MODELLISTICA E CALCOLO SCIENTIFICO

MOX

POLITECNICO MILANO

MODELING AND SCIENTIFIC COMPUTING

MOX-Report No. 14/2016

A matrix discrete empirical interpolation method for the efficient model reduction of parametrized nonlinear PDEs: application to nonlinear elasticity problems

Bonomi, D.; Manzoni, A.; Quarteroni, A.

A matrix discrete empirical interpolation method for the efficient model reduction of parametrized nonlinear PDEs: application to nonlinear elasticity problems[*]

Diana Bonomi[†]   Andrea Manzoni[‡]   Alfio Quarteroni[‡]

Abstract

When using Newton iterations to solve nonlinear parametrized PDEs in the context of Reduced Basis (RB) methods, the assembling of the RB arrays in the online stage depends in principle on the high-fidelity approximation. This task is even more challenging when dealing with fully nonlinear problems, for which the global Jacobian matrix has to be entirely reassembled at each Newton step. In this paper the Discrete Empirical Interpolation Method (DEIM) and its matrix version MDEIM are exploited to perform system approximation at a purely algebraic level, in order to evaluate both the residual vector and the Jacobian matrix very efficiently. We compare different ways to combine solution-space reduction and system approximation, and we derive a posteriori error estimates on the solution accounting for the contribution of DEIM/MDEIM errors. The capability of the proposed approach to generate accurate and efficient reduced-order models is demonstrated on the solution of two nonlinear elasticity problems.

Keywords: Reduced Basis Method; Proper Orthogonal Decomposition; Empirical Interpolation Method; Nonlinear Elasticity; Computational Mechanics.

1   Introduction

The rapid numerical solution of partial differential equations (PDEs) is of paramount importance in PDE-constrained optimization, multi-scenario and/or real-time analysis, uncertainty quantification. In all these contexts, reduced order modeling (ROM) techniques have proven to be reliable alternatives to traditional high-fidelity techniques – such as the finite element method – providing accurate solutions at a greatly reduced cost. This is usually achieved by seeking the solution in a subspace of much smaller dimension N than the original finite dimensional space employed by a full-order model (FOM), typically featuring a dimension N_h of several thousands, or even millions, of degrees of freedom depending on the application at hand. Very often, this is required for parameter-dependent PDEs, namely, by considering varying inputs to the PDE problem, such as initial and/or boundary conditions, physical coefficients or sources. The ability of a ROM to provide the approximate solution to a PDE relies on *(i)* global, low dimensional spaces built over a set of snapshots, that is, FOM solutions computed for a set of parameter inputs, and *(ii)* (Petrov)-Galerkin projection to construct the low-dimensional $N \times N$ problem providing the ROM solution. In the case of parametrized PDEs, the reduced basis (RB) method has been extensively used to perform such a *solution-space reduction*, relying on greedy algorithms or proper orthogonal decomposition to generate the reduced space.

[†]MOX, Dipartimento di Matematica, Politecnico di Milano, P.za Leonardo da Vinci 32, I-20133 Milano, Italy, `diana.bonomi@polimi.it`

[‡]CMCS-MATHICSE-SB, Ecole Polytechnique Fédérale de Lausanne, Station 8, CH-1015 Lausanne, Switzerland, `andrea.manzoni@epfl.ch` (corresponding author), `alfio.quarteroni@epfl.ch`

Being able to assemble and solve the reduced problem almost inexpensively is possible if the ROM arrays are independent of the FOM dimension N_h; in this case, we exploit an Offline-Online decomposition, since for any input parameter the ROM problem can be assembled in the online phase by combining (possibly few) N-dimensional arrays stored during the offline phase – this is nowadays a standard procedure when dealing with linear PDEs showing an affine parametric dependence. In the case of more complex, nonaffine parametric dependence, constructing the ROM for any new parameter would require to first assemble the FOM arrays, then to project them onto the reduced space, thus entailing a computational complexity still of order N_h. To avoid this, an affine approximation of the linear operators, which we can refer to as *system approximation*, has to be computed during the offline phase. These two operations are usually performed sequentially in the case of steady problems, or simultaneously in the case of time-dependent problems.

The empirical interpolation method (EIM) has been originally proposed in [4, 24] to approximate nonaffinely parametrized functions, as well as tensor components, appearing in full-order operators, prior to reduction; see, e.g., [18, 26] for applications in the framework of RB methods. The discrete variant of EIM, the so-called Discrete Empirical Interpolation Method (DEIM), was originally developed in [11] to efficiently deal with nonlinear problems, but has also been applied to nonaffinely parametrized linear operators [2]. Nevertheless, very often an expensive pre-processing is required in order to obtain a version of parametrized operators which EIM or DEIM can be applied to. More recently, a *matrix version* of DEIM (MDEIM) has been developed [10] to approximate the full-order parametrized operators in a purely algebraic way, independently of the way the parametrized operators are generated, thus avoiding to act on parametrized functions and directly employing parameter-dependent matrices. This technique has been employed in [30] to address the efficient reduction of nonaffinely parametrized steady and time-dependent PDEs, with applications to PDE-constrained optimization and coupled problems.

In the case of parametrized nonlinear PDEs, performing system approximation is mandatory in order to assemble and solve at each step of a Newton method the reduced-order problem independently of the dimension N_h of the underlying FOM. First applications dealing with semilinear PDEs, as well as nonlinear PDEs featuring low-order polynomial nonlinearities, have exploited EIM and DEIM to deal with nonlinear terms, see e.g. [18]; more recent applications can be found, e.g., in [43, 41, 29]. A slightly different alternative, the so-called *best point interpolation* method, has been addressed in [31]. Note that in the case of nonlinear affine PDEs that are at most quadratically nonlinear in the state variable – such as in the case of Navier-Stokes equations – an efficient ROM can be obtained without relying necessarily on EIM/DEIM, even if this implies decomposing nonlinear terms into the sum of $O(N^2)$ terms, see e.g. [25].

In the case of higher-order, or nonpolynomial, nonlinearities, system approximation usually goes under the name of *hyper-reduction* and several techniques have been proposed to recover an approximate affine structure of nonlinear terms: besides EIM and DEIM, missing point estimation [3] and gappy POD [15] have also been developed, e.g. within the so-called Gauss-Newton with approximated tensors (GNAT) method, see e.g. [8].

Extending a recent methodology proposed in [30] for the efficient system reduction of parametrized linear PDEs, in this paper we show how to take advantage of both DEIM and matrix DEIM to perform, at a purely algebraic level, hyper-reduction of three-dimensional, fully nonlinear problems arising in computational mechanics. Matrix DEIM (MDEIM) has been recently proposed in [9, 42, 10] to directly approximate the full operator instead than its nonlinear terms, as done in the case of DEIM [11]. Indeed, although a DEIM approximation of vector arrays is a standard technique in the framework of nonlinear reduced-order modeling, MDEIM for the approximation of Jacobian matrices has only been exploited in a few applications so far, see e.g. [40]. Our proposed DEIM/MDEIM framework thus allows to approximate all the structures arising from the discretization of nonlinear parametrized PDEs and can be combined with RB methods where reduced spaces are obtained thanks to either greedy or POD techniques. We provide a detailed comparison of different strategies obtained by interplaying DEIM/MDEIM system approximation and POD solution-space reduction and we propose a new snapshots selection method based on the introduction of a semi hyper-reduced problem, able to provide a great reduction of the computational time.

Moreover, we develop a computable, residual-based a posteriori error bound that accounts for the DEIM/MDEIM approximation errors. Such a POD-DEIM-MDEIM framework can in principle be applied to any parametrized nonlinear PDE. Here we consider, for the sake of numerical examples, two problems arising in nonlinear mechanics: a shear test on a Saint-Venant-Kirchoff material, and a coupled electro-mechanical problem simulating (a simplified version of) cardiac contraction. This latter can be seen as a first proof of concept of our proposed strategy, whose application to more complex geometries within a time-dependent, electro-mechanical framework will be the object of a forthcoming publication.

We point out that, differently than fluid dynamics, fewer problems in structural mechanics have been tackled by means of reduced order models, due to their intrinsic complexity when dealing with involved nonlinear constitutive laws. We mention former contributions dealing with linear elasticity problems (see, e.g. [21, 27, 1]) and nonlinear elasticity of two-dimensional structures, featuring e.g. polynomial nonlinearities [44], or homogenization techniques [44]. As for three dimensional problems, preliminary applications of model order reduction to structural mechanics can be found in [23], where the focus is on time dependent problems, but no hyper-reduction techniques are considered. More recent contributions to nonlinear three-dimensional structural problems can be found e.g. in [7, 22]. In [36] a POD-DEIM method has been recently applied to nonlinear structural problems, however dealing with an approximation of the Jacobian matrix which still depends on the high-fidelity problem dimension. In this work we exploit a MDEIM approximation strategy to approximate the Jacobian matrix too, which is able to significantly reduce the computational time, still retaining the accuracy of the FOM solution.

The structure of the paper is as follows. In Section 2 we introduce the RB methodology applied to parametrized nonlinear problems. In section 3 we recall the POD technique, used to construct the reduced space. In section 4 we first introduce the classical DEIM approach suitable for the approximation of the residual vectors; then we show how the MDEIM technique can be extended to nonlinear parametrized problems. Details of our reduced framework are then illustrated and different strategies to obtain accurate DEIM basis are proposed. In the final part of the section we also show the algorithms related to the Offline and the Online stages of the reduced order model presented. A new reliable a posteriori error bound is provided in section 5. In section 6 we present two different applications to nonlinear mechanical problems, in order to assess the computational performance of the proposed strategy. Conclusions are drawn in section 8.

2   RB methods for nonlinear parametrized PDEs

In this work we aim to develop a reliable and efficient ROM for stationary nonlinear parametrized problems: find $u(\boldsymbol{\mu}) \in V$ such that

$$R(\mathbf{u}(\boldsymbol{\mu}); \boldsymbol{\mu}) = 0 \qquad \text{in } V', \tag{2.1}$$

being $V = V(\Omega_0)$ a suitable Hilbert space and V' its dual; $\Omega_0 \subset \mathbb{R}^n$, $n = 2, 3$ is the (reference) spatial domain. Here we denote by $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_d) \in \mathcal{D}$ a parameter vector and by $\mathcal{D} \subset \mathbb{R}^d$ the parameter domain. To solve problem (2.1) we rely on the Newton method, yielding at each step a problem under the form

$$J(\hat{\mathbf{u}}(\boldsymbol{\mu}); \boldsymbol{\mu})[\delta\mathbf{u}(\boldsymbol{\mu})] = -R(\hat{\mathbf{u}}(\boldsymbol{\mu}); \boldsymbol{\mu}),$$

arising from the linearization of (2.1) around $\hat{\mathbf{u}}$ (the solution at the previous iteration). Here, $J(\hat{\mathbf{u}}(\boldsymbol{\mu}); \boldsymbol{\mu}) = DR(\hat{\mathbf{u}}(\boldsymbol{\mu}); \boldsymbol{\mu})$ denotes the Fréchet differential of R evaluated at $\hat{\mathbf{u}}$. Our FOM for approximating the solution of (2.1) is based on the finite element method. We introduce a finite-dimensional approximation space $V_h \subset V$,

$$V_h = X_h^r \cup V, \qquad X_h^r = \{\mathbf{v}_h \in (C^0(\overline{\Omega}_0))^3 : \mathbf{v}_h|_K \in \mathbb{P}_r \ \forall K \in \mathcal{T}_h\};$$

here X_h^r denotes the space of finite elements (FE) of degree $r \geq 1$, $\mathcal{T}_h$ a suitable triangulation of the domain $\overline{\Omega}_0$, h the mesh size and $N_h = dim(V_h)$ the dimension of the FOM space, respectively.

We then approximate the weak form of (2.1) by the finite element method yielding a nonlinear algebraic system. By applying the Newton method we obtain the following iterative procedure: given $\mathbf{u}_h^{(0)}(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$, for $k = 0, 1, \ldots$, find $\delta\mathbf{u}_h \in \mathbb{R}^{N_h}$ s.t.

$$\begin{cases} \mathbf{J}(\mathbf{u}_h^{(k-1)}(\boldsymbol{\mu}); \boldsymbol{\mu})\delta\mathbf{u}_h(\boldsymbol{\mu}) = -\mathbf{R}(\mathbf{u}_h^{(k-1)}(\boldsymbol{\mu}); \boldsymbol{\mu}), \\ \mathbf{u}_h^{(k)}(\boldsymbol{\mu}) = \mathbf{u}_h^{(k-1)}(\boldsymbol{\mu}) + \delta\mathbf{u}_h(\boldsymbol{\mu}) \end{cases} \tag{2.2}$$

and iterate until $||\mathbf{R}(\mathbf{u}_h^k(\boldsymbol{\mu}); \boldsymbol{\mu})||_{L^2(\Omega)} < \varepsilon$, being $\varepsilon > 0$ a small, fixed tolerance. Here

$$[\mathbf{J}(\mathbf{u}(\boldsymbol{\mu}); \boldsymbol{\mu})]_{ij} = \langle J(\mathbf{u}(\boldsymbol{\mu}); \boldsymbol{\mu})[\boldsymbol{\varphi}_j], \boldsymbol{\varphi}_i \rangle, \qquad [\mathbf{R}(\mathbf{u}(\boldsymbol{\mu}); \boldsymbol{\mu})]_i = R(\mathbf{u}(\boldsymbol{\mu}); \boldsymbol{\mu})(\boldsymbol{\varphi}_i), \qquad i, j = 1, \ldots, N_h$$

are the components of the Jacobian $\mathbf{J}(\mathbf{u}(\boldsymbol{\mu}); \boldsymbol{\mu}) \in \mathbb{R}^{N_h \times N_h}$ and the residual $\mathbf{R}(\mathbf{u}(\boldsymbol{\mu}); \boldsymbol{\mu}) \in \mathbb{R}^{N_h}$, respectively, where $\{\boldsymbol{\varphi}_j, j = 1, \ldots, N_h\}$ denote the basis functions of the FOM space V_h. Herein, we refer to (2.2) as to our FOM (or high-fidelity model).

Solving the high-dimensional problem $(2.2)_1$ can be rather expensive: the computational burden is represented by the assembling of the Jacobian matrix, which can demand almost the entire CPU time required by each Newton step. To reduce the computational complexity, we set up a projection-based ROM: for any $\boldsymbol{\mu} \in \mathcal{P}$ the high-fidelity solution is approximated as

$$\mathbf{u}_h(\boldsymbol{\mu}) \approx \mathbf{V}\mathbf{u}_N(\boldsymbol{\mu}), \qquad N \ll N_h, \tag{2.3}$$

being $\mathbf{u}_N(\boldsymbol{\mu}) \in \mathbb{R}^N$ and $\mathbf{V} \in \mathbb{R}^{N_h \times N}$ a matrix collecting a set of reduced basis (RB) functions, and, given $\mathbf{u}_N^{(0)}(\boldsymbol{\mu})$, find $\delta\mathbf{u}_N \in \mathbb{R}^N$ by solving for any $k \geq 1$

$$\begin{cases} \mathbf{W}^T\mathbf{J}(\mathbf{V}\mathbf{u}_N^{(k-1)}(\boldsymbol{\mu}); \boldsymbol{\mu})\mathbf{V}\delta\mathbf{u}_N(\boldsymbol{\mu}) = -\mathbf{W}^T\mathbf{R}(\mathbf{V}\mathbf{u}_N^{(k-1)}(\boldsymbol{\mu}); \boldsymbol{\mu}), \\ \mathbf{u}_N^{(k)}(\boldsymbol{\mu}) = \mathbf{u}_N^{(k-1)}(\boldsymbol{\mu}) + \delta\mathbf{u}_N(\boldsymbol{\mu}), \end{cases} \tag{2.4}$$

and iterate until $||\mathbf{W}^T\mathbf{R}(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}); \boldsymbol{\mu})||_2 < \varepsilon_{RB}$, being $\varepsilon_{RB} > 0$ a small, fixed tolerance. Problem $(2.4)_1$ is obtained by requiring that the (Petrov-Galerkin) projection over W_N of the FOM residual computed on the ansatz (2.3) vanishes, being the test space W_N an N-dimensional space generated by the columns of a matrix $\mathbf{W} \in \mathbb{R}^{N_h \times N}$; in the case where $\mathbf{W} = \mathbf{V}$, trial V_N and test W_N spaces coincide, and we rely instead on a Galerkin projection. See, e.g. [35] for further details; from now on we consider the Galerkin case, although the extension of the whole framework to the Petrov-Galerkin case is straightforward.

3 Solution-space reduction: POD technique

Our RB method relies on proper orthogonal decomposition (POD) for the construction of the reduced space (that is, the matrix $\mathbf{V}$), and on DEIM/MDEIM hyper-reduction techniques to efficiently assemble the Jacobian matrix and the residual vector appearing in $(2.4)_1$. In this section we recall the main properties of POD, which will be extensively employed also for the sake of basis construction when performing hyper-reduction.

