | 128  | 9.64194970951(1.44) | 9.64194970951(1.44) | 85651 | 6   |
|          | 256  | 9.640665163096(1.25) | 9.640665163096(1.25) | 338736 | 7   |

Table 9

| Mesh     | Method | n = 16 | n = 32 | n = 64 | n = 128 | n = 256 | n = 512 |
|----------|--------|--------|--------|--------|---------|---------|---------|
| Crisscross | FOM    | 0.9    | 0.40   | 2.07   | 9.69    | 48.94   | 226.98  |
|          | ROM    | 0.01   | 0.03   | 0.10   | 0.45    | 2.31    | 9.68    |
| Right    | FOM    | 0.05   | 0.22   | 1.09   | 5.15    | 23.60   | 124.88  |
|          | ROM    | 0.008  | 0.01   | 0.04   | 0.22    | 1.07    | 4.60    |
| Delaunay | FOM    | 0.10   | 0.59   | 2.63   | 13.47   | 109.02  | –       |
|          | ROM    | 0.01   | 0.03   | 0.11   | 0.57    | 3.32    | –       |

4.1.2. Results on the L-shaped domain

In this last section we investigate how our ROM method is performing on the L-shaped domain \((-1, 1)^2 \times [0, 1] \times [0, -1]\) where singular solutions are expected. We consider three types of triangular meshes: crisscross, Delaunay and what we call mixed uniform mesh, which is a combination of the right and left meshes. A sample mesh has been shown in Fig. 3. In this case \(n = 4\) is the number of sub-intervals in each side of the unit square and the L-shaped domain consists of three unit squares.

In Table 8 we reported the results of our reduced order model on the singular L-shaped domain using three different triangular meshes and P1 FEM in the full order model. The rate of convergence is reported within parentheses. We report the CPU time for FOM and ROM in Table 9. We can see that ROM is at least 13 times faster than FOM. For example, on the crisscross uniform mesh with \(n = 256\), the time required for FOM is 48.94 s, whereas CPU time for ROM is 2.31 s, that is 21 times less. The rate of convergence is around 1.4 where we take as a reference value for the first eigenvalue 9.6397238440219. It can be seen that the eigenvalue matches up to only 2 decimal places the exact value; this is as usually due to the singularity present in the domain. Similarly, we can see for the right mesh the order of convergence is approximately 1.4, but for Delaunay mesh it is varying because it is an unstructured mesh. In all cases we can see that the reduced model is at least 10 times faster.

In Table 10 we report the results of P2 FEM on the L-shaped domain. Here also, we have calculated results on the three types of mesh: crisscross, mixed, and Delaunay. In this case the order of convergence is 1.33 for the crisscross and mixed meshes and varying for the Delaunay mesh because it is a non-structured mesh. In this case the ROM is at least 13 times faster than the FOM, which can be observed from Table 11, where we have included the CPU time for FOM and ROM, respectively. In the case of the P2 FEM the eigenvalue is approximated correctly up to 3 decimal places.
Table 10
Comparison of first eigenvalue using ROM with three different mesh with P2 FEM On L-shaped Domain.

| Mesh     | n   | FOM             | ROM             | DOF | N  |
|----------|-----|-----------------|-----------------|-----|----|
| Crisscross | 8   | 9.654566452569  | 9.654566452611  | 1601| 5  |
|          | 16  | 9.645648130148(1.33) | 9.645648130148(1.33) | 6273| 6  |
|          | 32  | 9.642079379771(1.33) | 9.642079379771(1.33) | 24833| 6  |
|          | 64  | 9.640659153042(1.33) | 9.640659153042(1.33) | 98817| 7  |
|          | 128 | 9.64095077363(1.33)   | 9.64095077363(1.33) | 394241| 7  |
|          | 256 | 9.63971173962(1.33)   | 9.63971173962(1.33) | 1574913| 7  |
| Mixed    | 8   | 9.663207242395  | 9.663207242479  | 833 | 5  |
|          | 16  | 9.649100751840(1.32) | 9.649100751840(1.32) | 3201| 6  |
|          | 32  | 9.64359074163(1.33) | 9.64359074163(1.33) | 12545| 6  |
|          | 64  | 9.641208083352(1.33) | 9.641208083352(1.33) | 49665| 6  |
|          | 128 | 9.64031308825(1.33) | 9.64031308825(1.33) | 19763| 6  |
|          | 256 | 9.63957710191(1.33) | 9.63957710191(1.33) | 788481| 7  |
| Delaunay | 8   | 9.655492013317  | 9.655492013320  | 1385| 5  |
|          | 16  | 9.646108037396(1.30) | 9.646108037396(1.30) | 5293| 6  |
|          | 32  | 9.642596253225(1.15) | 9.642596253225(1.15) | 21157| 6  |
|          | 64  | 9.640617826930(1.68) | 9.640617826930(1.68) | 85109| 6  |
|          | 128 | 9.640070529727(1.37) | 9.640070529727(1.37) | 341577| 6  |
|          | 256 | 9.63986064681(1.10) | 9.63986064681(1.10) | 1352893| 7  |

Table 11
CPU time for FOM and ROM corresponding to the results of P2-FEM on L-shaped domain.