Starting from a set of n_s solutions (also called snapshots) $\mathbf{u}_i = \mathbf{u}_h(\boldsymbol{\mu}_i)$, $i = 1, \ldots, n_s$ to the FOM problem (2.2), POD constructs a low-dimensional basis of V_h retaining as much as possible of the snapshots content of information. Here $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_{n_s}$ are randomly sampled points of $\mathcal{D}$; more ad-hoc strategies, such as *latin hypercube sampling*, could be exploited especially for high-dimensional parameter spaces. The POD basis is obtained performing the singular value decomposition (SVD) of the snapshot matrix

$$\mathbf{U} = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \ldots & \mathbf{u}_{n_s} \end{bmatrix} \in \mathbb{R}^{N_h \times n_s},$$

thus yielding the factorization

$$\mathbf{U} = \boldsymbol{\Phi}\boldsymbol{\Sigma}\mathbf{W}^T$$

where $\boldsymbol{\Sigma} = diag(\sigma_1, \sigma_2, \ldots)$, being $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_{n_s} > 0$ the n_s singular values of $\mathbf{U}$. The POD basis $\boldsymbol{\Phi}_N$ of dimension $N \leq n_s$ is obtained by collecting the first N left singular vectors of U,

corresponding to the first N (largest) singular values, that is, $\mathbf{\Phi}_N = [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_N] \in \mathbb{R}^{N \times N_h}$, with $N \ll n_s$; we can set the basis dimension N as the minimum integer such that

$$\frac{\sum_{i=1}^{N} \sigma_i^2}{\sum_{i=1}^{n_s} \sigma_i^2} \geq 1 - \varepsilon_{POD},$$

given a suitable, small tolerance ε_{POD}. The reduced basis provided by POD is optimal, meaning that for any set $\{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$ of N orthonormal vectors in $\mathbb{R}^{N_h}$, the POD basis $\mathbf{\Phi}_N$ solves the minimization problem

$$\min \left\{ \sum_{i=1}^{n_s} \|\mathbf{u}_i - \Pi_{Z_N} \mathbf{u}_i\|_2^2, \quad \mathbf{z}_i \in \mathbb{R}^{N_h}, \ \mathbf{z}_i^T \mathbf{z}_j = \boldsymbol{\delta}_{ij} \quad \forall i, j = 1, \ldots, N_h \right\}.$$

where $\Pi_{Z_N} \mathbf{u}$ denotes the orthogonal projection of $\mathbf{u} \in \mathbb{R}^{N_h}$ onto $Z_N = \mathrm{span}\{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$ with respect to the Euclidean norm $\|\cdot\|_2$. In other words, the POD basis minimizes the sum of the squared distances between each snapshot and the corresponding projection onto the subspace. We summarize the POD method in Algorithm 1; different norms can also be used instead of the euclidean one, see e.g. [35] for further details.

Algorithm 1 POD algorithm

INPUT: Snapshots matrix $\mathbf{U} \in \mathbb{R}^{N_h \times n_s}$, tolerance ε_{POD}
OUTPUT: $\mathbf{\Phi}_N$
1: Perform the SVD of $\mathbf{U}$: $\mathbf{U} = \mathbf{\Phi} \mathbf{\Sigma} \mathbf{W}^T$
2: Set the basis dimension N as the minimum integer such that

$$\frac{\sum_{i=1}^{N} \sigma_i^2}{\sum_{i=1}^{n_s} \sigma_i^2} \geq 1 - \varepsilon_{POD}$$

3: Construct $\mathbf{\Phi}_N = [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_N]$ selecting the first N columns of the matrix $\mathbf{\Phi}$.

---

4 System approximation: DEIM/MDEIM techniques

The reduced Newton problem (2.4) still depends on the high fidelity dimension N_h. Indeed, the Jacobian matrix and the residual vector depend on the solution computed at the previous step, so that at each Newton step we would need to reassemble them, then to project them onto the reduced space, in order to obtain the corresponding ROM arrays. Hyper-reduction is crucial in order to avoid this stage, and then recover the usual offline-online decomposition, essential for computational efficiency. For that, we rely on two DEIM algorithms: the former (see Sect. 4.1) is the classical one, and is employed on residual vectors, whereas the latter (MDEIM, see Sect. 4.2) is suitable to directly treat the Jacobian matrix.

4.1 DEIM for residual approximation

For the problem at hand, DEIM [11] allows to efficiently express the residual vector as a linear combination of (possibly few) $\boldsymbol{\mu}$-independent terms so that, at each Newton step, the $\boldsymbol{\mu}$-dependent weights of this combination can be efficiently computed by solving an interpolation problem. In this work, we approximate the reduced residual in (2.4) as

$$\mathbf{V}^T \mathbf{R}(\mathbf{V}\mathbf{u}_N(\boldsymbol{\mu}); \boldsymbol{\mu}) \approx \sum_{i=1}^{m_R} \theta_R^i(\boldsymbol{\mu}) \mathbf{V}^T \boldsymbol{\phi}_R^i, \tag{4.1}$$

where $\boldsymbol{\phi}_R^i \in \mathbb{R}^{N_h}$, $i = 1, \ldots, m_R$, is a set of basis functions that can be computed (and stored) offline once for all. We point out that, in order to obtain (4.1), we have to apply DEIM to a set of

high-fidelity residual vectors of the form $\mathbf{R}(\mathbf{V}\mathbf{u}_N(\boldsymbol{\mu}); \boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ and then to project the approximate residual onto the reduced space at a later time. In particular, we project the residual vector onto a low-dimensional subspace spanned by a basis $\boldsymbol{\Phi}_R \in \mathbb{R}^{N_h \times m_R}$ such that

$$\mathbf{R}(\mathbf{V}\mathbf{u}_N(\boldsymbol{\mu}); \boldsymbol{\mu}) \approx \mathbf{R}_m(\mathbf{V}\mathbf{u}_N(\boldsymbol{\mu}); \boldsymbol{\mu}) = \boldsymbol{\Phi}_R \boldsymbol{\theta}_R(\boldsymbol{\mu}),$$

where $\boldsymbol{\theta}_R(\boldsymbol{\mu}) \in \mathbb{R}^{m_R}$ is a coefficient vector to be determined.

The basis $\boldsymbol{\Phi}_R$ can be computed (once for all) by performing POD on a set of snapshots $\{\mathbf{R}(\mathbf{V}\mathbf{u}_N(\boldsymbol{\mu}_i); \boldsymbol{\mu}_i), \ i = 1, \dots, n_s\}$. Since we are dealing with a nonlinear problem, in order to obtain the residual snapshots $\mathbf{R}(\mathbf{V}\mathbf{u}_N(\boldsymbol{\mu}_i); \boldsymbol{\mu}_i)$, we need in principle to solve the reduced problem (2.4) for different values of $\boldsymbol{\mu}$ and, at each Newton iteration, to store the computed residual vectors. This would be expensive, since problem (2.4) requires to reassemble the residual vector and the Jacobian matrix at each Newton step and then to project them onto the reduced space in order to obtain the corresponding reduced order quantities. This issue will be addressed in section 4.4.

Regarding the coefficient vector $\boldsymbol{\theta}_R(\boldsymbol{\mu})$, it can be evaluated for each new $\boldsymbol{\mu}$ by imposing a set of m interpolation constraints on a subset $\wp = [\wp_1, \dots, \wp_m]$ of entries of $\mathbf{R}(\mathbf{V}\mathbf{u}_N(\boldsymbol{\mu}); \boldsymbol{\mu})$, selected by the DEIM algorithm (see Algorithm 2). In particular, $\boldsymbol{\theta}_R(\boldsymbol{\mu})$ is the solution of

$$\boldsymbol{\Phi}_R|_{\wp} \boldsymbol{\theta}_R(\boldsymbol{\mu}) = \mathbf{R}(\mathbf{V}\mathbf{u}_N(\boldsymbol{\mu}); \boldsymbol{\mu})|_{\wp}$$

where $\mathbf{R}(\mathbf{V}\mathbf{u}_N(\boldsymbol{\mu}); \boldsymbol{\mu})|_{\wp} \in \mathbb{R}^m$ and $\boldsymbol{\Phi}_R|_{\wp} \in \mathbb{R}^{n \times m}$ are the restrictions of $\mathbf{R}(\mathbf{V}\mathbf{u}_N(\boldsymbol{\mu}); \boldsymbol{\mu})$ and $\boldsymbol{\Phi}_R$ to the subset of indices $\wp$. We remark that, since at each Newton step the residual vector depends on the reduced solution computed at the previous iteration, the same holds for the coefficient vector $\boldsymbol{\theta}_R(\boldsymbol{\mu})$. Consequently, we need to compute a new $\boldsymbol{\theta}_R(\boldsymbol{\mu})$ at each Newton iteration.

Concerning the DEIM approximation, it can be shown that (see [11])

$$||\mathbf{R}(\boldsymbol{\mu}) - \mathbf{R}_m(\boldsymbol{\mu})||_2 \leq ||(\boldsymbol{\Phi}_R|_{\wp})^{-1}||_2 \mathcal{E}(\mathbf{R}(\boldsymbol{\mu})) \tag{4.2}$$

where $\mathcal{E}(\mathbf{R}(\boldsymbol{\mu})) = ||(I - \boldsymbol{\Phi}_R \boldsymbol{\Phi}_R^T)\mathbf{R}(\boldsymbol{\mu})||_2$; this latter quantity can then be approximated by the first discarded singular value in the POD computation of $\boldsymbol{\Phi}_R$, i.e.

$$\mathcal{E}(\mathbf{R}(\boldsymbol{\mu})) \approx \sigma_{m_R+1}.$$

We observe that this approximation holds for any $\boldsymbol{\mu}$ provided that a suitable sampling of the parameter domain has been carried out to construct the basis, see [11] for further details.

Remark 1. Constructing a basis able to retain as much as possible of the content of information related to the residuals vectors can be a challenging task. Indeed, residual vectors usually vary over a large range of values as they approach to the tolerance of the Newton method at the last Newton steps, while assuming large values at the first steps. Thus, a large number m_R of DEIM terms can be needed in order to accurately approximate the residuals in our reduced order model. Dealing with normalized or rescaled residuals does not seem to cure this difficulty. Another option is to express the residuals as the sum of different (non vanishing) components and then to perform DEIM separately on each component; nevertheless for the case at hand, this option does not represent a more efficient alternative regarding the number of terms m_R.

4.2 MDEIM for Jacobian approximation

In order to efficiently assemble the Jacobian matrix, we consider a matrix version of DEIM (MDEIM), recently proposed in [9], [30] to efficiently handle nonaffine parametrized problems. In this work we exploit this technique to perform hyper-reduction of those matrices arising in nonlinear parametrized problems. We approximate the reduced Jacobian matrix under the form

$$\mathbf{V}^T \mathbf{J}(\mathbf{V}\mathbf{u}_N(\boldsymbol{\mu}); \boldsymbol{\mu})\mathbf{V} \approx \sum_{i=1}^{m_J} \theta_J^i(\boldsymbol{\mu}) \mathbf{V}^T \mathbf{J}^i \mathbf{V}, \tag{4.3}$$

Algorithm 2 DEIM algorithm (as originally proposed in [11])

> INPUT: $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_m] \in \mathbb{R}^{n \times m}$ made by linearly independent columns
> OUTPUT: $\wp = [\wp_1, \ldots, \wp_m] \in \mathbb{R}^m$
> 1: $\wp_1 = maxpos\{\boldsymbol{\phi}_1\}$
> 2: $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1]$, $\mathbf{P} = [\mathbf{e}_{\wp_1}]$
> 3: **for** $k = 2, \ldots, m$ **do**
> 4:   Solve $(\mathbf{P}^T \boldsymbol{\Phi})\mathbf{c} = (\mathbf{P}^T \boldsymbol{\phi}_k)$
> 5: $\mathbf{r} = \boldsymbol{\phi}_k - \boldsymbol{\Phi}\mathbf{c}$
> 6: $\wp_k = maxpos\{\mathbf{r}\}$
> 7: $\boldsymbol{\Phi} \leftarrow [\boldsymbol{\Phi} \quad \boldsymbol{\phi}_k]$, $\mathbf{P} \leftarrow [\mathbf{P} \quad \mathbf{e}_{\wp_k}]$
> 8: **end for**

---

where $\mathbf{J}^i$ are $\boldsymbol{\mu}$-independent matrices that can be computed once for all and $\theta_J^i(\boldsymbol{\mu})$ is a coefficient vector. In general, MDEIM allows to approximate a parameter dependent matrix as a linear combination of $\boldsymbol{\mu}$-independent matrices, weighed by $\boldsymbol{\mu}$-dependent coefficients. We thus can express our Jacobian matrix $\mathbf{J}(\mathbf{Vu}_N(\boldsymbol{\mu}); \boldsymbol{\mu}) \in \mathbb{R}^{N_h \times N_h}$ as:

$$\mathbf{J}(\mathbf{Vu}_N(\boldsymbol{\mu}); \boldsymbol{\mu}) \approx \mathbf{J}_m(\mathbf{Vu}_N(\boldsymbol{\mu}); \boldsymbol{\mu}) = \sum_{i=1}^{m_J} \boldsymbol{\theta}_J^i(\boldsymbol{\mu}) \mathbf{J}^i, \tag{4.4}$$

being $\{\mathbf{J}^i \in \mathbb{R}^{N_h \times N_h}, \; i = 1, \ldots, m_J\}$ a set of $\boldsymbol{\mu}$-independent matrices. This approximation is obtained by defining

$$j(\mathbf{Vu}_N(\boldsymbol{\mu}); \boldsymbol{\mu}) = \text{vec}(\mathbf{J}(\mathbf{Vu}_N(\boldsymbol{\mu}); \boldsymbol{\mu}) \in \mathbb{R}^{n^2}$$

as the vector obtained by stacking all the columns of $\mathbf{J}(\mathbf{Vu}_N(\boldsymbol{\mu}); \boldsymbol{\mu})$ and approximating $j(\mathbf{Vu}_N(\boldsymbol{\mu}); \boldsymbol{\mu})$ by its DEIM counterpart

$$j(\mathbf{Vu}_N(\boldsymbol{\mu}); \boldsymbol{\mu}) \approx \boldsymbol{\Phi}_J \boldsymbol{\theta}_J(\boldsymbol{\mu}), \quad \boldsymbol{\Phi}_J = [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_n] \in \mathbb{R}^{n^2 \times m_J}.$$

Then, the matrices $\mathbf{J}^i$ can be computed transforming each column $\boldsymbol{\phi}_i \in \mathbb{R}^{n^2}$ of $\boldsymbol{\Phi}_J$ into a matrix $\mathbf{J}^i \in \mathbb{R}^{N_h \times N_h}$ by reverting the vec operation, as

$$\mathbf{J}^i = \text{vec}^{-1}(\boldsymbol{\phi}_i).$$

The computation of $\boldsymbol{\Phi}_J$ and $\boldsymbol{\theta}_J(\boldsymbol{\mu})$ is analogous to the one described for the residual vectors. Thus, we need to collect snapshots of $\mathbf{J}(\mathbf{Vu}_N(\boldsymbol{\mu}_i); \boldsymbol{\mu}_i)$, $i = 1, \ldots, n_s$, which are evaluated in the reduced solution. As already detected in the previous section, this is not a trivial issue, since it would require n_s solutions of (2.4), in addition to the n_s solutions of the high-fidelity problem, required to compute the basis $\mathbf{V}$. Different strategies to face this problem are discussed in section 4.4. The error bound (4.2) can be easily extended to the Matrix DEIM technique (see e.g. [30]), in order to estimate the error between $\mathbf{J}$ and $\mathbf{J}_m$. In particular, we have that

$$||\mathbf{J}(\boldsymbol{\mu}) - \mathbf{J}_m(\boldsymbol{\mu})||_2 \leq ||(\boldsymbol{\Phi}_J|_{\wp})^{-1}||_2 ||(I - \boldsymbol{\Phi}_J \boldsymbol{\Phi}_J^T)\mathbf{J}(\boldsymbol{\mu})||_2, \tag{4.5}$$

where the norm $||(I - \boldsymbol{\Phi}_J \boldsymbol{\Phi}_J^T)\mathbf{J}(\boldsymbol{\mu})||_2$ can be approximated by

$$||(I - \boldsymbol{\Phi}_J \boldsymbol{\Phi}_J^T)\mathbf{J}(\boldsymbol{\mu})||_2 \approx \sigma_{m_J+1},$$

provided that an accurate sampling of the parameter space has been performed.

Remark 2. Note that all matrices are stored in a sparse format. As a result, the actual dimension of the vectorized matrices is n_z rather than n^2, where n_z denotes the number of nonzero entries of the matrix. See e.g. [41] for further details.

Remark 3. For nonsingular matrices, the Bauer-Fike theorem guarantees that, as M increases, the singular values of the approximated matrix become closer and closer to the singular values of the original matrix; see e.g. [17]. This property ensures that the approximate matrix is nonsingular, and thus invertible at each Newton step, provided that a sufficiently large number of basis functions is chosen.

4.3 Efficient assembling on a reduced mesh

The DEIM algorithm allows to select a small subset of the grid nodes yielding a reduced mesh where vectors and matrices are then efficiently assembled. In the FE context the entries of the residual vector $\mathbf{R}$ are associated to the degrees of freedom of the problem so that $\mathbf{R}_\wp$ can be computed by simply integrating the residual only on the quadrature points belonging to the mesh elements which provide a non-zero contribution to the entries $\wp$; this set of elements is usually referred to as reduced mesh. Then, to compute the coefficient vector $\boldsymbol{\theta}_R(\boldsymbol{\mu})$ we need only to evaluate $\mathbf{R}$ on the reduced mesh and the same holds for the Jacobian matrix.

Hence, substituting (4.1) and (4.3) into (2.4), the hyper-reduced problem can be expressed as follows: given $\mathbf{u}_{N,m}^{(0)} \in \mathbb{R}^N$, at each Newton step $k \geq 1$ we seek $\delta\mathbf{u}_{N,m}^{(k)} \in \mathbb{R}^N$, satisfying

$$\begin{cases} \sum_{i=1}^{m_J} \theta_J^i(\boldsymbol{\mu}) \mathbf{V}^T \mathbf{J}^i \mathbf{V} \, \delta\mathbf{u}_{N,m}(\boldsymbol{\mu}) = -\sum_{i=1}^{m_r} \theta_R^i(\boldsymbol{\mu}) \mathbf{V}^T \mathbf{R}^i \\ \mathbf{u}_{N,m}^{(k)}(\boldsymbol{\mu}) = \mathbf{u}_{N,m}^{(k-1)}(\boldsymbol{\mu}) + \delta\mathbf{u}_{N,m}(\boldsymbol{\mu}). \end{cases} \tag{4.6}$$

and iterate until $\|\sum_{i=1}^{m_r} \theta_R^i(\boldsymbol{\mu}) \mathbf{V}^T \mathbf{R}^i\|_2 < \varepsilon_{RB}$. For each new value of $\boldsymbol{\mu}$, we have to compute only the coefficient vectors solving two interpolation problems. Consequently, during the online stage the reduced residual and the reduced Jacobian matrix can be assembled on the respective reduced meshes, exploiting previously computed arrays, thus yielding a significant reduction of computational time, above all for fully nonlinear problems.

Remark 4. We observe that, instead of approximating the Jacobian matrix with MDEIM, we could directly compute the derivative of the DEIM approximation of the residual by assembling the Jacobian matrix on the reduced mesh obtained for the residual vector (see e.g. [23]). However, since the approximation of the residual vectors often requires a considerable number of DEIM terms (see Remark 1), the associated reduced mesh often turns out be larger than the MDEIM reduced mesh. Therefore, using the MDEIM strategy to approximate the Jacobian matrix allows to obtain a greater reduction in terms of computational costs.

4.4 A combined POD/DEIM/MDEIM strategy

Interlacing the generation of the reduced basis functions and of the arrays required to assemble the hyper-reduced problem is not as straightforward as in the case of a linear elliptic problem, where the system approximation can take place before the solution-space reduction. In this section we discuss three possible strategies to combine the POD solution-space reduction and the proposed DEIM/MDEIM system approximation, showing how to rely on collected snapshots related to the solutions, the residual vectors and the Jacobian matrices. Indeed, as we will show, the choice of a suitable snapshots selection strategy has a dramatic impact both on the computational saving and on the accuracy of the reduced solution.

We recall that, during the offline stage, we need to solve the high-fidelity problem (2.2) n_s times, for properly selected parameter values. For each combination of the parameters and for each Newton iteration k, we store the displacement $\mathbf{u}_h^{(k)}(\boldsymbol{\mu}_i)$, $i = 1, \ldots, n_s$. These quantities form the snapshots matrix $\mathbf{U}$ required to compute the POD basis $\mathbf{V}$ for the solutions.

In addition, we need to construct the POD basis $\boldsymbol{\Phi}_R$ and $\boldsymbol{\Phi}_J$ for DEIM and Matrix DEIM approximations of the residual and the Jacobian matrix, respectively. We observe that the hyper-reduced problem (4.6) is the approximation of the reduced problem (2.4). Thus, the best way to construct the DEIM and MDEIM basis is to solve problem (2.4) n_s times to collect the snapshots of the residual vectors and the Jacobian matrices, required to assemble the snapshots matrices $\mathbf{U}_R$ and $\mathbf{U}_J$. We point out that, using this strategy, the POD solution-space reduction is first performed, while the bases associated to the the DEIM and MDEIM approximation of $\mathbf{R}$ and $\mathbf{J}$ are computed at a later time; for this reason we refer to this strategy, described in Algorithm 3, as *two-stages* HROM (hyper-reduced order model). Note that in this method, similarly to what is done in the so-called GNAT procedure [8], the snapshots from which we construct the basis functions for both $\mathbf{J}$ and $\mathbf{R}$ are computed by relying on the ROM, rather than on the FOM.

The major drawback of this strategy is that problem (2.4) is computationally demanding, since it requires to reassemble the residual vector and the Jacobian matrix at each Newton step and then to project them onto the reduced space in order to obtain the corresponding ROM arrays. Thus, we would like to avoid the computation of n_s solutions of this problem.

A possible way to overcome this issue may be a simultaneous solution-space approximation/system reduction strategy, which can be performed by storing the snapshots of the residual vectors and the Jacobian matrices when computing the solution of the high-fidelity problem. This *single-stage* strategy (Algorithm 4) has been widely used in the RB literature, see e.g. [11, 36]; however, in some cases, this method can provide a worst approximation of the residual vectors. In particular, a large number of DEIM terms is often required in order to correctly approximate the high-fidelity problem.

In this work, we propose a new enhanced strategy to collect the snapshots, referred to as *hybrid* HROM (Algorithm 5). This strategy is based on the introduction of a semi hyper-reduced problem involving a MDEIM approximation of the Jacobian matrix and a residual vector which is instead exactly computed: given $\mathbf{u}_{N,m}^{(0)} \in \mathbb{R}^N$, at each Newton step we search $\delta\mathbf{u}_{N,m}^{(k)} \in \mathbb{R}^N$, $k \geq 1$ satisfying

$$
\begin{cases}
\sum_{i=1}^{m_J} \theta_i^J(\boldsymbol{\mu}) \mathbf{V}^T \mathbf{J}^i \mathbf{V}\, \delta\mathbf{u}_{N,m}(\boldsymbol{\mu}) = -\mathbf{V}^T \mathbf{R}(\mathbf{V}\mathbf{u}_{N,m}^{(k-1)}(\boldsymbol{\mu})) \\
\mathbf{u}_{N,m}^{(k)}(\boldsymbol{\mu}) = \mathbf{u}_{N,m}^{(k-1)}(\boldsymbol{\mu}) + \delta\mathbf{u}_{N,m}(\boldsymbol{\mu})
\end{cases}
\tag{4.7}
$$

and iterate until $||\mathbf{V}^T \mathbf{R}(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}); \boldsymbol{\mu})|| < \varepsilon_{RB}$. Solving this problem turns out to be significantly faster than solving problem (2.4), since $\mathbf{J}$ is assembled only onto the reduced mesh. Not only, problem (4.7) requires almost the same effort of the full hyper-reduced problem (4.6), since the assembling of the Jacobian matrix takes almost the full time of a Newton iteration. Generating the residual snapshots according to this method allows to correctly approximate the residual vectors using a small number of bases, still retaining low offline computational costs.

Algorithm 3 Two-stages HROM

1: Select n_s combinations of parameters $\{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_{n_s}\}$
2: **for** $i = 1, \ldots, n_s$ **do**
3:     Solve problem (2.2) for $\boldsymbol{\mu}_i$
4:     At each Newton iteration k:

       • $\mathbf{U} \leftarrow [\mathbf{U} \quad \mathbf{u}_{h,i}^k(\boldsymbol{\mu}_i)]$

5: **end for**
6: $\mathbf{V} = \text{POD}(\mathbf{U}, \varepsilon)$
7: **for** $i = 1, \ldots, n_s$ **do**
8:     Solve problem (2.4) for $\boldsymbol{\mu}_i$
9:     At each Newton iteration k:

       • $\mathbf{U}_J \leftarrow [\mathbf{U}_J \quad vec(\mathbf{J}(\mathbf{V}\mathbf{u}_N^{k-1}(\boldsymbol{\mu}_i), \boldsymbol{\mu}_i))]$

       • $\mathbf{U}_R \leftarrow [\mathbf{U}_R \quad \mathbf{R}(\mathbf{V}\mathbf{u}_N^{k-1}(\boldsymbol{\mu}_i), \boldsymbol{\mu}_i)]$

10: **end for**
11: $\boldsymbol{\Phi}_R = \text{POD}(\mathbf{U}_R, \varepsilon_R)$, $\boldsymbol{\Phi}_J = \text{POD}(\mathbf{U}_J, \varepsilon_J)$.

4.5 Offline-Online decomposition

We remark that, during the offline stage, we first need to construct the basis $\mathbf{V}$, $\boldsymbol{\Phi}_R$ and $\boldsymbol{\Phi}_J$ using one of the methods proposed in the previous section. In a subsequent stage, we use the DEIM algorithm (2) to select the vectors of interpolation indices $\wp_R$ and $\wp_J$ in order to identify the nodes of the reduced meshes and, consequently, the reduced meshes elements, which are a priori different for the DEIM and the MDEIM cases. Finally, we compute and store all the structures required during the online phase, see Algorithm 6.

Algorithm 4 Single-stage HROM

1: Select n_s combinations of parameters $\{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_{n_s}\}$
2: **for** $i = 1, \ldots, n_s$ **do**
3:     Solve problem (2.2) for $\boldsymbol{\mu}_i$
4:     At each Newton iteration k:

      • $\mathbf{U} \leftarrow [\mathbf{U} \quad \mathbf{u}_{h,i}^k(\boldsymbol{\mu}_i)]$

      • $\mathbf{U}_J \leftarrow [\mathbf{U}_J \quad vec(\mathbf{J}(\mathbf{u}_{h,i}^{k-1}(\boldsymbol{\mu}_i), \boldsymbol{\mu}_i))]$

      • $\mathbf{U}_R \leftarrow [\mathbf{U}_R \quad \mathbf{R}(\mathbf{u}_{h,i}^{k-1}(\boldsymbol{\mu}_i), \boldsymbol{\mu}_i)]$

5: **end for**
6: $\mathbf{V} = \text{POD}(\mathbf{U}, \varepsilon)$, $\boldsymbol{\Phi}_R = \text{POD}(\mathbf{U}_R, \varepsilon_R)$, $\boldsymbol{\Phi}_J = \text{POD}(\mathbf{U}_J, \varepsilon_J)$.

Algorithm 5 Hybrid HROM

1: Select n_s combinations of parameters $\{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_{n_s}\}$
2: **for** $i = 1, \ldots, n_s$ **do**
3:     Solve problem (2.2) for $\boldsymbol{\mu}_i$
4:     At each Newton iteration k:

      • $\mathbf{U} \leftarrow [\mathbf{U} \quad \mathbf{u}_{h,i}^k(\boldsymbol{\mu}_i)]$

      • $\mathbf{U}_J \leftarrow [\mathbf{U}_J \quad vec(\mathbf{J}(\mathbf{u}_{h,i}^{k-1}(\boldsymbol{\mu}_i), \boldsymbol{\mu}_i))]$

5: **end for**
6: $\mathbf{V} = \text{POD}(\mathbf{U}, \varepsilon)$, $\boldsymbol{\Phi}_J = \text{POD}(\mathbf{U}_J, \varepsilon_J)$
7: **for** $i = 1, \ldots, n_s$ **do**
8:     Solve problem (4.7) for $\boldsymbol{\mu}_i$
9:     At each Newton iteration k:

      • $\mathbf{U}_R \leftarrow [\mathbf{U}_R \quad \mathbf{R}(\mathbf{V}\mathbf{u}_N^{k-1}(\boldsymbol{\mu}_i), \boldsymbol{\mu}_i)]$

10: **end for**
11: $\boldsymbol{\Phi}_R = \text{POD}(\mathbf{U}_R, \varepsilon_R)$.

The online stage can then be performed in a very inexpensive way since, at each Newton step, it only requires the assembling of the $\wp_R$ and $\wp_J$ on a small subset of the grid elements and the solution of low dimensional linear systems. In particular, for each online Newton iteration, we need to solve two interpolation problems and a $N \times N$ linear system to compute the solution, see Algorithm 7. The cost of the online phase strictly depends on the number of DEIM/MDEIM basis functions considered, especially for the Jacobian matrix, whose assembling is by far the most expensive operation.

We point out that, in order to correctly integrate the Jacobian matrix and the residual on the reduced mesh, we need at each Newton step the solution entries corresponding to the subset $\tilde{\wp}$ which contains all the nodes belonging to the elements of the reduced mesh. We report in figure 1 the sketch of a reduced mesh in a simple two-dimensional case; the set $\wp$ of the DEIM nodes is highlighted in black while the set $\tilde{\wp}$ contains both the black and the blue nodes.

Specifically, we deal with two different subsets $\tilde{\wp}_R$ and $\tilde{\wp}_J$, having two different reduced meshes for the approximation of the residual vector and the Jacobian matrix. We introduce $\mathbf{u}|_{\tilde{\wp}_R}$ and $\mathbf{u}|_{\tilde{\wp}_J}$ as the restriction of the solution to the subsets $\tilde{\wp}_R \in \mathbb{R}^{\tilde{m}_R}$ and $\tilde{\wp}_J \in \mathbb{R}^{\tilde{m}_J}$, respectively, and the restriction matrices

$$\tilde{\mathbf{P}}_R = [\mathbf{e}|_{\tilde{\wp}_{R1}}, \mathbf{e}|_{\tilde{\wp}_{R2}}, \ldots], \quad \tilde{\mathbf{P}}_J = [\mathbf{e}|_{\tilde{\wp}_{J1}}, \mathbf{e}|_{\tilde{\wp}_{J2}}, \ldots].$$

In order to efficiently compute $\mathbf{u}|_{\tilde{\wp}_R}$ and $\mathbf{u}|_{\tilde{\wp}_J}$ at each online Newton iteration, we pre-assemble the matrices

$$\mathbf{S}_R = \tilde{\mathbf{P}}_R \mathbf{V} \in \mathbb{R}^{\tilde{m}_R \times N}, \quad \mathbf{S}_J = \tilde{\mathbf{P}}_J \mathbf{V} \in \mathbb{R}^{\tilde{m}_J \times N}$$

which allow to transform $\mathbf{u}_{N,m}$ in $\mathbf{u}|_{\tilde{\wp}_R} = \mathbf{S}_R \mathbf{u}_{N,m}$ and $\mathbf{u}|_{\tilde{\wp}_J} = \mathbf{S}_J \mathbf{u}_{N,m}$. The entire computational
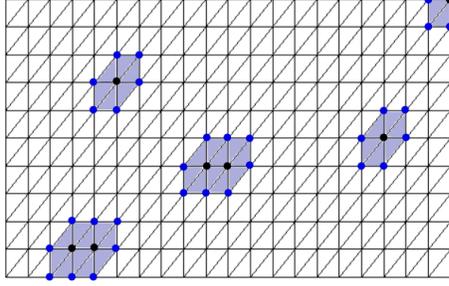
Figure 1: Sketch of a reduced mesh in a simple two-dimensional case.

procedure is summarized in Algorithms 6 and 7.