| Mesh     | n = 8 | n = 16 | n = 32 | n = 64 | n = 128 | n = 256 |
|----------|-------|--------|--------|--------|---------|---------|
| Crisscross | FOM   | 0.13   | 0.45   | 2.13   | 10.59   | 52.57   | 253.54  |
|          | ROM   | 0.01   | 0.03   | 0.14   | 0.70    | 3.30    | 17.69   |
| Mixed    | FOM   | 0.077  | 0.25   | 1.34   | 5.78    | 27.18   | 141.82  |
|          | ROM   | 0.007  | 0.02   | 0.07   | 0.32    | 1.47    | 6.55    |
| Delaunay | FOM   | 0.13   | 0.74   | 2.92   | 15.15   | 95.08   | 501.30  |
|          | ROM   | 0.01   | 0.03   | 0.15   | 0.69    | 3.60    | 24.35   |

Table 12
Comparison of first eigenvalue on ROM three different meshes with adaptive P2-FEM on L-shaped Domain.

| Mesh     | FOM | ROM | DOF | N  |
|----------|-----|-----|-----|----|
| Crisscross | 9.6397259448(1.64) | 9.6397259448(1.64) | 128876| 9  |
|          | 9.6397249480(2.15) | 9.6397249480(2.15) | 173946| 9  |
|          | 9.6397244634(1.76) | 9.6397244634(1.76) | 241486| 10 |
|          | 9.6397241783(1.57) | 9.6397241783(1.57) | 357343| 10 |
|          | 9.6397240778(1.14) | 9.6397240778(1.14) | 488670| 11 |
|          | 9.6397239925(1.57) | 9.6397239925(1.57) | 652305| 11 |
| Mixed    | 9.6397258141(1.92) | 9.6397258141(1.92) | 112184| 9  |
|          | 9.6397248627(1.63) | 9.6397248627(1.63) | 168046| 10 |
|          | 9.6397245461(1.16) | 9.6397245461(1.16) | 231542| 10 |
|          | 9.6397243056(1.44) | 9.6397243056(1.44) | 309858| 11 |
|          | 9.6397240956(1.89) | 9.6397240956(1.89) | 427413| 10 |
|          | 9.6397239677(1.83) | 9.6397239677(1.83) | 629926| 10 |
| Delaunay | 9.6397309233(1.84) | 9.6397309233(1.84) | 73219 | 8  |
|          | 9.6397272642(1.81) | 9.6397272642(1.81) | 109465| 8  |
|          | 9.6397255112 (1.83)| 9.6397255112 (1.83)| 162254| 8  |
|          | 9.6397247163(1.71) | 9.6397247163(1.71) | 236753| 9  |
|          | 9.6397243381(1.53) | 9.6397243381(1.53) | 343460| 10 |
|          | 9.6397241186 (1.65)| 9.6397241186 (1.65)| 490769| 11 |
Table 13
CPU time for FOM and ROM corresponding to the results of adaptive P2-FEM on L-shaped domain.

| Mesh       | Method | I     | II    | III   | IV    | V     | VI    |
|------------|--------|-------|-------|-------|-------|-------|-------|
| Crisscross | FOM    | 31.84 | 46.12 | 74.80 | 115.27| 163.06| 230.88|
|            | ROM    | 1.56  | 1.82  | 3.16  | 5.31  | 8.10  | 11.54 |
| Mixed      | FOM    | 26.25 | 42.02 | 67.98 | 98.71 | 137.83| 218.03|
|            | ROM    | 1.08  | 2.07  | 2.90  | 4.65  | 6.54  | 10.80 |
| Delaunay   | FOM    | 17.08 | 26.89 | 44.05 | 71.36 | 121.29| 177.53|
|            | ROM    | 0.70  | 1.16  | 1.75  | 2.79  | 5.55  | 8.85  |

In order to improve the approximation of the singular eigenfunction, we apply an adaptive scheme based on P2 FEM in order to calculate the solution of the full order model. Using the selected snapshots from the full order model we calculate the approximation of eigenvalues and eigenvectors with the reduced order model. For the adaptive method we adopt the usual strategy: Solve, Estimate, Mark, and Refine. The following residual based error estimator is used locally in each element $K$

$$
\eta_K^2 = h_K^2 \|r\|_0,K^2 + \frac{1}{2} \sum_{e \in K} h_e \|R|_0,e^2
$$

with

$$
r|_K = \Delta u_h + \lambda h u_h \quad \forall K \quad \text{and} \quad R|_e = \nabla u_h|_{K_1,e} \cdot n_{K_1,e} + \nabla u_h|_{K_2,e} \cdot n_{K_2,e}
$$

where $K_1,e$ and $K_2,e$ are the elements sharing the edge $e$ and $n_{K_1,e}$ and $n_{K_2,e}$ are the outward unit normal vectors on $\partial K_1,e$ and $\partial K_2,e$, respectively.

The results on the crisscross, Delaunay, and mixed meshes are shown in Table 12. We have reported the eigenvalues computed with the FOM in the second column, while the ROM eigenvalues are presented in the third column. In Table 13 we have reported the CPU time taken by FOM and ROM for all the results. In the final refinement we get the approximate eigenvalues equal to 9.6397239925 for both FOM and ROM on the crisscross mesh, which is quite a good result. In this case we get order of convergence equal to approximately 1.57. Also the ROM is much faster than the full order model because in FOM we need to calculate error, refine the mesh and solve the problem. But in ROM we have the mesh stored in FOM and we only solve the reduced problem.

Similarly, in the final refinement we get an approximate eigenvalues of about 9.6397241186 on the Delaunay mesh, which is correct up to 5 decimal digits. In this case the minimum order of convergence is 1.4. On the mixed uniform mesh the computed eigenvalue is 9.6397239677, which is correct up to 6 decimal digits and the minimum order of convergence is 1.4. In these two cases also the ROM is much faster than the full order model. In all the cases the number of reduced basis function used is nine or ten.

4.2. Results for the second eigenvalue using Technique I

In this subsection, we will present the results for the second eigenvalue using ROM on square domain and L-shaped domain with different mesh. Note that the second and third eigenvalues are same on the square domain in this case it is interesting to see how the eigenvectors look like for the ROM.

4.2.1. Results on the square domain

Here we are observing an interesting phenomena, that is the second eigenvectors corresponding to different triangular meshes are different. This is a well known behavior in presence of multiple eigenvalues. Indeed, the second and third eigenvalues of the Laplace problem are the same and correspond to a two dimensional eigenspace. Depending on the used mesh, the two computed eigenvectors may be different; in Fig. 4 the second eigenfunction is plotted for the Crisscross, Right, Left, and Delaunay mesh, respectively, on the square domain while in Fig. 5 the second eigenfunction is plotted for the Crisscross, Mixed, and Delaunay mesh, respectively, on the L-shaped domain. We showed the eigenvectors for P1-FEM, but similar phenomena have been observed for P2-FEM also.
In Table 14, we have included the second eigenvalue of the model problem on square domain and the solution in FOM is obtained using P1-FEM. We have considered three types of triangulation of the square domain, namely Crisscross, Right, and Delaunay mesh. Note that the eigenvalues converge to the exact eigenvalue 5 as we decrease the mesh size with rate of convergence 2. Similarly, in Table 15, we have shown the second eigenvectors of the model problem on square mesh and using P2-FEM for the full order model. In this case eigenvalues are converging to the exact eigenvalues with rate of convergence 4.

4.2.2. Results on the L-shaped domain

After getting success for the square domain we have tried to apply for the L-shaped domain. For the L-shaped domain we have used the exact second eigenvalue as 15.19725192 following the Ref. [47]. The results corresponding
to P1-FEM are shown in Table 16 and P2-FEM are shown in Table 17. For P1-FEM case the results are correct up to 3 decimal places and the order of convergence is 2 in all cases.

For the results of P2-FEM, the eigenvalues are matching with the exact one up to 6–7 decimal places. The rate of converges is decreasing when we are taking finer mesh. For the Crisscross mesh we got rate of convergence as 2.5 in the last case where the DOF is 1574913.

4.3. Results for the second eigenvalue using Technique II

In this subsection, we will present the results using the technique described in Section 3.2.2. Here we present the results for $p = 4$ that is we are interested in first four eigenvalues of the problem. For simplicity we will present
Table 14
Comparison for the second eigenvalue using ROM and P1-FEM FOM in the domain \((0, \pi)^2\). In the bracket we have indicated the rate of convergence.

| Mesh    | n  | FOM        | ROM          | DOF | N  |
|---------|----|------------|--------------|-----|----|
| Crisscross | 8  | 5.1650870182 | 5.1650875667 | 145 | 4  |
|         | 16 | 5.0408442412(2.1) | 5.0408442630(2.1) | 545 | 5  |
|         | 32 | 5.0101831478(2.1) | 5.0101831842(2.1) | 2113 | 5 |
|         | 64 | 5.0025440252(2.0) | 5.0025440253(2.0) | 8321 | 6  |
|         | 128 | 5.0006358959(2.0) | 5.0006358959(2.0) | 33025 | 7 |
|         | 256 | 5.0001589671(2.0) | 5.0001589671(2.0) | 131585 | 7 |
| Right   | 8  | 5.5005033551 | 5.3325135872 | 81  | 6  |
|         | 16 | 5.0858074725(2.8) | 5.0829176659(2.2) | 289 | 7  |
|         | 32 | 5.0207679900(2.1) | 5.0207206081(2.1) | 1089 | 7 |
|         | 64 | 5.0051804590(2.1) | 5.0051799162(2.1) | 4225 | 7  |
|         | 128 | 5.0012949110(2.0) | 5.0012949110(2.0) | 16641 | 7 |
|         | 256 | 5.0003237233(2.0) | 5.0003237233(2.0) | 66049 | 7 |
| Delaunay | 8  | 5.1779301074 | 5.1720770108 | 127 | 6  |
|         | 16 | 5.0449834439(2.1) | 5.0446649292(2.1) | 467 | 7  |
|         | 32 | 5.0114651017(2.0) | 5.0114033507(2.0) | 1784 | 7 |
|         | 64 | 5.0028582554(2.0) | 5.0028582566(2.0) | 7140 | 7  |
|         | 128 | 5.0007046487(2.0) | 5.0007046487(2.0) | 28523 | 7 |
|         | 256 | 5.001751808(2.0) | 5.001751808(2.0) | 113846 | 7 |