Algorithm 6 Offline stage

1: Compute basis $\mathbf{V}$, $\boldsymbol{\Phi}_R$, $\boldsymbol{\Phi}_J$ using Algorithm 4 or 5
2: Assemble offline structures:

  - $\mathbf{J}_N^i = \mathbf{V}^T \mathbf{J}^i \mathbf{V}, \quad i = 1, \ldots m_j, \quad \mathbf{J}^i = vec^{-1}(\boldsymbol{\Phi}_J^i)$

  - $\mathbf{R}_N^i = \mathbf{V}^T \boldsymbol{\Phi}_R^i, \quad i = 1, \ldots, m_r$

3: $\wp_R$=DEIM($\boldsymbol{\Phi}_R$), $\wp_J$=DEIM($\boldsymbol{\Phi}_J$)
4: Construct the reduced meshes and compute $\mathbf{S}_R = \tilde{\mathbf{P}}_R \mathbf{V} \in \mathbb{R}^{\tilde{m}_R \times N}$ and $\quad \mathbf{S}_J = \tilde{\mathbf{P}}_J \mathbf{V} \in \mathbb{R}^{\tilde{m}_J \times N}$

Algorithm 7 Online stage

  INPUT: $\boldsymbol{\mu}$
  OUTPUT: $\mathbf{u}_{N,m}(\boldsymbol{\mu})$
1: **while** $||\mathbf{R}_N^{(k)}|| > \varepsilon$ **do**
2: Assemble $\mathbf{R}^{(k)}(\boldsymbol{\mu})|_\wp$, $\mathbf{J}^{(k)}(\boldsymbol{\mu})|_\wp$ on the reduced meshes
3: Compute the coefficient vectors solving the linear systems

$$\boldsymbol{\Phi}_R|_\wp \boldsymbol{\theta}_R(\boldsymbol{\mu}) = \mathbf{R}^{(k)}(\boldsymbol{\mu})|_\wp, \qquad \boldsymbol{\Phi}_J|_\wp \boldsymbol{\theta}_J(\boldsymbol{\mu}) = \mathbf{J}^{(k)}(\boldsymbol{\mu})|_\wp$$

4: Compute $\mathbf{R}_N^{(k)}(\boldsymbol{\mu}) = \sum_{i=1}^{M_R} \theta_R^i(\boldsymbol{\mu})\mathbf{R}_N^i, \quad \mathbf{J}_N^{(k)}(\boldsymbol{\mu}) = \sum_{i=1}^{M_J} \theta_J^i(\boldsymbol{\mu})\mathbf{J}_N^i$
5: Solve $\mathbf{J}_N^{(k)}(\boldsymbol{\mu})\delta\mathbf{u}_{N,m}^{(k)}(\boldsymbol{\mu}) = -\mathbf{R}_N^{(k)}(\boldsymbol{\mu})$
6: $\mathbf{u}_{N,m}^{(k+1)}(\boldsymbol{\mu}) = \mathbf{u}_{N,m}^{(k)}(\boldsymbol{\mu}) + \delta\mathbf{u}_{N,m}^{(k)}(\boldsymbol{\mu})$
7: $\mathbf{u}|_{\tilde{\wp}_R}(\boldsymbol{\mu}) = \mathbf{S}_R\mathbf{u}_{N,m}^{(k+1)}(\boldsymbol{\mu}), \quad \mathbf{u}|_{\tilde{\wp}_J}(\boldsymbol{\mu}) = \mathbf{S}_J\mathbf{u}_{N,m}^{(k+1)}(\boldsymbol{\mu})$
8: **end while**

---

5 A posteriori error estimation

The aim of this section is to recover an a posteriori error estimator for the norm of the error $e_{N,m}(\boldsymbol{\mu}) = \mathbf{u}_h(\boldsymbol{\mu}) - \mathbf{u}_{N,m}(\boldsymbol{\mu})$ between the high-fidelity solution and the reduced solution. The proposed error estimate accounts for both the error related to the solution-space reduction and the errors associated with the DEIM and MDEIM approximations of $\mathbf{R}$ and $\mathbf{J}$.

Let us denote by $\mathbf{X}_h$ the matrix associated to a discrete norm in V_h. Moreover, we define the stability factor

$$\beta_m^N(\boldsymbol{\mu}) = ||\mathbf{X}_h^{1/2}\mathbf{J}_m(\mathbf{V}\mathbf{u}_{N,m}(\boldsymbol{\mu}); \boldsymbol{\mu})^{-1}\mathbf{X}_h^{1/2}||_2^{-1} = \sigma_{min}(\mathbf{X}_h^{-1/2}\mathbf{J}_m(\mathbf{V}\mathbf{u}_{N,m}(\boldsymbol{\mu}); \boldsymbol{\mu})\mathbf{X}_h^{-1/2})$$

and the quantity

$$r(\boldsymbol{\mu}) = \frac{2}{\beta_m^N(\boldsymbol{\mu})}(||\mathbf{R}_h(\mathbf{Vu}_{N,m}(\boldsymbol{\mu});\boldsymbol{\mu})||_{\mathbf{X}_h^{-1}} + ||\mathbf{J}_h(\mathbf{Vu}_{N,m}(\boldsymbol{\mu});\boldsymbol{\mu}) - \mathbf{J}_m(\mathbf{Vu}_{N,m}(\boldsymbol{\mu});\boldsymbol{\mu})||_{\mathbf{X}_h,\mathbf{X}_h^{-1}}),$$

where we denote the $(\mathbf{X}_h, \mathbf{X}_h^{-1})$ norm of a generic matrix $\mathbf{A} \in \mathbb{R}^{N_h \times N_h}$ by

$$||\mathbf{A}||_{\mathbf{X}_h,\mathbf{X}_h^{-1}} = \sup_{\mathbf{v} \in \mathbb{R}^{N_h}} \frac{||\mathbf{Av}||_{\mathbf{X}_h^{-1}}}{||\mathbf{v}||_{\mathbf{X}_h}} = \sup_{\mathbf{v} \in \mathbb{R}^{N_h}} \frac{||\mathbf{X}_h^{-1/2}\mathbf{A}\mathbf{X}_h^{-1/2}\mathbf{v}||_2}{||\mathbf{v}||_2} \qquad \forall \mathbf{A} \in \mathbb{R}^{N_h \times N_h}.$$

Finally, we denote by $\overline{B}_r(\mathbf{v})$ the closed ball centered in $\mathbf{v}$ with radius r.

Theorem 5.1. *Assume that $\mathbf{u}_{N,m}(\boldsymbol{\mu})$ is a solution of problem (4.6) and that $\mathbf{J}_h(\mathbf{Vu}_{N,m}(\boldsymbol{\mu});\boldsymbol{\mu})$ is locally Lipschitz continuous at $\mathbf{Vu}_{N,m}(\boldsymbol{\mu})$, i.e., there exists $K_h^N(\boldsymbol{\mu}) > 0$ such that for all $\mathbf{v} \in \overline{B}_{r(\boldsymbol{\mu})}(\mathbf{Vu}_{N,m}(\boldsymbol{\mu}))$*

$$||\mathbf{J}_h(\mathbf{Vu}_{N,m}(\boldsymbol{\mu});\boldsymbol{\mu}) - \mathbf{J}_h(\mathbf{v};\boldsymbol{\mu})||_{\mathbf{X}_h,\mathbf{X}_h^{-1}} \leq K_h^N(\boldsymbol{\mu})||\mathbf{Vu}_{N,m}(\boldsymbol{\mu}) - \mathbf{v}||_{\mathbf{X}_h}. \tag{5.1}$$

Moreover, we introduce

$$\tau_{N,1}(\boldsymbol{\mu}) = \frac{6K_h^N(\boldsymbol{\mu})||\mathbf{R}_h(\mathbf{Vu}_{N,m}(\boldsymbol{\mu});\boldsymbol{\mu})||_{\mathbf{X}_h^{-1}}}{(\beta_m^N(\boldsymbol{\mu}))^2}$$

and

$$\tau_{N,2}(\boldsymbol{\mu}) = \frac{12K_h^N(\boldsymbol{\mu})||\mathbf{J}_h(\mathbf{Vu}_{N,m}(\boldsymbol{\mu});\boldsymbol{\mu}) - \mathbf{J}_m(\mathbf{Vu}_{N,m}(\boldsymbol{\mu});\boldsymbol{\mu})||_{\mathbf{X}_h,\mathbf{X}_h^{-1}}^2}{(\beta_m^N(\boldsymbol{\mu}))^2}.$$

If $\tau_{N,1}(\boldsymbol{\mu}) \leq 1$ and $\tau_{N,2}(\boldsymbol{\mu}) \leq 1$, there exists a unique solution $\mathbf{u}_h(\boldsymbol{\mu}) \in \overline{B}_{r(\boldsymbol{\mu})}(\mathbf{Vu}_{N,m}(\boldsymbol{\mu}))$. Furthermore, the following a posteriori error estimate holds:

$$\begin{aligned}
||\mathbf{u}_h(\boldsymbol{\mu}) - \mathbf{Vu}_N(\boldsymbol{\mu})||_{\mathbf{X}_h} \leq \frac{2}{\beta_m(\boldsymbol{\mu})}\Big(&||\mathbf{R}_m(\mathbf{Vu}_N(\boldsymbol{\mu});\boldsymbol{\mu})||_{\mathbf{X}_h^{-1}} \\
&+ ||\mathbf{R}_h(\mathbf{Vu}_N(\boldsymbol{\mu});\boldsymbol{\mu}) - \mathbf{R}_m(\mathbf{Vu}_N(\boldsymbol{\mu});\boldsymbol{\mu})||_{\mathbf{X}_h^{-1}} \\
&+ ||\mathbf{J}_h(\mathbf{Vu}_N(\boldsymbol{\mu});\boldsymbol{\mu}) - \mathbf{J}_m(\mathbf{Vu}_N(\boldsymbol{\mu});\boldsymbol{\mu})||_{\mathbf{X}_h,\mathbf{X}_h^{-1}}^2\Big).
\end{aligned} \tag{5.2}$$

Proof. For the sake of notation, hereon we omit the $\boldsymbol{\mu}$-dependence. In order to prove the existence of a unique solution $\mathbf{u}_h$ in the closed ball $\overline{B}_r(\mathbf{Vu}_{N,m})$, let us define the map $\mathbf{H}: \mathbb{R}^{N_h} \to \mathbb{R}^{N_h}$

$$\mathbf{H}(\mathbf{v}) = \mathbf{v} - \mathbf{J}_m(\mathbf{Vu}_{N,m})^{-1}\mathbf{R}_h(\mathbf{v}),$$

and show that $\mathbf{H}$ is a strict contraction in $\overline{B}_r(\mathbf{Vu}_{N,m})$. First, we prove that $\mathbf{H}$ maps $\overline{B}_r(\mathbf{Vu}_{N,m})$ into itself; indeed, for any $\mathbf{v} \in \overline{B}_r(\mathbf{Vu}_{N,m})$ we can write

$$\begin{aligned}
\mathbf{H}(\mathbf{v}) - \mathbf{Vu}_{N,m} &= \mathbf{v} - \mathbf{J}_m(\mathbf{Vu}_{N,m})^{-1}\mathbf{R}_h(\mathbf{v}) - \mathbf{Vu}_{N,m} \\
&= \mathbf{J}_m(\mathbf{Vu}_{N,m})^{-1}[\mathbf{J}_m(\mathbf{Vu}_{N,m})(\mathbf{v} - \mathbf{Vu}_{N,m}) - \mathbf{R}_h(\mathbf{v}) + \mathbf{R}_h(\mathbf{Vu}_{N,m}) - \mathbf{R}_h(\mathbf{Vu}_{N,m})].
\end{aligned}$$

Using the mean value theorem, we obtain

$$\mathbf{R}_h(\mathbf{v}) - \mathbf{R}_h(\mathbf{Vu}_{N,m}) = \int_0^1 \mathbf{J}_h(\mathbf{Vu}_{N,m} + s(\mathbf{v} - \mathbf{Vu}_{N,m}))(\mathbf{v} - \mathbf{Vu}_{N,m})ds,$$

so that

$$\begin{aligned}
\mathbf{H}(\mathbf{v}) - \mathbf{Vu}_{N,m} = \mathbf{J}_m(\mathbf{Vu}_{N,m})^{-1}\Big[&(\mathbf{J}_m(\mathbf{Vu}_{N,m}) - \mathbf{J}_h(\mathbf{Vu}_{N,m}))(\mathbf{v} - \mathbf{Vu}_{N,m}) \\
&+ \int_0^1 (\mathbf{J}_h(\mathbf{Vu}_{N,m}) - \mathbf{J}_h(\mathbf{Vu}_{N,m} + s(\mathbf{v} - \mathbf{Vu}_{N,m}))(\mathbf{v} - \mathbf{Vu}_{N,m})ds - \mathbf{R}_h(\mathbf{Vu}_{N,m})\Big].
\end{aligned}$$

Then,

$$||\mathbf{H}(\mathbf{v}) - \mathbf{V}\mathbf{u}_{N,m}||_{\mathbf{X}_h} \leq \frac{1}{\beta_m^N}\Big(||\mathbf{v} - \mathbf{V}\mathbf{u}_{N,m}||_{\mathbf{X}_h}||\mathbf{J}_h(\mathbf{V}\mathbf{u}_{N,m}) - \mathbf{J}_m(\mathbf{V}\mathbf{u}_{N,m})||_{\mathbf{X}_h,\mathbf{X}_h^{-1}}$$

$$+ ||\mathbf{v} - \mathbf{V}\mathbf{u}_{N,m}||_{\mathbf{X}_h}\int_0^1 ||\mathbf{J}_h(\mathbf{V}\mathbf{u}_{N,m}) - \mathbf{J}_h(\mathbf{V}\mathbf{u}_{N,m} + s(\mathbf{v} - \mathbf{V}\mathbf{u}_{N,m}))||_{\mathbf{X}_h,\mathbf{X}_h^{-1}}ds$$

$$+ ||\mathbf{R}_h(\mathbf{V}\mathbf{u}_{N,m})||_{\mathbf{X}_h^{-1}}\Big).$$

Using the Young inequality with $\varepsilon = 1/K_h^N$ and the Lipschitz property (5.1), we obtain

$$||\mathbf{H}(\mathbf{v}) - \mathbf{V}\mathbf{u}_{N,m}||_{\mathbf{X}_h} \leq \frac{1}{\beta_m^N}\Big(\frac{1}{2K_h^N}||\mathbf{J}_h(\mathbf{V}\mathbf{u}_{N,m}) - \mathbf{J}_m(\mathbf{V}\mathbf{u}_{N,m})||^2_{\mathbf{X}_h,\mathbf{X}_h^{-1}}$$

$$+ \frac{3K_h^N}{2}||\mathbf{v} - \mathbf{V}\mathbf{u}_{N,m}||^2_{\mathbf{X}_h} + ||\mathbf{R}_h(\mathbf{V}\mathbf{u}_{N,m})||_{\mathbf{X}_h^{-1}}\Big).$$

Recalling that $\mathbf{v} \in \overline{B}_r(\mathbf{V}\mathbf{u}_{N,m})$ and requiring that $\tau_{N,1} \leq 1$ and $\tau_{N,2} \leq 1$, we can get the following bound

$$||\mathbf{H}(\mathbf{v}) - \mathbf{V}\mathbf{u}_{N,m}||_{\mathbf{X}_h} \leq \frac{1}{\beta_m^N}\Big[\frac{1}{2K_h^N}||\mathbf{J}_h(\mathbf{V}\mathbf{u}_{N,m}) - \mathbf{J}_m(\mathbf{V}\mathbf{u}_{N,m})||^2_{\mathbf{X}_h,\mathbf{X}_h^{-1}}$$

$$+ \frac{3K_h^N}{2}\Big(\frac{4}{(\beta_m^N)^2}||\mathbf{J}_h(\mathbf{V}\mathbf{u}_{N,m}) - \mathbf{J}_m(\mathbf{V}\mathbf{u}_{N,m})||^4_{\mathbf{X}_h,\mathbf{X}_h^{-1}}$$

$$+ \frac{4}{(\beta_m^N)^2}||\mathbf{R}_h(\mathbf{V}\mathbf{u}_{N,m})||_{\mathbf{X}_h^{-1}}||\mathbf{J}_h(\mathbf{V}\mathbf{u}_{N,m}) - \mathbf{J}_m(\mathbf{V}\mathbf{u}_{N,m})||^2_{\mathbf{X}_h,\mathbf{X}_h^{-1}}$$

$$+ \frac{4}{(\beta_m^N)^2}||\mathbf{R}_h(\mathbf{V}\mathbf{u}_{N,m})||^2_{\mathbf{X}_h^{-1}}\Big) + ||\mathbf{R}_h(\mathbf{V}\mathbf{u}_{N,m})||_{\mathbf{X}_h^{-1}}\Big],$$

so that

$$||\mathbf{H}(\mathbf{v}) - \mathbf{V}\mathbf{u}_{N,m}||_{\mathbf{X}_h} \leq \frac{1}{\beta_m^N}\Big(2||\mathbf{J}_h(\mathbf{V}\mathbf{u}_{N,m}) - \mathbf{J}_m(\mathbf{V}\mathbf{u}_{N,m})||^2_{\mathbf{X}_h,\mathbf{X}_h^{-1}} + 2||\mathbf{R}_h(\mathbf{V}\mathbf{u}_{N,m})||_{\mathbf{X}_h^{-1}}\Big) = r.$$

(5.3)

Therefore, $\mathbf{H}(\mathbf{v}) \in \overline{B}_r(\mathbf{V}\mathbf{u}_{N,m})$. We now show that $\mathbf{H}$ is a strict contraction. Let us consider $\mathbf{v}_1, \mathbf{v}_2 \in \overline{B}_r(\mathbf{V}\mathbf{u}_{N,m})$, then

$$\mathbf{H}(\mathbf{v}_1) - \mathbf{H}(\mathbf{v}_2) = \mathbf{v}_1 - \mathbf{J}_m(\mathbf{V}\mathbf{u}_{N,m})^{-1}\mathbf{R}_h(\mathbf{v}_1) - \mathbf{v}_2 + \mathbf{J}_m(\mathbf{V}\mathbf{u}_{N,m})^{-1}\mathbf{R}_h(\mathbf{v}_2)$$

$$= \mathbf{J}_m(\mathbf{V}\mathbf{u}_{N,m})^{-1}\Big[(\mathbf{v}_1 - \mathbf{v}_2)(\mathbf{J}_m(\mathbf{V}\mathbf{u}_{N,m}) - \mathbf{J}_h(\mathbf{V}\mathbf{u}_{N,m}))$$

$$+ \int_0^1 \mathbf{J}_h(\mathbf{V}\mathbf{u}_{N,m}) - \mathbf{J}_h(\mathbf{v}_2 + s(\mathbf{v}_1 - \mathbf{v}_2))(\mathbf{v}_1 - \mathbf{v}_2)ds\Big].$$