Table 15
Comparison for the second eigenvalue using ROM and P2-FEM FOM in the domain \((0, \pi)^2\). In the bracket we have indicated the rate of convergence.

| Mesh    | n  | FOM        | ROM          | DOF | N  |
|---------|----|------------|--------------|-----|----|
| Crisscross | 8  | 5.0019772105 | 5.0019773172 | 545 | 5  |
|         | 16 | 5.0001272082(4.1) | 5.0001272097(4.1) | 2113 | 6  |
|         | 32 | 5.0000080111(4.0) | 5.0000080134(4.0) | 8321 | 6  |
|         | 64 | 5.0000005017(4.0) | 5.0000005017(4.0) | 33025 | 7 |
|         | 128 | 5.0000000314(4.0) | 5.0000000314(4.0) | 131585 | 7 |
|         | 256 | 5.0000000020(4.0) | 5.0000000020(4.0) | 525313 | 7 |
|         | 512 | 5.0000000001(4.0) | 5.0000000001(4.0) | 2099201 | 8 |
| Right   | 8  | 5.0072625371 | 5.0072736921 | 289 | 5  |
|         | 16 | 5.0004443122(4.2) | 5.0004443193(4.2) | 1089 | 5  |
|         | 32 | 5.0000280160(4.1) | 5.0000280184(4.1) | 4225 | 5  |
|         | 64 | 5.0000017569(4.0) | 5.0000017569(4.0) | 16641 | 6 |
|         | 128 | 5.0000001099(4.0) | 5.0000001099(4.0) | 66049 | 6  |
|         | 256 | 5.0000000069(4.0) | 5.0000000069(4.0) | 263169 | 6  |
|         | 512 | 5.0000000004(4.0) | 5.0000000004(4.0) | 1050625 | 7 |
| Delaunay | 8  | 5.0023834056 | 5.0023837583 | 471 | 5  |
|         | 16 | 5.0001565518(4.1) | 5.0001565543(4.1) | 1799 | 5  |
|         | 32 | 5.0000100102(4.1) | 5.0000100119 (4.1) | 7003 | 5  |
|         | 64 | 5.0000006117(4.0) | 5.0000006118(4.0) | 28299 | 5  |
|         | 128 | 5.000000374(4.0) | 5.000000374(4.0) | 113575 | 6 |

the result for crisscross mesh only. In Table 18, we have presented the first four eigenvalues of the EVP on square domain \((0, \pi)^2\) using P1-FEM in the FOM and the simultaneous technique for the ROM. We have included the CPU time for both FOM and ROM. Note that for ROM the CPU time is the time taken after getting the matrix \(V_{1:4}\). The rate of convergence is 2 for all the four eigenvalues in this case.

In Table 19, we have presented the first four eigenvalues of the EVP on L-shaped domain using P1-FEM in the FOM and the simultaneous technique for the ROM. We have included the CPU time for both FOM and ROM. For the first eigenvector the rate of convergence are 1.6,1.5,1.4,1.4,1.4 respectively. For the second, third and fourth
Table 16
Comparison for the second eigenvalue using ROM and P1-FEM FOM on L-shaped domain. In the bracket we have indicated the rate of convergence.

| Mesh   | n  | FOM          | ROM          | N  |
|--------|----|--------------|--------------|----|
| Crisscross | 8  | 15.3425275459 | 15.3425275459 | 5  |
|        | 16 | 15.2338080549(2.1) | 15.2338080549(2.1) | 6  |
|        | 32 | 15.2064356395(2.0) | 15.2064356395(2.0) | 6  |
|        | 64 | 15.195554505(2.0)  | 15.195554505(2.0)  | 7  |
|        | 128| 15.1978290422(0.0) | 15.1978290422(0.0) | 7  |
|        | 256| 15.1973964024(2.0) | 15.1973964024(2.0) | 7  |
| Mixed  | 8  | 15.5786633739 | 15.5786633739 | 5  |
|        | 16 | 15.2936973793(2.1) | 15.2936973794(2.1) | 5  |
|        | 32 | 15.2215076782(2.1) | 15.2215076782(2.1) | 5  |
|        | 64 | 15.203369582(0.0)  | 15.203369582(0.0)  | 6  |
|        | 128| 15.1987764145(2.0) | 15.1987764145(2.0) | 6  |
|        | 256| 15.197633552(2.0)  | 15.197633552(2.0)  | 7  |
| Delaunay | 8  | 15.3597242210 | 15.3597242210 | 4  |
|        | 16 | 15.2401433624(2.0) | 15.2401433625(2.0) | 5  |
|        | 32 | 15.2079387198(2.0) | 15.2079387198(2.0) | 5  |
|        | 64 | 15.199806229(0.0)  | 15.199806229(0.0)  | 5  |
|        | 128| 15.1979034628(2.0) | 15.1979034628(2.0) | 6  |

Table 17
Comparison for the second eigenvalue using ROM and P2-FEM FOM on L-shaped domain. In the bracket we have indicated the rate of convergence.