Thus, we obtain

$$||\mathbf{H}(\mathbf{v}_1) - \mathbf{H}(\mathbf{v}_2)||_{\mathbf{X}_h} \leq \frac{1}{\beta_m^N}\Big(||\mathbf{v}_1 - \mathbf{v}_2||_{\mathbf{X}_h}||\mathbf{J}_h(\mathbf{V}\mathbf{u}_{N,m}) - \mathbf{J}_m(\mathbf{V}\mathbf{u}_{N,m})||_{\mathbf{X}_h,\mathbf{X}_h^{-1}} + K_h^N||\mathbf{v}_1 - \mathbf{v}_2||^2_{\mathbf{X}_h}\Big)$$

$$\leq \frac{1}{\beta_m^N}\Big(||\mathbf{J}_h(\mathbf{V}\mathbf{u}_{N,m}) - \mathbf{J}_m(\mathbf{V}\mathbf{u}_{N,m})||_{\mathbf{X}_h,\mathbf{X}_h^{-1}} + K_h^N R\Big)||\mathbf{v}_1 - \mathbf{v}_2||_{\mathbf{X}_h}$$

$$\leq \Big(\frac{1}{\beta_m^N}||\mathbf{J}_h(\mathbf{V}\mathbf{u}_{N,m}) - \mathbf{J}_m(\mathbf{V}\mathbf{u}_{N,m})||_{\mathbf{X}_h,\mathbf{X}_h^{-1}}$$

$$+ \frac{2K_h^N}{(\beta_m^N)^2}||\mathbf{J}_h(\mathbf{V}\mathbf{u}_{N,m}) - \mathbf{J}_m(\mathbf{V}\mathbf{u}_{N,m})||^2_{\mathbf{X}_h,\mathbf{X}_h^{-1}} + \frac{2K_h^N}{(\beta_m^N)^2}||\mathbf{R}_h(\mathbf{V}\mathbf{u}_{N,m})||_{\mathbf{X}_h^{-1}}\Big)||\mathbf{v}_1 - \mathbf{v}_2||_{\mathbf{X}_h}.$$

Recalling that $\tau_{N,1} \leq 1$ and $\tau_{N,2} \leq 1$, and thus also $\sqrt{\tau_{N,2}} \leq 1$, we can obtain the following bound

$$||\mathbf{H}(\mathbf{v}_1) - \mathbf{H}(\mathbf{v}_2)||_{\mathbf{X}_h} \leq \Big(\frac{1}{\sqrt{12K_h^N}} + \frac{1}{6} + \frac{1}{3}\Big)||\mathbf{v}_1 - \mathbf{v}_2||_{\mathbf{X}_h}$$

$$\leq \Big(\frac{1}{\sqrt{12}} + \frac{1}{2}\Big)||\mathbf{v}_1 - \mathbf{v}_2||_{\mathbf{X}_h} < ||\mathbf{v}_1 - \mathbf{v}_2||_{\mathbf{X}_h}.$$

Thanks to the Banach fixed-point theorem (see e.g. [12]), there exists a unique fixed point $\mathbf{u}_h \in \overline{B}_r(\mathbf{Vu}_{N,m})$ of $\mathbf{H}$, i.e. $\mathbf{H}(\mathbf{u}_h) = \mathbf{u}_h$, hence $\mathbf{R}_h(\mathbf{u}_h) = 0$. In conclusion, using (5.3) we have

$$
\begin{aligned}
||\mathbf{u}_h - \mathbf{Vu}_N||_{\mathbf{X}_h} \;&\leq\; \frac{2}{\beta_m(\boldsymbol{\mu})}\Big(||\mathbf{R}_h(\mathbf{Vu}_N)||_{\mathbf{X}_h^{-1}} + ||\mathbf{J}_h(\mathbf{Vu}_N) - \mathbf{J}_m(\mathbf{Vu}_N)||^2_{\mathbf{X}_h,\mathbf{X}_h^{-1}} \Big) \\
&\leq\; \frac{2}{\beta_m(\boldsymbol{\mu})}\Big(||\mathbf{R}_m(\mathbf{Vu}_N)||_{\mathbf{X}_h^{-1}} + ||\mathbf{R}_h(\mathbf{Vu}_N) - \mathbf{R}_m(\mathbf{Vu}_N)||_{\mathbf{X}_h^{-1}} \\
&\qquad\qquad + ||\mathbf{J}_h(\mathbf{Vu}_N) - \mathbf{J}_m(\mathbf{Vu}_N)||^2_{\mathbf{X}_h,\mathbf{X}_h^{-1}} \Big).
\end{aligned}
$$

$\square$

We observe that the provided error bound appearing on the left hand side of (5.2) still depends on the high-fidelity residual and Jacobian matrix; the computation of these quantities may entail high computational costs, especially if the error bound has to be evaluated for many values of the parameters. A possible way to overcome this drawback is to exploit the DEIM error bounds (4.2) and (4.5) in order to estimate the hyper-reduction error components. Recalling that

$$
||\mathbf{v}||_{\mathbf{X}_h^{-1}} \leq \frac{||\mathbf{v}||_2}{(\lambda_{min}(\mathbf{X}_h))^{1/2}}, \qquad ||\mathbf{J}||_{\mathbf{X}_h,\mathbf{X}_h^{-1}} \leq \frac{||\mathbf{J}||_2}{(\lambda_{min}(\mathbf{X}_h))^{1/2}},
$$

for any $\mathbf{v} \in \mathbb{R}^{N_h}$, $\mathbf{J} \in \mathbb{R}^{N_h \times N_h}$, we obtain the following "*approximate*" error estimate:

$$
\begin{aligned}
||\mathbf{u}_h(\boldsymbol{\mu}) - \mathbf{Vu}_N(\boldsymbol{\mu})||_{\mathbf{X}_h} \;&\leq\; \frac{2}{\beta_m(\boldsymbol{\mu})}\Big(||\mathbf{R}_m(\mathbf{Vu}_N(\boldsymbol{\mu});\boldsymbol{\mu})||_{\mathbf{X}_h^{-1}} \\
&\qquad + ||\mathbf{R}_h(\mathbf{Vu}_N(\boldsymbol{\mu});\boldsymbol{\mu}) - \mathbf{R}_m(\mathbf{Vu}_N(\boldsymbol{\mu});\boldsymbol{\mu})||_{\mathbf{X}_h^{-1}} \\
&\qquad + ||\mathbf{J}_h(\mathbf{Vu}_N(\boldsymbol{\mu});\boldsymbol{\mu}) - \mathbf{J}_m(\mathbf{Vu}_N(\boldsymbol{\mu});\boldsymbol{\mu})||^2_{\mathbf{X}_h,\mathbf{X}_h^{-1}} \Big) \\
&\approx\; \frac{2}{\beta_m(\boldsymbol{\mu})}\Big(||\mathbf{R}_m(\mathbf{Vu}_N(\boldsymbol{\mu});\boldsymbol{\mu})||_{\mathbf{X}_h^{-1}} + \frac{\sigma_{M_R+1}}{(\lambda_{min}(\mathbf{X}_h))^{1/2}}||(\boldsymbol{\Phi}^R|_\wp)^{-1}||_2 \\
&\qquad + \frac{\sigma^2_{M_J+1}}{\lambda_{min}(\mathbf{X}_h)}||(\boldsymbol{\Phi}^J|_\wp)^{-1}||^2_2 \Big).
\end{aligned}
\tag{5.4}
$$

We remark that all the quantities appearing in (5.4) are easily computable; in particular, for each new $\boldsymbol{\mu} \in \mathcal{D}$, we need to evaluate only $\beta_m(\boldsymbol{\mu})$ and $\mathbf{R}_m(\mathbf{Vu}_N(\boldsymbol{\mu}))$, which require exclusively the approximations of the residual vector and of the Jacobian matrix based on DEIM and MDEIM respectively. Nevertheless, we remark that the "approximated" error estimator in (5.4) may yield a coarse overestimate of the real error thus leading to a less effective error bound, while the estimator provided by (5.2) turns out to be sharper.

6 Numerical results

In order to show the effectiveness of the proposed framework, we present two different applications related to nonlinear mechanical problems: a structural test on a simple geometry and a more complex application to cardiac electromechanics. Numerical simulations have been entirely performed within the parallel finite element library LifeV (see www.lifev.org).

Before addressing these two examples, let us briefly recall a general framework to formulate our mechanical problems. We consider a reference configuration Ω_0 and an actual configuration Ω. A deformation is a map $\boldsymbol{\varphi} : \Omega_0 \to \Omega$ from the reference to the actual configuration, such that $\mathbf{x} = \boldsymbol{\varphi}(\mathbf{X})$ for any $\mathbf{X} \in \Omega_0$, $\mathbf{x} \in \Omega$. The deformation gradient tensor $\mathbf{F} : \Omega_0 \to \Omega$ is defined as

$$
\mathbf{F} = \frac{\partial \boldsymbol{\varphi}}{\partial \mathbf{X}}.
$$

Then, let us introduce the displacement vector field defined by the map

$$
\mathbf{u} : \Omega_0 \to \Omega, \qquad \mathbf{u}(\mathbf{X}) = \varphi(\mathbf{X}) - \mathbf{X};
$$

the deformation gradient tensor can be written in terms of the displacement as

$$\mathbf{F} = \mathbf{I} + \nabla \mathbf{u}.$$

We also denote by $J = det(\mathbf{F})$ the determinant of $\mathbf{F}$ and by $\mathbf{C} : \Omega_0 \to \Omega$, $\mathbf{C} = \mathbf{F}^T \mathbf{F}$ the left Cauchy-Green strain tensor. Denoting with $\mathcal{W}$ the strain energy function, we introduce the Piola tensor $\mathbf{P}$, related to $\mathcal{W}$ through the relation

$$\mathbf{P} = \frac{\partial \mathcal{W}}{\partial \mathbf{F}}. \tag{6.1}$$

In order to compute the deformation $\mathbf{u}$ of a body occupying the original deformation Ω_0, the problem we have to solve is given by the balance of the linear momentum (in material coordinates), which reads as follows:

$$div(\mathbf{P}(\mathbf{u}(\boldsymbol{\mu}); \boldsymbol{\mu})) = \mathbf{f}(\boldsymbol{\mu}) \qquad \text{in } \Omega_0, \tag{6.2}$$

with suitable boundary conditions, which will be detailed for each problem in the following sections. This equation can be written under the form (2.1), where $R(\mathbf{u}(\boldsymbol{\mu}); \boldsymbol{\mu}) = div(\mathbf{P}(\mathbf{u}(\boldsymbol{\mu}); \boldsymbol{\mu})) - \mathbf{f}(\boldsymbol{\mu})$; the high-fidelity solution of this equation can thus be computed relying on the Newton method (2.2).

6.1 Shear test for a cubic domain

In this first test case we assess the performance of the proposed method on a typical structural mechanics problem; in particular, we perform a shear test on a cubic domain $\Omega_0 = [0, 1]^3$. The high fidelity approximation is built on a conforming mesh with 3072 elements and 729 vertices, resulting in an high-fidelity space V_h of dimension $N_h = 2187$.
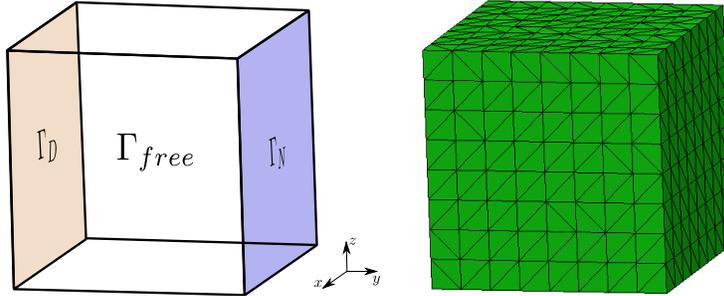
Figure 2: Test case 1: domain and computational grid.

We consider a Saint Venant-Kirchhoff material, which is characterized by the following strain energy function

$$\mathcal{W}(\mathbf{E}) = \frac{\lambda}{2}[tr(\mathbf{E})]^2 + \mu tr(\mathbf{E}^2),$$

where λ and μ are the Lamé constants and $\mathbf{E} = \frac{1}{2}(\mathbf{C} - \mathbf{I})$ is the Lagrangian Green strain tensor. We recall that the Lamé constants depend on the Young modulus E and the Poisson coefficient ν through the following relation

$$\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)} \qquad \mu = \frac{E}{2(1+\nu)}.$$

We point out that the choice of the Saint Venant-Kirchhoff material leads to the following Piola tensor, which is characterized by a polynomial nonlinearity:

$$\mathbf{P} = \lambda tr(\mathbf{E})\mathbf{F} + 2\mu \mathbf{F}\mathbf{E},$$

15

and the nonlinear mechanical problem we are going to solve reads as follows:

$$\begin{cases} div(\mathbf{P}(\mathbf{u}(\boldsymbol{\mu}); \boldsymbol{\mu})) = \mathbf{0} & \text{in } \Omega_0 \\ \mathbf{P}(\mathbf{u}(\boldsymbol{\mu}); \boldsymbol{\mu})\mathbf{n} = g\hat{\mathbf{z}} & \text{on } \Gamma_N \\ \mathbf{P}(\mathbf{u}(\boldsymbol{\mu}); \boldsymbol{\mu})\mathbf{n} = 0 & \text{on } \Gamma_{free} \\ \mathbf{u}(\boldsymbol{\mu}) = \mathbf{0} & \text{on } \Gamma_D. \end{cases}$$

The Dirichlet and the Neumann boundaries are reported in Figure 2 together with the computational mesh. For the case at hand, we consider the following parameters:

- Young modulus $E \in [6 \cdot 10^4, 7 \cdot 10^4]$,
- Poisson coefficient $\nu \in [0.3, 0.4]$,
- External load $g \in [1000, 2000]$.

In order to show the approximation properties of the proposed framework and to analyze its performances we aim to compare the three options to obtain an HROM proposed in section 4.4. In Figure 3 and 4 we report the normalized eigenvalues of the correlation matrix $\mathbf{U}^T\mathbf{U}$, related to the displacements, the Jacobian matrices and the residual vectors for the three options proposed. We observe that in all cases there is a rapid decay of the eigenvalues magnitude, which guarantees a satisfactory approximation of the displacements, the Jacobian matrices and the residual vectors with a small number of basis functions.
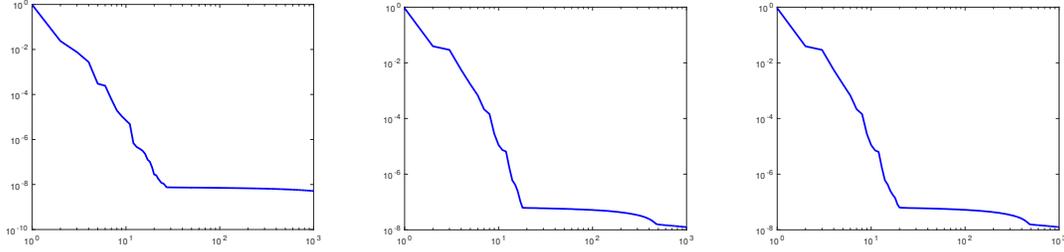
Figure 3: Normalized eigenvalues of the correlation matrix related to the displacement (left) and the Jacobian matrix using the two-stages HROM (center) and the single-stage and hybrid HROM (right).
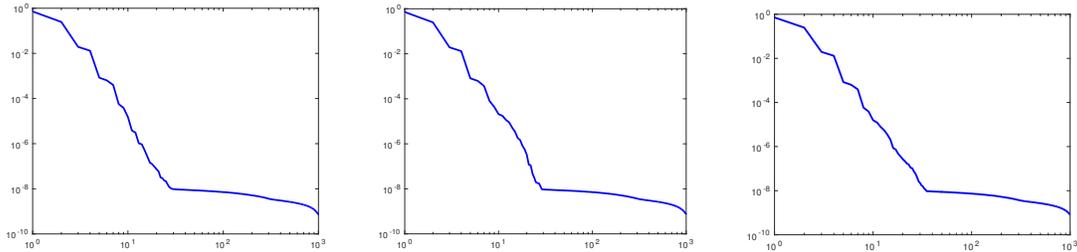
Figure 4: Normalized eigenvalues (blue) of the correlation matrix related to the residual vectors using the two-stages HROM (left), the hybrid HROM (center) and the single-stage HROM (right).

We show in Figures 5 and 6 the average $||\cdot||_2$ errors of the DEIM approximation for the Jacobian matrix and the residual vectors using the three hyper-reduction options proposed. The error has been computed over a test sample of 50 parameters, chosen during the online stage, different from the ones used to compute the snapshots. These figures highlight that the two-stages and the hybrid methods are able to capture the residual vectors and the Jacobian matrices relying on a significantly smaller number of basis functions than the number of bases required by the single-stage HROM.

16
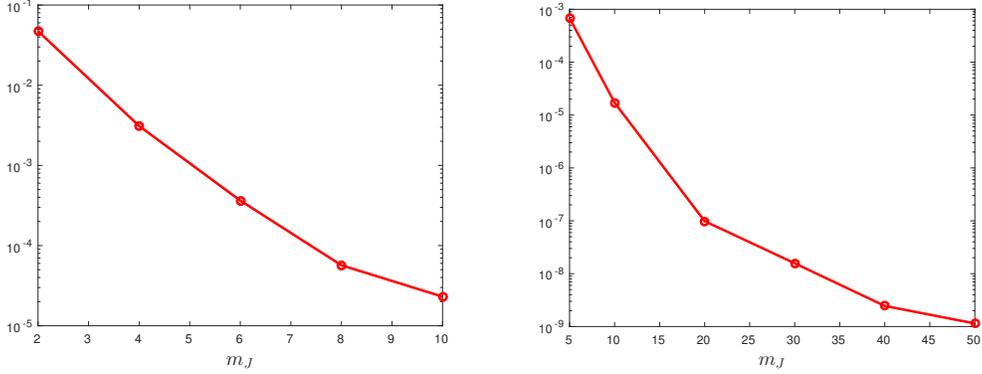
Figure 5: Average $|| \cdot ||_2$ error related to the Matrix DEIM approximation of the Jacobian matrix using the two-stages HROM (left) and the single-stage and hybrid HROM (right).
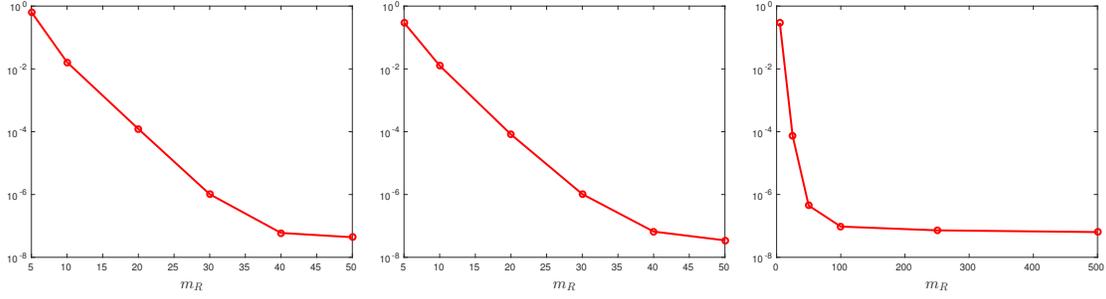
Figure 6: Average $|| \cdot ||_2$ error related to the DEIM approximation of the residual vectors using the two-stages HROM (left), the hybrid HROM (center) and the single-stage HROM (right).

This difference can be explained recalling that the single-stage method tries to approximate residual vectors arising from the ROM with a basis obtained from snapshots related to the FOM.

In Figure 7 we compare the error $||\mathbf{u}_h - \mathbf{V}\mathbf{u}_{N,m}||_{\mathbf{X}_h}$ as a function of the number of selected basis functions and the related a posteriori error bound for the three hyper-reduction options. The matrix $\mathbf{X}_h$ is associated with the norm induced in the discrete space by the continous H^1-norm. The error has been computed over our testing set of 50 randomly chosen parameters. The Figure shows that we need less than 20 basis functions in order to correctly reproduce the high fidelity solution.

The two-stages and the hybrid methods lead to similar results, requiring about 50 DEIM terms to approximate the residuals and 10 or 20 basis, respectively, to correctly approximate the Jacobian matrix. The hybrid method requires instead a huge number of basis functions (about 500) to reconstruct the residual vectors. The number of DEIM and MDEIM terms to be used in the online phase is chosen in order to guarantee the convergence of the reduced Newton problem for all the parameter combinations considered.

Details related to the computational performances of the three methods are shown in Table 1. We point out that the online CPU time required by the hybrid method is more than two times bigger than the times required by the other two strategies, whereas the offline CPU times are comparable in these two cases. Hence, we can conclude that the snapshots selection described in the single-stage HROM is not enough accurate in order to construct an efficient ROM. On the contrary, both the two-stages and the hybrid method allow to obtain a RB solution which correctly approximate the high-fidelity solution, significantly reducing the computational time. Regarding the online stage, these two strategies lead to comparable results, in terms of N and CPU times. However, the hybrid

17
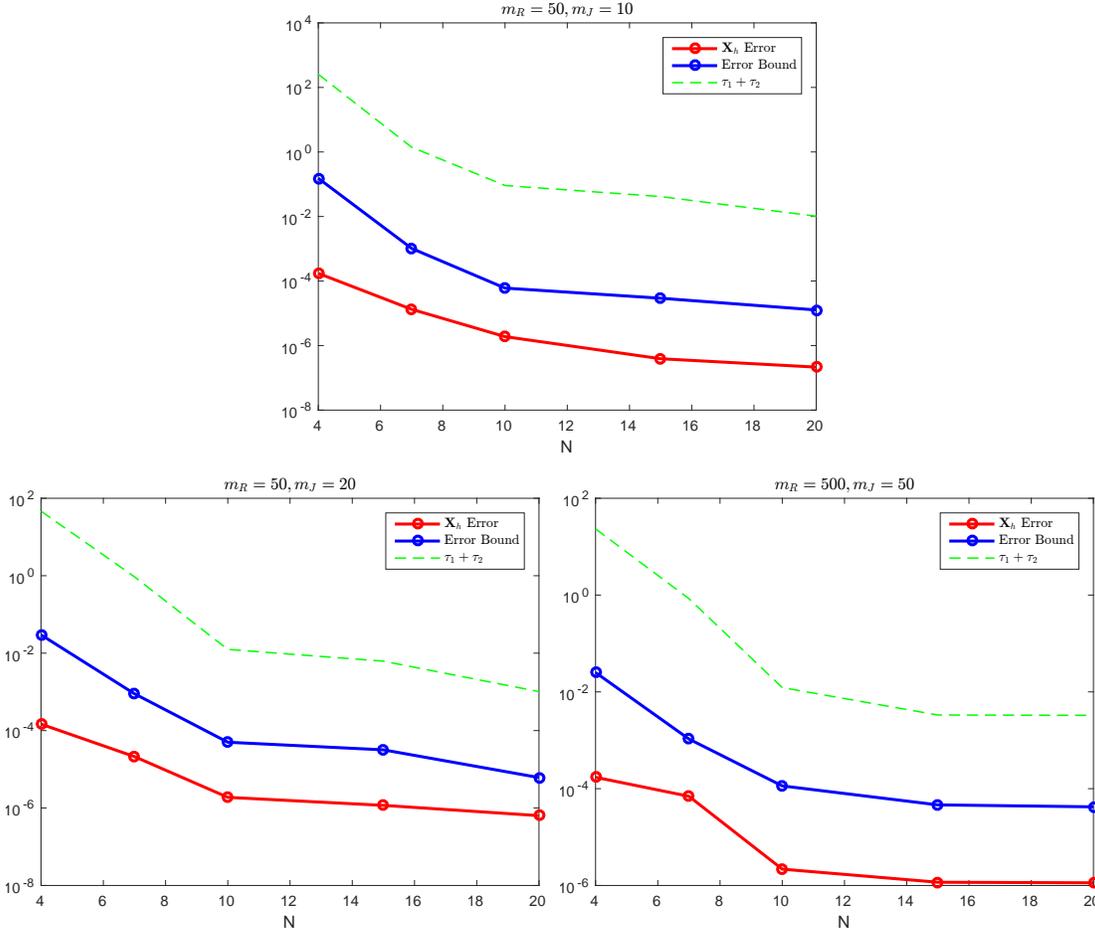
Figure 7: Average $\mathbf{X}_h$ error (red), a posteriori error bound (blue) and $\tau_1 + \tau_2$ (green) computed over a testing set of 50 parameters using the two-stages HROM (up), the hybrid HROM (left) and the single-stage HROM (right).

	Two-stages HROM	Hybrid HROM	Single-stage HROM
Residual DEIM terms	50	50	500
Jacobian DEIM terms	10	20	50
RB time Jacobian assembling	0.07s	0.1s	0.22s
Offline time	3h30	2h15	2h
Online RB time	0.83s	0.9s	2s

Table 1: Comparison of the three different hyper-reduction options proposed.

method is the only one able to provide a reduction strategy characterized by accurate and fast online stage and efficient offline phase. In fact, the offline time required by the hybrid method is about 60% of the offline time associated to the two-stages method. This analysis underlines the crucial role of the snapshots selection method in the construction of an efficient reduced framework. The results obtained in this test case suggest that the hybrid method has to be preferred, since it guarantees good approximation properties during the online stage, while retaining reasonable offline computational costs.

In Figure 8 we report the high-fidelity solutions, the reduced solutions and their differences,

computed for three different values of the parameters; we can observe that the reduced model is able to preserve the accuracy of the high-fidelity model, showing a maximum difference between the two solutions of about 10^{-6}. Here, the reduced solutions have been computed using the hybrid method, with $N = 10$, $m_R = 50$ and $m_J = 20$.
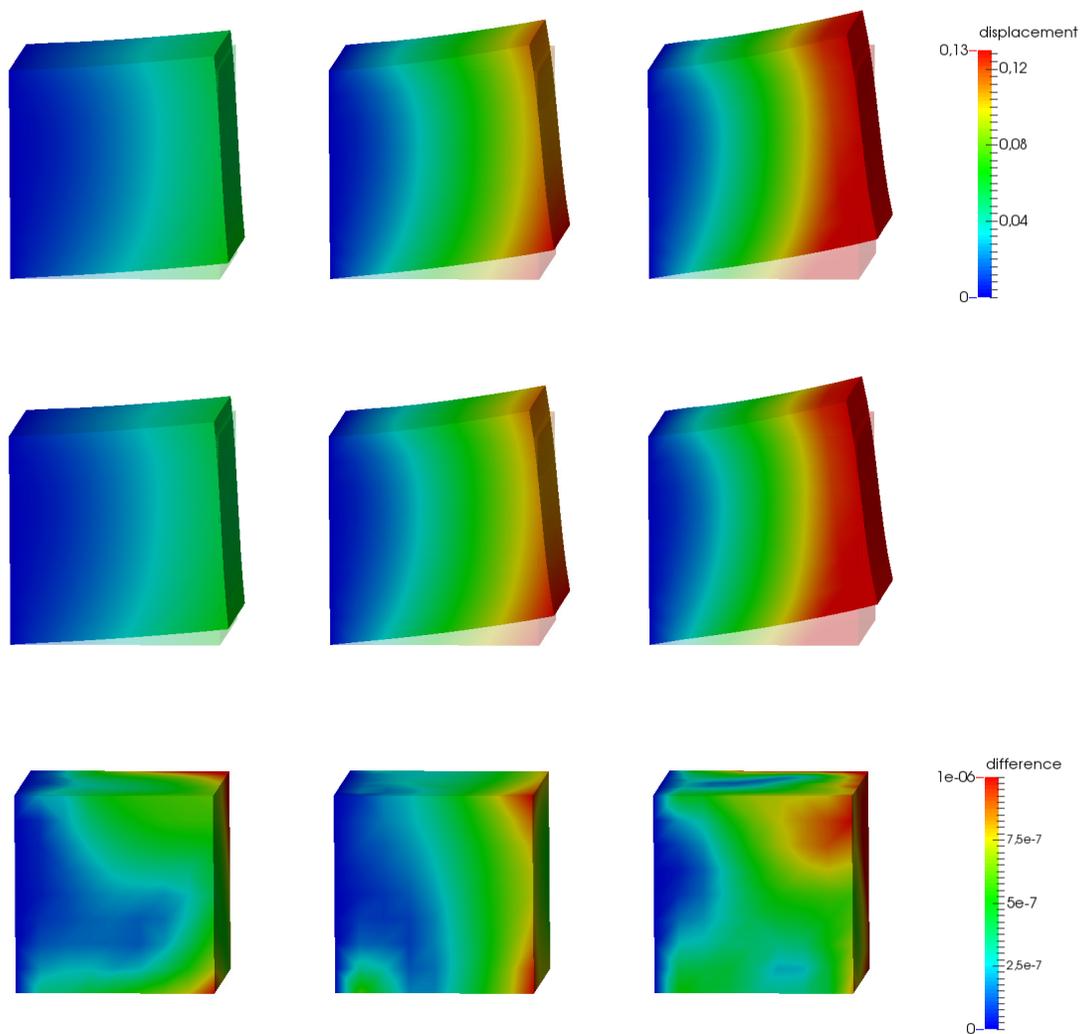
Figure 8: High-fidelity solutions (top), reduced order solutions (middle) and their differences (bottom) for three different parameter values ($\boldsymbol{\mu}_1 = [6.2 \cdot 10^4, 0.32, 1009.4]$, $\boldsymbol{\mu}_2 = [6.6 \cdot 10^4, 0.37, 1685.43]$, $\boldsymbol{\mu}_3 = [6.9 \cdot 10^4, 0.39, 1920.39]$).

Concerning the a posteriori error estimation, the proposed error bound has effectivity of about $O(10^2)$ and it can be considered a reliable estimation, provided that we are treating a fully nonlinear problem. For the case at hand, it is possible to evaluate the Lipschitz constant $K_h^N(\boldsymbol{\mu})$ explicitly, obtaining

$$K_h^N(\boldsymbol{\mu}) = (2\lambda + 3\mu)(2 + 2C + r(\boldsymbol{\mu})),$$

where $C = \sup_{\boldsymbol{\mu} \in \mathcal{D}} ||\mathbf{V}\mathbf{u}_{N,m}(\boldsymbol{\mu})||_{\mathbf{X}_h}$; see the appendix for a detailed derivation. As λ, μ and r are

parameters dependent, we denote by K_h^N an upper bound of the Lipschitz constant:

$$K_h^N(\boldsymbol{\mu}) \leq K_h^N = (2\lambda_{max} + 3\mu_{max})(2 + 2C + r_{max}) \qquad \forall \boldsymbol{\mu} \in \mathcal{D}.$$

For the case we are considering, we obtain $K_h^N \leq 10^6$, where C has been numerically estimated as the average value of $||\mathbf{Vu}_{N,m}(\boldsymbol{\mu})||_{\mathbf{X}_h}$, computed over a test sample of 100 parameters. In Tables 2, 3 and 4 we report further details and the values of $\tau_{N,1}$ and $\tau_{N,2}$ (computed using the three different hyper-reduction approaches) varying the number of reduced basis functions, in order to show that they are actually smaller than 1.

We observe that the hypoteses of Theorem 4.1 hold when $N \geq 10$ for the two-stages method and when $N \geq 7$ for the hybrid and the single-stage method. This is due to the fact that we can obtain inaccurate approximations of the high fidelity solution when considering too few RB functions. In conclusion we achieve a computational speed-up of about one order of magnitude in terms of CPU time, as we can see in Table 5, where we report some numerical details and data related to the computational performance of our technique for the case at hand.

	$||\mathbf{R}_h(\mathbf{Vu}_{N,m})||_{\mathbf{X}_h^{-1}}$	$||\mathbf{J}_h(\mathbf{Vu}_{N,m}) - \mathbf{J}_m(\mathbf{Vu}_{N,m})||_{\mathbf{X}_h,\mathbf{X}_h^{-1}}$	$\tau_{N,1}$	$\tau_{N,2}$
N=4	14.624	32.017	1.515	253.5
N=7	0.791	2.304	0.088	1.313
N=10	0.098	0.565	0.012	0.079
N=15	0.006	0.405	$7 \cdot 10^{-4}$	0.040
N=20	0.004	0.191	$1 \cdot 10^{-4}$	0.009

Table 2: Values of $\tau_{N,1}$ and $\tau_{N,2}$ varying the number of reduced basis using the two-stages method.

	$||\mathbf{R}_h(\mathbf{Vu}_{N,m})||_{\mathbf{X}_h^{-1}}$	$||\mathbf{J}_h(\mathbf{Vu}_{N,m}) - \mathbf{J}_m(\mathbf{Vu}_{N,m})||_{\mathbf{X}_h,\mathbf{X}_h^{-1}}$	$\tau_{N,1}$	$\tau_{N,2}$
N=4	12.606	13.457	1.602	44.78
N=7	0.811	1.841	0.100	0.838
N=10	0.098	0.037	0.012	$3 \cdot 10^{-4}$
N=15	0.050	0.005	0.006	$5 \cdot 10^{-6}$
N=20	0.008	0.005	0.001	$5 \cdot 10^{-6}$

Table 3: Values of $\tau_{N,1}$ and $\tau_{N,2}$ varying the number of reduced basis using the hybrid method.