| Mesh   | n  | FOM          | ROM          | N  |
|--------|----|--------------|--------------|----|
| Crisscross | 8  | 15.1982228490 | 15.1982228826 | 5  |
|        | 16 | 15.1973481522(3.4) | 15.1973481958(3.4) | 5  |
|        | 32 | 15.1972653485(3.1) | 15.1972653485(3.1) | 6  |
|        | 64 | 15.1972535377(2.9) | 15.1972535377(2.9) | 6  |
|        | 128| 15.1972521666(2.7) | 15.1972521666(2.7) | 6  |
|        | 256| 15.1972519634(2.5) | 15.1972519634(2.5) | 7  |
| Mixed  | 8  | 15.2012085976 | 15.2012086022 | 4  |
|        | 16 | 15.1975932373(3.6) | 15.1975932448(3.6) | 4  |
|        | 32 | 15.1972875040(3.3) | 15.1972875121(3.3) | 4  |
|        | 64 | 15.1972563831(3.0) | 15.1972563831(3.0) | 5  |
|        | 128| 15.1972525565(2.8) | 15.1972525565(2.8) | 5  |
|        | 256| 15.1972520212(2.7) | 15.1972520212(2.7) | 6  |
| Delaunay | 8  | 15.1983777570 | 15.1983777847 | 4  |
|        | 16 | 15.1973805547(3.3) | 15.1973807136(3.3) | 4  |
|        | 32 | 15.1972685127(3.0) | 15.1972685197(3.0) | 4  |
|        | 64 | 15.1972537133(3.2) | 15.1972537144(3.2) | 5  |
|        | 128| 15.1972521906(2.7) | 15.1972521905(2.7) | 6  |

eigenvalues the rate of convergence is 2. To calculate the rate of convergence we assume that the exact values are 9.63972384, 15.19725192, 19.739208802178 and 29.52148111 respectively for the L-shaped domain [47].

4.4. Eigenvalues for parametric eigenvalue problems

In this section we show how the method that we have presented so far can be extended to parametric eigenvalue problems.

Let us consider the following parameter dependent eigenvalue problem:

\[
\begin{align*}
-\Delta u_{xx}(\mu) - (1 + \mu)u_{yy}(\mu) &= \lambda(\mu)u(\mu) & \text{in } \Omega = (-1, 1)^2 \\
\frac{\partial u(\mu)}{\partial n} &= 0 & \text{on } \partial \Omega
\end{align*}
\]  (27)
Table 18  
Comparison for the first four eigenvalues using ROM and FOM with P1-FEM on square mesh of type crisscross.  

| n   | Method | 1st            | 2nd            | 3rd            | 4th            | Time   |
|-----|--------|----------------|----------------|----------------|----------------|--------|
| 8   | FOM    | 2.0215680546   | 5.1650870182   | 5.1650870182   | 8.3520743438   | 0.09   |
|     | ROM    | 2.0215680601   | 5.1650870183   | 5.1650870198   | 8.3520743441   | 0.04   |
| 16  | FOM    | 2.0053639950   | 5.0408442412   | 5.0408442412   | 8.0862722183   | 0.20   |
|     | ROM    | 2.0053639951   | 5.0408442412   | 5.0408442412   | 8.0862722183   | 0.01   |
| 32  | FOM    | 2.0013392384   | 5.0101831478   | 5.0101831478   | 8.0214559802   | 0.69   |
|     | ROM    | 2.0013392386   | 5.0101831478   | 5.0101831478   | 8.0214559802   | 0.02   |
| 64  | FOM    | 2.0003346994   | 5.0025440252   | 5.0025440252   | 8.0053569534   | 4.76   |
|     | ROM    | 2.0003346997   | 5.0025440252   | 5.0025440252   | 8.0053569534   | 0.02   |
| 128 | FOM    | 2.0000836680   | 5.0006358959   | 5.0006358959   | 8.0013387977   | 19.65  |
|     | ROM    | 2.0000836684   | 5.0006358959   | 5.0006358959   | 8.0013387977   | 0.02   |
| 256 | FOM    | 2.0000209166   | 5.0001589671   | 5.0001589671   | 8.0003346719   | 93.55  |
|     | ROM    | 2.0000209167   | 5.0001589671   | 5.0001589671   | 8.0003346719   | 0.06   |

Table 19  
Comparison for the first four eigenvalues using ROM and FOM with P1-FEM on L-shaped domain with crisscross mesh.  