	$||\mathbf{R}_h(\mathbf{Vu}_{N,m})||_{\mathbf{X}_h^{-1}}$	$||\mathbf{J}_h(\mathbf{Vu}_{N,m}) - \mathbf{J}_m(\mathbf{Vu}_{N,m})||_{\mathbf{X}_h,\mathbf{X}_h^{-1}}$	$\tau_{N,1}$	$\tau_{N,2}$
N=4	19.316	9.345	1.715	21.599
N=7	0.819	1.713	0.118	0.726
N=10	0.087	0.076	0.011	0.0014
N=15	0.022	0.046	0.003	$5 \cdot 10^{-4}$
N=20	0.022	0.046	0.003	$5 \cdot 10^{-4}$

Table 4: Values of $\tau_{N,1}$ and $\tau_{N,2}$ varying the number of reduced basis using the single-stage method.

Number of FE dofs	2187	Number of RB dofs	10
POD tolerance	10^{-8}	DEIM tolerance	10^{-12}
Newton tolerance	10^{-10}	MDEIM tolerance	10^{-10}
FE time	$\sim 10s$	FE time Jacobian assembling	$\sim 1s$
RB time (hybrid HROM)	$\sim 0.9s$	RB time Jacobian assembling (hybrid HROM)	$\sim 0.1s$

Table 5: Numerical details (test case 1)

6.2 Cardiac electromechanics in an idealized left ventricle

In this section we apply the proposed framework to the solution of a cardiac electromechanical problem, whose equations are characterized by the presence of highly nonlinear terms, and we present a numerical test on an idealized left ventricle geometry. Cardiac electromechanics is a coupled problem consisting of an electrical subproblem, which describes the propagation of the electrical signal triggering the heart contraction, and a mechanical subproblem, that describes the deformations of the cardiac muscle. In this work, we consider the full electromechanical model and we apply our reduction technique to the mechanical subproblem. This choice is motivated by the fact that the electrical model is considerably fast compared with the mechanical one. Our ultimate goal is to apply the proposed technique to the efficient characterization of the electromechanical behavior of the heart in a range of scenarios described in terms of relevant physiological parameters. Here, we perform a preliminary analysis relying on a static mechanical model, in order to assess the performances of the proposed method when considering a coupled problem, characterized by a complex constitutive law. The main limitation of this analysis is that a static mechanical model only allows to investigate the first portion of the heart beat, characterized by small deformations of the myocardium. The extension of the proposed framework to the solution of a time dependent problem covering a complete heart beat is the object of a forthcoming work. We point out that our method allows to consider parameters which are related with both the mechanical subproblem (e.g. physical properties of the myocardium as the Young modulus) and the electrophysiology (e.g. conductivity velocities), which thus affect the mechanics through the solution of the electrical problem.

6.2.1 Problem setting

In this section we briefly describe the cardiac electromechanical model we adopt, focusing on the mechanical subproblem, see e.g. [34] for a general introduction to the subject. The common approach to describe cardiac electrophysiology is to couple a ionic model, which describes the evolution of ion concentrations and ionic currents in the cell, together with a tissue model describing the spreading of the signal in the heart tissue. In this work we adopt the minimal model introduced by Bueno and Orovio in [6], coupled with the monodomain model (a complete derivation can be found in e.g. [13]), which are able to capture the relevant phenomena required to describe the normal electromechanical coupling.

To describe the cardiac muscle displacements, we assume an orthotropic mechanical constitutive law, that accounts for two preferred directions: muscular fibers and sheets directions, which are essential to determine the ability of the ventricle to twist and swell correctly during the filling phase. For the case at hand, we consider the hyperelastic model proposed in [20], characterized by an invariant-based formulation. This model relies on the following strain energy function

$$\mathcal{W} = \frac{a}{2b}\left[e^{b(\mathcal{I}_1-3)} - 1\right] + \frac{a_f}{2b_f}\left[e^{b_f(\mathcal{I}_{4,\mathbf{f}_0}-1)^2} - 1\right] + \frac{a_s}{2b_s}\left[e^{b_s(\mathcal{I}_{4,\mathbf{s}_0}-1)^2} - 1\right] + \frac{a_{fs}}{2b_{fs}}\left[e^{b_{fs}\mathcal{I}_{8,\mathbf{f}_0\mathbf{s}_0}^2} - 1\right],$$

where $\mathbf{f}_0$, $\mathbf{s}_0$ are the two (fibers and sheets respectively) preferred directions and $\mathcal{I}_1$, $\mathcal{I}_{4,\mathbf{f}_0}$, $\mathcal{I}_{4,\mathbf{s}_0}$, $\mathcal{I}_{8,\mathbf{f}_0\mathbf{s}_0}$ are the invariants of the right Cauchy-Green strain tensor, defined as

$$\mathcal{I}_1 = tr(\mathbf{C}), \quad \mathcal{I}_{4,\mathbf{f}_0} = \mathbf{f}_0 \cdot \mathbf{C}\mathbf{f}_0, \quad \mathcal{I}_{4,\mathbf{s}_0} = \mathbf{s}_0 \cdot \mathbf{C}\mathbf{s}_0, \quad \mathcal{I}_{8,\mathbf{f}_0\mathbf{s}_0} = \mathbf{f}_0 \cdot \mathbf{C}\mathbf{s}_0,$$

respectively. Coefficients of the Holzapfel-Ogden constitutive law are taken from the literature, see e.g. [14]. Fibers and sheets vectors are computed using the algorithm proposed in [37]: the idea of this procedure is based on the assumption that sheets are lying along the radial direction $\mathbf{s}_0$. Fibers are then obtained constructing a rotation matrix which describes the rotation of the fiber field around the $\mathbf{s}_0$ axis. Their orientation varies from an angle $-\theta_{max}$ on the epicardium to an angle $+\theta_{max}$ on the endocardium. For an idealized human ventricle, we obtain the fibers distribution shown in Figure 9.

In order to properly describe the myocardium deformations, we include a suitable incompressibility constraint. In this work we consider a quasi-incompressible formulation such as the one
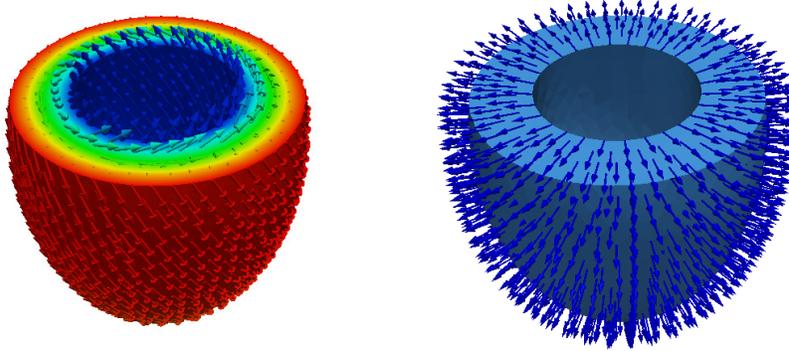
Figure 9: Muscular fibers with $\theta_{max} = 60°$ (left) and sheets (right).

described in [38, 19]. This formulation has different advantages with respect to a full incompressible constraint, from both a modeling and a numerical viewpoint; see e.g. [33] for further details. Moreover, in order to model the systolic part of the cardiac cycle, the active contraction of the muscular fibers has to be included in the force balance. In order to couple electrophysiology and mechanics, we need to define a modified Piola tensor $\mathbf{P}$ which depends on the solution of the electrical problem. In this work we rely on the active stress approach (see e.g. [28, 16, 32]), where an active contribution $\mathbf{P}_a$ is added to the Piola tensor; this latter thus becomes

$$\mathbf{P}_{tot} = \mathbf{P} + \mathbf{P}_a,$$

where $\mathbf{P}$ is the classical Piola tensor defined in (6.1). Several expressions for P_a can be found in literature, see e.g. [39, 5]. In this work we consider the following relation

$$\mathbf{P}_a = \mathbf{P}_a(\mathbf{u}, H(v_t)) = T_a H^2(v_t)\mathbf{F}\mathbf{f}_0 \otimes \mathbf{f}_0,$$

where the activation H is a spatial field obtained by the solution of the following ordinary differential equation for each $\mathbf{x} \in \Omega$ (further details and parameters tuning can be found in [37])

$$\begin{cases} \beta\dot{H}(v_t) = \alpha v_t - 2T_a H(v_t) & t \in (0, T), \\ H(0) = H_0. \end{cases} \tag{6.3}$$

Here $\dot{H}$ denotes the time-derivative of the activation, whereas v_t is the transmembrane potential, obtained by solving the electrophysiology problem on $[0, t]$. In conclusion, the full mechanical problem reads as follows

$$\begin{cases} div(\mathbf{P}_{tot}(\mathbf{u}; H(v_t), \boldsymbol{\mu})) = 0 & \text{in } \Omega_0 \\ \mathbf{P}_{tot}(\mathbf{u}; H(v_t), \boldsymbol{\mu})\mathbf{n} = 0 & \text{on } \Gamma_{endo} \\ \mathbf{P}_{tot}(\mathbf{u}; H(v_t), \boldsymbol{\mu})\mathbf{n} + \alpha\mathbf{u} = \mathbf{0} & \text{on } \Gamma_{epi}, \Gamma_{base}, \end{cases} \tag{6.4}$$

where we impose homogeneus Neumann boundary conditions on the endocardium and Robin boundary conditions on the epicardium and on the base in order to avoid the presence of rigid motion. We point out that homogeneus Neumann boundary conditions on the endocardium are not physiological, since we are neglecting the pressure caused by the presence of blood in the ventricular chamber. However, when solving an electromechanical model, the pressure value is unknown, and thus we neglect the effect of the fluid on the ventricular deformation.

We observe that the full Piola tensor depends on the solution of the electrical problem, which acts as a forcing term for the system. Moreover, although the mechanical model is static, it depends on time through the electrical activation. We point out that in order to describe a full heart beat,

22

the mechanical problem has to be solved at different time-steps. Here, we focus on a time interval $[0, t_m]$, $t_m \geq 0$, that covers the first phase of the cardiac cycle; for each $\boldsymbol{\mu}$ we thus solve the electrophysiology problem until time t_m and we compute the corresponding $H(v_{t_m})$, required to solve problem (6.4) and then finally obtain the displacement $\mathbf{u}(\boldsymbol{\mu})$.

6.2.2   Numerical results

The numerical test is performed on a geometrical configuration representing an idealized human ventricle. The high fidelity approximation is built on a computational mesh (see Figure 10) with 3552 elements and 1181 vertices, resulting in a high-fidelity space V_h of dimension $N_h = 3543$.
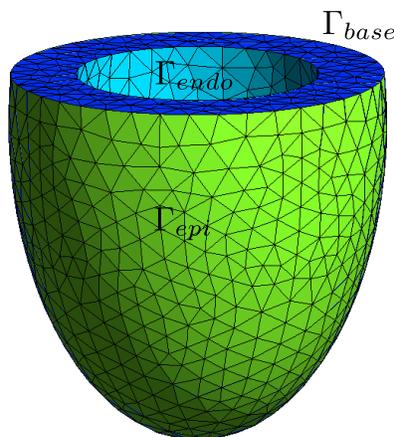
Figure 10: Idealized left ventricle computational grid.

We consider the following parameters:

- Fibers inclination $\theta_{max} \in [20°, 80°]$,
- Bulk modulus $\kappa \in [3 \cdot 10^5, 4 \cdot 10^5]$,
- Isotropic coefficient $a \in [3300, 3500]$.

We choose these parameters since they have a significant effect on the heart contraction. In particular, the fibers orientation is noticeably different from one patient to another and can have a crucial impact on the correct torsion and shortening of the ventricle. The Bulk modulus $\kappa > 0$ measures the material resistance to a uniform compression and plays an important role in the imposition of the material incompressibility. The isotropic coefficient is instead related to the stiffness of the cardiac muscle, and thus affects the ability of the ventricle to contract and the ejected quantity of blood. However, other parameters could be relevant from a clinical viewpoint, for instance the electrical conductivities of the myocardium, which affect the propagation of the electrical signal and, consequently, the correct heart contraction. Analysis related to these electrical parameters is currently ongoing.

For the case at hand, since the fibers inclination affects also the electrical equations, the electrophysiology problem has to be solved for every new parameter value during the online phase in order to compute the potential v, required to obtain the activation function. However, this task is computationally affordable, since solving the electrical problem by means of a FE approximation for the case at hand requires about 5 seconds. We underline that our method allows to consider parameters which are related with both the mechanical subproblem and the electrophysiology (such as the fibers inclination), which thus affect the mechanics through the solution of the electrical problem.

According to the analysis performed in section 6.1 the numerical results presented are obtained using the proposed hybrid method (see Algorithm 5). We show in Figure 11 the eigenvalues of the

correlation matrix related to the displacements, the residual vectors and the Jacobian matrices. In Figure 12 we also report the error of the DEIM and MDEIM approximation computed over a test sample of 40 parameters. We observe that the decay of both the eigenvalues and the DEIM and MDEIM errors is slower than the decay obtained for the previous test case; this is essentially due to the considerable complexity of this problem, originating from the highly nonlinear constitutive law, the coupling with the electrophysiology and the choice of the parameters considered.
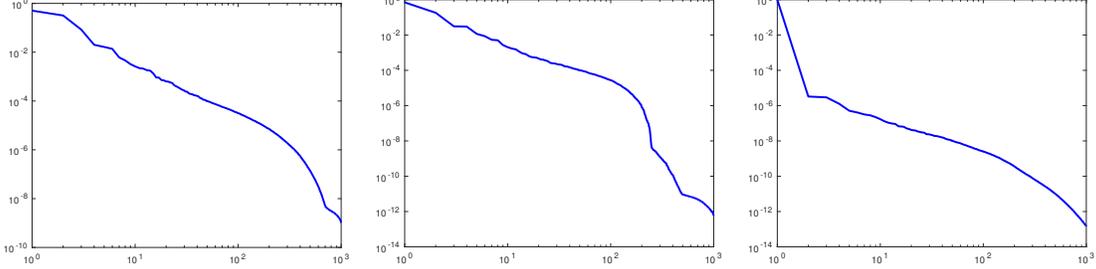
Figure 11: Normalized eigenvalues (blue) of the correlation matrix related to the displacement (left), to the Jacobian matrix (center) and to the residuals vector (right).
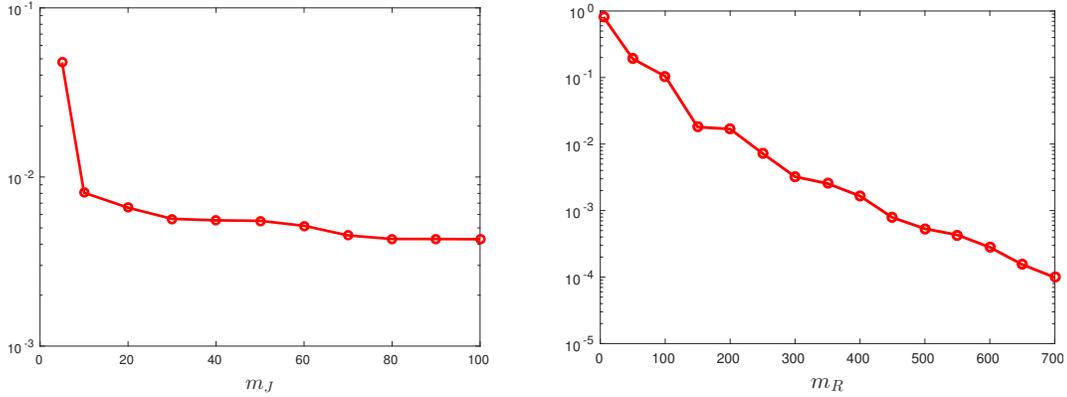
Figure 12: Average $||\cdot||_2$ error related to the MDEIM approximation of the Jacobian matrix (left) and to the DEIM approximation of the residuals vector (right).

In figures 13 and 14 we show respectively the displacement field and the corresponding deformations obtained for three different value of the parameters. As we are analyzing the initial phase of the heart beat, we observe small deformations, mainly concentrated in the ventricle apex, located at the bottom of the ventricle. However, the displacement field underlines the torsion typical of the myocardium contraction, caused by the presence of the muscular fibers.

We report in Figure 15 the high fidelity solutions, the RB solutions and the difference between them obtained for three different values of the parameters, randomly chosen during the online stage. We remark that the ROM is able to accurately approximate the high fidelity solution. Moreover, evaluating the average $\mathbf{X}_h$ error of the solution over a test sample of 40 parameters, we obtain a relative error equal to 10^{-3} with 40 basis functions (see Figure 16). In Figure 16 we report also the a posteriori error bound (5.2) computed over the same testing set of 40 parameters; the difference between the estimator and the effective error is less than 2 orders of magnitude.