| n   | Method | 1st            | 2nd            | 3rd            | 4th            | Time   |
|-----|--------|----------------|----------------|----------------|----------------|--------|
| 8   | FOM    | 9.7894610546   | 15.3425275459  | 19.9520769684  | 30.0729173795  | 0.12   |
|     | ROM    | 9.7894610598   | 15.3425275462  | 19.9520769805  | 30.0729173796  | 0.03   |
| 16  | FOM    | 9.6895509095   | 15.2338080549  | 19.7921493113  | 29.6586083579  | 0.46   |
|     | ROM    | 9.6895509319   | 15.2338080549  | 19.7921493114  | 29.6586083579  | 0.01   |
| 32  | FOM    | 9.6571939684   | 15.2064356395  | 19.7524265549  | 29.5557267179  | 2.04   |
|     | ROM    | 9.6571939744   | 15.2064356395  | 19.7524265551  | 29.5557267179  | 0.02   |
| 64  | FOM    | 9.6460897396   | 15.1995554505  | 19.7425121531  | 29.5300424160  | 12.63  |
|     | ROM    | 9.6460897889   | 15.1995554505  | 19.7425121531  | 29.5300424160  | 0.02   |
| 128 | FOM    | 9.6421091997   | 15.1978290422  | 19.7400345719  | 29.5236217214  | 63.02  |
|     | ROM    | 9.6421092739   | 15.1978290422  | 19.7400345719  | 29.5236217214  | 0.06   |
| 256 | FOM    | 9.6406353135   | 15.1973964024  | 19.7394152404  | 29.5220163313  | 322    |
|     | ROM    | 9.6406353254   | 15.1973964024  | 19.7394152404  | 29.5220163313  | 0.22   |

where the parameter $\mu \in [0, 5]$ and $\mathbf{x} = (x, y) \in \Omega$. The analytical eigensolutions to (27) can be explicitly computed by separation of variables, and are given by:

$$
\lambda_{n,m}(\mu) = \frac{\pi^2}{4} (m^2 + (1 + \mu)n^2),
$$

$$
\psi_{n,m} = \cos\left(\frac{m \pi}{2} x\right) \cos\left(\frac{n \pi}{2} y\right), \quad \forall m, n \in \mathbb{N}.
$$

(28)

Introducing the fictitious time parameter and proceeding as for the Laplace eigenvalue problem, we will get the matrix form of our time continuation technique as:

$$
AU^{k+1}(\mu) + \mu A_1 U^{k+1}(\mu) + \frac{1}{\Delta t} MU^{k+1}(\mu) = \left(\lambda^k_{h}(\mu) + \frac{1}{\Delta t}\right) MU^k(\mu),
$$

(29)

where $A, M$ are the same as defined in (12) and

$$(A_1)_{i,j} = \int_{\Omega} \frac{\partial \varphi_i(x)}{\partial y} \frac{\partial \varphi_j(x)}{\partial y} \, dx.
$$

(30)

First, we will select few parameters from the parameter space and for each sample parameter we solve the time continuation problem and use all the solutions to form the “offline” snapshot matrix. In this case the word “offline” is used in a more traditional and conventional way than before. in our computations we have selected only three sample parameters, $\mu = 0, 2, 4$. We have calculated two eigenvalues using Technique II in the “online” phase.
Table 20
The sample parameters used in offline phase are $\mu = 0, 2, 4$. The test values of $\mu$ are 0.5, 1.5, 3.5, 4.5 in online phase. Here also we have used the snapshot matrix of type $S_4$.

| Mesh     | n   | eigenvalue | 0.5         | 1.5         | 3.5         | 4.5         |
|----------|-----|------------|-------------|-------------|-------------|-------------|
| Crisscross | 16  | 1st        | 6.18504667  | 8.65906533  | 13.60710266 | 16.08112133 |
|          |     | 2nd        | 13.68898717 | 16.17787528 | 21.15478158 | 23.64306346 |
|          | 32  | 1st        | 6.17263330  | 8.64168662  | 13.57979326 | 16.04884658 |
|          |     | 2nd        | 13.60017343 | 16.07291137 | 21.01811966 | 23.49067075 |
|          | 64  | 1st        | 66.16953505 | 8.63734907  | 13.57297711 | 16.04079112 |
|          |     | 2nd        | 13.57808896 | 16.04680401 | 20.98419952 | 23.45289242 |
|          | 128 | 1st        | 6.16876080  | 8.63626513  | 13.57127377 | 16.03877809 |
|          |     | 2nd        | 13.57250056 | 16.04207201 | 20.97572699 | 23.44343906 |
| Delaunay | 16  | 1st        | 6.19163596  | 8.66824148  | 13.62139201 | 16.09795172 |
|          |     | 2nd        | 13.69753052 | 16.19901655 | 21.20185539 | 23.70324225 |
|          | 32  | 1st        | 6.17412519  | 8.64373431  | 13.58296829 | 16.05257802 |
|          |     | 2nd        | 13.60389644 | 16.07960068 | 21.03094359 | 23.50659956 |
|          | 64  | 1st        | 6.16990826  | 8.63786340  | 13.57377251 | 16.04172673 |
|          |     | 2nd        | 13.57884091 | 16.04831239 | 20.98722709 | 23.45667591 |
|          | 128 | 1st        | 6.16884961  | 8.63638780  | 13.57146197 | 16.03898854 |
|          |     | 2nd        | 13.57271520 | 16.04062627 | 20.97644411 | 23.44435195 |
|          | 256 | 1st        | 6.16858908  | 8.63602425  | 13.57089403 | 16.03832878 |
|          |     | 2nd        | 13.57120317 | 16.03873085 | 20.97378395 | 23.44130988 |

Table 21
CPU time for FOM and ROM corresponding to the parametric EVP.

| Mesh     | Method | n = 16   | n = 32   | n = 64   | n = 128  | n = 256  |
|----------|--------|----------|----------|----------|----------|----------|
| Crisscross | FOM    | 0.04     | 0.16     | 0.87     | 3.89     | 18.92    |
|          | ROM    | 0.005    | 0.017    | 0.003    | 0.012    | 0.027    |
| Delaunay | FOM    | 0.04     | 0.19     | 1.05     | 4.96     | 26.12    |
|          | ROM    | 0.006    | 0.019    | 0.004    | 0.014    | 0.033    |

The results obtained in the “online” phase are converging to the exact ones when the mesh is refined. It is interesting to see that the use of the time continuation technique allows a more flexible choice of the sample parameters for the “offline” phase than if we had proceeded without it; in this case only three sample parameters are enough to guarantee a good convergence of the ROM solution. This aspect will be the object of further investigations and comparisons. We report the CPU time for online phase/ROM and offline phase/FOM for this problem in Table 21. The time taken to solve the time continuation problem for one sample parameter in offline phase has been reported as CPU time in FOM similarly for the online phase. Form the table one can also see that the ROM is much faster than the FOM.

Declaration of competing interest

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

Data availability

No data was used for the research described in the article.
Acknowledgments

This research was supported by the Competitive Research Grants Program CRG2020 “Synthetic data-driven model reduction methods for modal analysis” awarded by the King Abdullah University of Science and Technology (KAUST). Daniele Boffi is member of the INdAM Research group GNCS and his research is partially supported by IMATI/CNR and by PRIN/MIUR.

References

[1] G. Strang, J.J. Fix, An Analysis of the Finite Element Method, Prentice-Hall, Englewood Cliffs N.J, 1973.

[2] I. Babuška, J. Osborn, Eigenvalue Problems, Handbook of Numerical Analysis, vol. II, North-Holland, Amsterdam, 1991, pp. 641–787.

[3] D. Boffi, Finite element approximation of eigenvalue problems, Acta Numer. 19 (2010) 1–120.

[4] D. Boffi, F. Brezzi, M. Fortin, Mixed Finite Element Methods and Applications, Springer Series in Computational Mathematics, 44, Springer, Heidelberg, 2013.

[5] D. Boffi, F. Brezzi, L. Gastaldi, On the convergence of eigenvalues for mixed formulations, Ann. Sc. Norm. Super Pisa Cl. Sci. 25 (1–2) (1997) 131–154.

[6] D. Boffi, F. Brezzi, L. Gastaldi, On the problem of spurious eigenvalues in the approximation of linear elliptic problems in mixed form, Math. Comp. 69 (229) (2000) 121–140.

[7] B.O. Almroth, P. Stern, F.A. Brogan, Automatic choice of global shape functions in structural analysis, AIAA J. 16 (5) (1978) 525–528.

[8] J.P. Fink, C.W. Rheinboldt, On the error behavior of the reduced basis technique for nonlinear finite element approximations, Z. Angew. Math. Mech. 63 (1) (1983) 21–28.

[9] A.K. Noor, J.M. Peters, Reduced basis technique for nonlinear analysis of structures, AIAA J. 18 (4) (1980) 455–462.

[10] T.A. Porsching, Estimation of the error in the reduced basis method solution of nonlinear equations, Math. Comp. 45 (172) (1985) 487–496.

[11] J.S. Peterson, The reduced basis method for incompressible viscous flow calculations, SIAM J. Sci. Stat. Comput. 10 (4) (1989) 777–786.

[12] K. Ito, S.S. Ravindran, A reduced-order method for simulation and control of fluid flows, J. Comput. Phys. 143 (2) (1998) 403–425.

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

[14] K. Kunisch, S. Volkwein, Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics, SIAM J. Numer. Anal. 40 (2) (2002) 492–515.

[15] C. Prud’homme, D.V. Rovas, K. Veroy, A.T. Patera, A mathematical and computational framework for reliable real-time solution of parametrized partial differential equations, ESAIM Math. Model. Numer. Anal. 36 (5) (2002) 747–771.

[16] C. Prud’homme, D.V. Rovas, K. Veroy, L. Machiels, Y. Maday, A.T. Patera, G. Turinici, Reliable real-time solution of parametrized partial differential equations: Reduced-basis output bound methods, J. Fluids Eng. 124 (1) (2002) 70–80.

[17] K. Willcox, J. Peraire, Balanced model reduction via the proper orthogonal decomposition, AIAA J. 40 (11) (2002) 2323–2330.

[18] K. Veroy, C. Prud’homme, D.V. Rovas, A.T. Patera, A posteriori error bounds for reduced-basis approximations of parametrized noncoercive and nonlinear elliptic partial differential equations, in: 16th AIAA Computational Fluid Dynamics Conference, 2003.

[19] P. Brenner, V. Mehrmann, D. Sorensen, Dimension reduction of large-scale systems, Lecture notes in computational science and engineering, Springer, Heidelberg, 2003.

[20] N.C. Nguyen, K. Veroy, A.T. Patera, Certified real-time solution of parametrized partial differential equations, in: Yip S(ed) Handbook of materials modeling, Springer, Berlin, 2005.

[21] D.B.P. Huynh, G. Rozza, S. Sen, A.T. Patera, A successive constraint linear optimization method for lower bounds of parametric coercivity and inf–sup stability constants, C. R. Math. 345 (8) (2007) 473–478.

[22] A. Quarteroni, G. Rozza, Numerical solution of parametrized Navier-Stokes equations by reduced basis methods, Numer. Methods Partial Differential Equations 23 (4) (2007) 923–948.

[23] G. Rozza, D.B.P. Huynh, A.T. Patera, Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations, Arch. Comput. Methods Eng. 15 (2008) 229–275.

[24] Y. Chen, J. Hesthaven, Y. Maday, J. Rodríguez, A monotonic evaluation of lower bounds for inf-sup stability constants in the frame of reduced basis approximations, C. R. Math. 346 (23) (2008).

[25] B. Haasdonk, M. Ohlberger, Reduced basis method for finite volume approximations of parametrized linear evolution equations, ESAIM Math. Model. Numer. Anal. 42 (2) (2008) 277–302.

[26] C. Canuto, T. Tonn, K. Urban, A posteriori error analysis of the reduced basis method for nonaffine parametrized nonlinear PDEs, SIAM J. Numer. Anal. 47 (3) (2009) 2001–2022.

[27] J.V. Aguado, A. Huerta, F. Chinesta, E. Cueto, Real-time monitoring of thermal processes by reduced-order modeling, Internat. J. Numer. Methods Engrg. 102 (5) (2015) 991–1017.

[28] J. Hesthaven, G. Rozza, B. Stamm, Certified Reduced Basis Methods for Parametrized Partial Differential Equations, Springer International Publishing, Bilbao, 2016.

[29] M. Lyly, C. Lovadina, R. Stenberg, A posteriori estimates for the Stokes eigenvalue problem, Numer. Methods Partial Differential Equations 25 (2009) 244–257.

[30] L. Olson, T. Vendini, Eigenproblems from finite element analysis of fluid–structure interactions, Comput. Struct. 33 (3) (1989) 679–687.
[31] S.J. Cooke, B. Levush, Eigenmode solution of 2-D and 3-D electromagnetic cavities containing absorbing materials using the Jacobi–Davidson algorithm, J. Comput. Phys. 157 (2000) 350–370.

[32] A.G. Buchan, Christopher Pain, Ionel Navon, A POD reduced order model for eigenvalue problems with application to reactor physics, Internat. J. Numer. Methods Engrg. 95 (12) (2013) 1011–1032.

[33] S. Giani, I.G. Graham, Adaptive finite element methods for computing band gaps in photonic crystals, Numer. Math. 121 (1) (2012) 31–64.

[34] Y. Maday, A.T. Patera, J. Peraire, A general formulation for a posteriori bounds for output functionals of partial differential equations; application to the eigenvalue problem, C. R. Acad. Sci. I 328 (9) (1999) 823–828.

[35] L. Machiels, Y. Maday, I.B. Oliveira, A.T. Patera, D.V. Rovas, Output bounds for reduced-basis approximations of symmetric positive definite eigenvalue problems, C. R. Acad. Sci. I 331 (2) (2000) 153–158.

[36] G.S.H. Pau, Reduced Basis Method for Quantum Models of Crystalline Solids, (Ph.D. thesis), Massachusetts Institute of Technology, 2008.

[37] G.S.H. Pau, Reduced-basis method for band structure calculations, Phys. Rev. E 76 (2007) 046704.

[38] G.S.H. Pau, Reduced basis method for simulation of nanodevices, Phys. Rev. B 78 (2008) 155425.

[39] I. Fumagalli, A. Manzoni, N. Parolini, M. Verani, Reduced basis approximation and a posteriori error estimates for parametrized elliptic eigenvalue problems, ESAIM Math. Model. Numer. Anal. 50 (6) (2016) 1857–1885.

[40] T. Horger, B. Wohlmuth, T. Dickopf, Simultaneous reduced basis approximation of parameterized elliptic eigenvalue problems, ESAIM: M2AN 51 (2) (2017) 443–465.

[41] P. German, J.C. Ragusa, Reduced-order modeling of parameterized multi-group diffusion k-eigenvalue problems, Ann. Nucl. Energy 134 (2019) 144–157.

[42] A. Quarteroni, A. Manzoni, F. Negri, Reduced Basis Methods for Partial Differential Equations: An Introduction, Springer, 2016, pp. 1–263.

[43] Y. Maday, A.T. Patera, G. Turinici, A priori convergence theory for reduced-basis approximations of single-parameter elliptic partial differential equations, J. Sci. Comput. 17 (2002) 437–446.

[44] Y. Maday, A.T. Patera, G. Turinici, Global a priori convergence theory for reduced-basis approximation of single-parameter symmetric coercive elliptic partial differential equations, C. R. Math 335 (2002) 289–294.

[45] M.G. Larson, F. Bengzon, The Finite Element Method: Theory, Implementation, and Applications, Springer, Heidelberg, 2013.

[46] S.A. Funken, A. Schmidt, Adaptive mesh refinement in 2D – an efficient implementation in matlab, Comput. Methods Appl. Math. 20 (3) (2020) 459–479.

[47] Quan Yuan, Zhiqing He, Bounds to eigenvalues of the Laplacian on L-shaped domain by variational methods, J. Comput. Appl. Math. 233 (4) (2009) 1083–1090.