We observe that the number of DEIM bases for the residual vector is remarkably higher with respect to the number of terms selected for the Jacobian matrix (see also Remark 1). Indeed, in order to guarantee the convergence of the Newton method for all parameters value we need

24
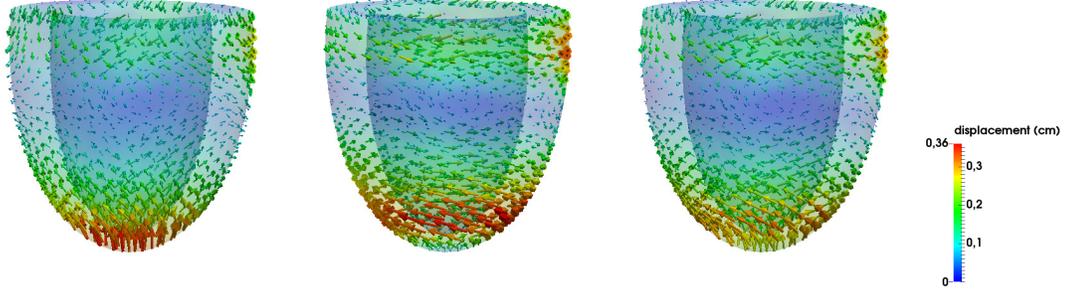
Figure 13: Displacement field obtained for three different parameter values ($\boldsymbol{\mu}_1 = [39°, 3.8 \cdot 10^5, 3465]$, $\boldsymbol{\mu}_2 = [52°, 3 \cdot 10^5, 3306]$, $\boldsymbol{\mu}_3 = [67°, 3.6 \cdot 10^5, 3400]$).
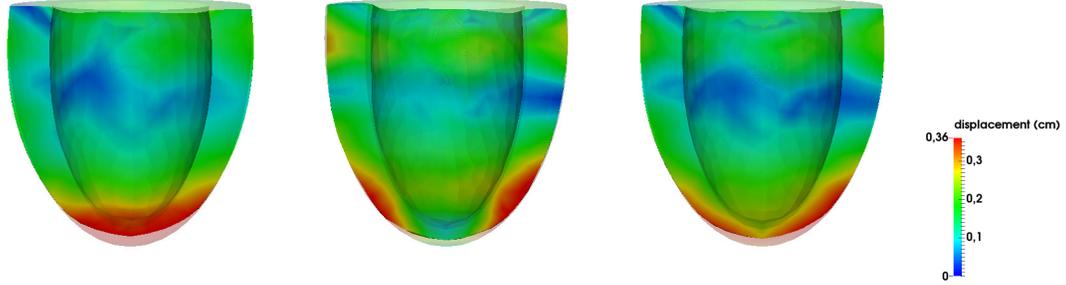
Figure 14: Displacement on a section of the ventricle for three different parameter values ($\boldsymbol{\mu}_1 = [39°, 3.8 \cdot 10^5, 3465]$, $\boldsymbol{\mu}_2 = [52°, 3 \cdot 10^5, 3306]$, $\boldsymbol{\mu}_3 = [67°, 3.6 \cdot 10^5, 3400]$).

about 500 terms and 5 terms for the residual and the Jacobian matrix approximation, respectively. However, a great reduction in computational time is achieved, since the effective bottleneck is the assembling of the Jacobian matrix. Actually, the RB problem takes about 4 seconds, while the FE problem requires about 2 minutes (see Table 6).

We report also in Figure 17 the reduced mesh obtained for the approximation of the Jacobian matrix. The number of elements of the reduced mesh is extremely lower than the number of elements of the full mesh, leading to a great reduction in the cost related to the assembling of the Jacobian matrix; this computation takes about $0.12s$ on the reduced mesh while it requires about $7s$ on the full mesh.

Number of FE dofs	3543	Number of RB dofs	20
POD tolerance	10^{-5}	DEIM tolerance	10^{-10}
Newton tolerance	10^{-3}	MDEIM tolerance	10^{-4}
Offline time	7h	Number of parameters	3
Residual DEIM terms	500	Jacobian MDEIM terms	5
FE time Jacobian assembling	$7s$	RB time Jacobian assembling	$0.12s$
FE time	$120s$	Online RB time	$4s$

Table 6: Numerical details (test case 2)
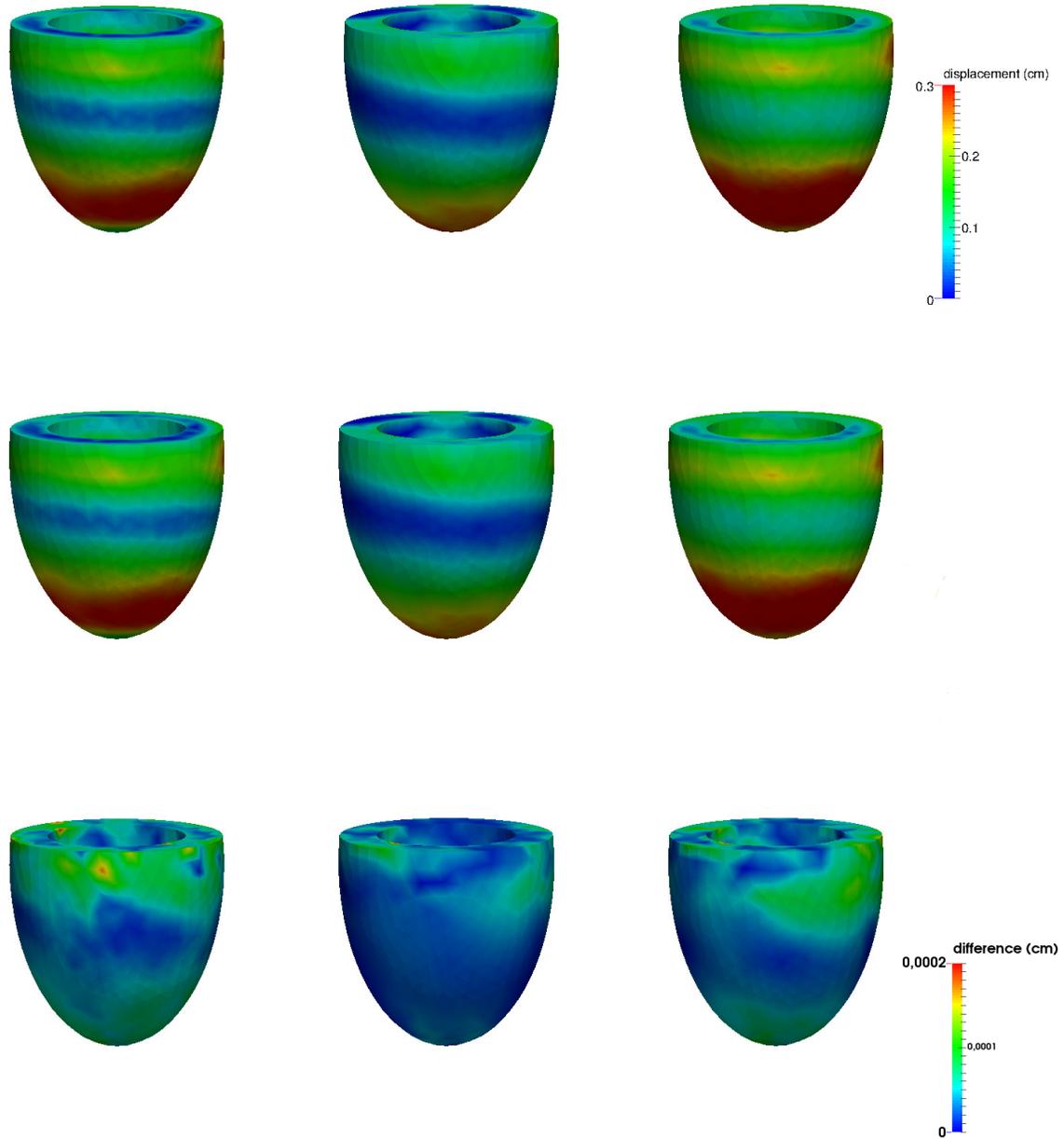
Figure 15: High-fidelity solutions (top), reduced order solutions (middle) and their differences (bottom) for three different parameter values ($\boldsymbol{\mu}_1 = [78°, 3.18 \cdot 10^5, 3335.7]$, $\boldsymbol{\mu}_2 = [46°, 3.58 \cdot 10^5, 3416.5]$, $\boldsymbol{\mu}_3 = [32°, 3.78 \cdot 10^5, 3456.7]$).
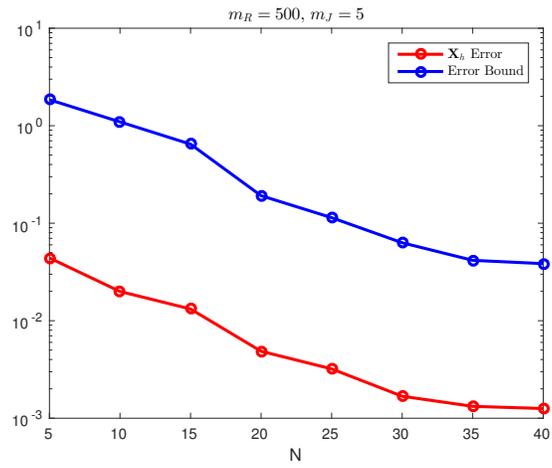
Figure 16: Average $\mathbf{X}_h$ error (red) and A posteriori error bound (blue) computed over a testing set of 40 parameters.
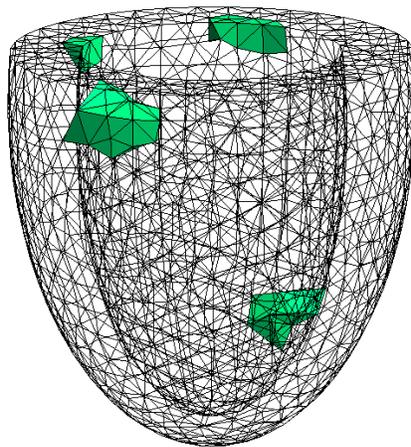
Figure 17: Reduced mesh for the approximation of the Jacobian matrix.

7 Conclusions

In this paper we have proposed a new reduction technique for nonlinear parametrized problems, which combines proper orthogonal decomposition for the selection of basis functions, Galerkin projection over a low dimensional subspace and a matrix discrete empirical interpolation (MDEIM) technique for efficiently handling the nonlinear terms. The resulting ROM allows to evaluate the problem solution at a very reduced computational cost, still retaining the same accuracy of the high fidelity model.

We have also proposed a new snapshots selection strategy able to provide accurate and fast online solutions, still retaining low offline computational times. We compared this strategy with two different methods, in order to highlight advantages and drawbacks related to the proposed approaches.

Moreover, we have characterized a new reliable a posteriori error bound, able to accurately estimate the difference between the high-fidelity and the reduced solution, taking into account separately the error components related to the Galerkin projection, the DEIM approximation of the residual vectors and the MDEIM approximation of the Jacobian matrices.

The effectiveness of the proposed framework has been assessed on two different test cases. In the first case we have performed a shear test on a cubic domain, relying on a Saint Venant-Kirchhoff material which is characterized by a polynomial nonlinearity. Then, we applied our method to the cardiac electromechanical problem, focusing on a time interval $[0, t_m], t_m \geq 0$, that covers the first phase of the heart beat. The extension to a complete heart beat is currently under investigation. This application demonstrates that our reduction strategy can be efficiently used also when treating problems characterized by very complex constitutive laws.

Appendix. Lipschitz constant derivation

We recall that the Piola tensor for a Saint Venant-Kirchoff material can be written under the following form

$$\mathbf{P} = \lambda tr(\mathbf{E})\mathbf{F} + 2\mu \mathbf{F}\mathbf{E},$$

being $\mathbf{E}$ the Lagrangian Green strain tensor and $\mathbf{F}$ the deformation gradient tensor. We thus obtain the following expression for the Jacobian of $\mathbf{P}$:

$$\langle J_{\mathbf{u}}(\mathbf{w}), \mathbf{z} \rangle = \lambda \int_{\Omega_0} \left[(\mathbf{F} : \nabla \mathbf{w})(\mathbf{F} : \nabla \mathbf{z}) + \frac{1}{2}(\mathcal{I}_1 - 3)(\nabla \mathbf{w} : \nabla \mathbf{z}) \right] d\Omega_0$$

$$+ \mu \int_{\Omega_0} \left[(\nabla \mathbf{w} \mathbf{C} : \nabla \mathbf{z}) + (\mathbf{F}\mathbf{F}^T \nabla \mathbf{w} : \nabla \mathbf{z}) - (\nabla \mathbf{w} : \nabla \mathbf{z}) + (\mathbf{F}\nabla \mathbf{w}^T \mathbf{F} : \nabla \mathbf{z}) \right] d\Omega_0.$$

In this section we want to show that $\mathbf{J}_h(\mathbf{V}\mathbf{u}_{N,m}(\boldsymbol{\mu}); \boldsymbol{\mu})$ is locally Lipschitz continuous at $\mathbf{u}_{N,m}(\boldsymbol{\mu})$, i.e., there exists $K_h^N(\boldsymbol{\mu}) > 0$ such that for all $\mathbf{v} \in \overline{B}_{r(\boldsymbol{\mu})}(\mathbf{V}\mathbf{u}_{N,m}(\boldsymbol{\mu}))$

$$||\mathbf{J}_h(\mathbf{V}\mathbf{u}_{N,m}(\boldsymbol{\mu}); \boldsymbol{\mu}) - \mathbf{J}_h(\mathbf{v}; \boldsymbol{\mu})||_{\mathbf{X}_h, \mathbf{X}_h^{-1}} \leq K_h^N(\boldsymbol{\mu})||\mathbf{V}\mathbf{u}_{N,m}(\boldsymbol{\mu}) - \mathbf{v}||_{\mathbf{X}_h}.$$

We can define

$$C = \sup_{\boldsymbol{\mu} \in \mathcal{D}} ||\mathbf{V}\mathbf{u}_{N,m}(\boldsymbol{\mu})||_{\mathbf{X}_h};$$

then, for each $\mathbf{v} \in \overline{B}_{r(\boldsymbol{\mu})}(\mathbf{V}\mathbf{u}_{N,m}(\boldsymbol{\mu}))$ it holds that

$$||\mathbf{v}||_{\mathbf{X}_h} \leq ||\mathbf{V}\mathbf{u}_{N,m}(\boldsymbol{\mu})||_{\mathbf{X}_h} + ||\mathbf{v} - \mathbf{V}\mathbf{u}_{N,m}(\boldsymbol{\mu})||_{\mathbf{X}_h} \leq C + r(\boldsymbol{\mu}).$$

For the sake of notation hereon we now omit the $\boldsymbol{\mu}$ dependence.

In order to derive the Lipschitz constant, we employ the following inequality

$$|\mathbf{F}^T(\mathbf{V}\mathbf{u}_{N,m})\mathbf{F}(\mathbf{V}\mathbf{u}_{N,m}) - \mathbf{F}^T(\mathbf{v})\mathbf{F}(\mathbf{v})| = |\mathbf{F}^T(\mathbf{V}\mathbf{u}_{N,m})(\mathbf{F}(\mathbf{V}\mathbf{u}_{N,m}) - \mathbf{F}(\mathbf{v})) + (\mathbf{F}^T(\mathbf{V}\mathbf{u}_{N,m}) - \mathbf{F}^T(\mathbf{v}))\mathbf{F}(\mathbf{v})|$$

$$\leq (2 + 2C + r)||\mathbf{V}\mathbf{u}_{N,m}(\boldsymbol{\mu}) - \mathbf{v}||_{\mathbf{X}_h}.$$

Then, it holds that

$$|\langle J_{\mathbf{Vu}_{N,m}}(\mathbf{w}), \mathbf{z}\rangle - \langle J_{\mathbf{v}}(\mathbf{w}), \mathbf{z}\rangle| = \lambda \int_{\Omega_0} (\mathbf{F}(\mathbf{Vu}_{N,m}) : \nabla \mathbf{w})(\mathbf{F}(\mathbf{Vu}_{N,m}) : \nabla \mathbf{z}) - (\mathbf{F}(\mathbf{v}) : \nabla \mathbf{w})(\mathbf{F}(\mathbf{v}) : \nabla \mathbf{z})d\Omega_0$$
$$+ \frac{1}{2}\lambda \int_{\Omega_0} (\mathcal{I}_1(\mathbf{Vu}_{N,m}) - \mathcal{I}_1(\mathbf{v}))(\nabla \mathbf{w} : \nabla \mathbf{z})d\Omega_0$$
$$+ \mu \int_{\Omega_0} (\nabla \mathbf{w}(\mathbf{F}^T(\mathbf{Vu}_{N,m})\mathbf{F}(\mathbf{Vu}_{N,m}) - \mathbf{F}^T(\mathbf{v})\mathbf{F}(\mathbf{v})) : \nabla \mathbf{z})d\Omega_0$$
$$+ \mu \int_{\Omega_0} (\nabla \mathbf{w}(\mathbf{F}(\mathbf{Vu}_{N,m})\mathbf{F}^T(\mathbf{Vu}_{N,m}) - \mathbf{F}(\mathbf{v})\mathbf{F}^T(\mathbf{v})) : \nabla \mathbf{z})d\Omega_0$$
$$+ \mu \int_{\Omega_0} ((\mathbf{F}(\mathbf{Vu}_{N,m})\nabla \mathbf{w}\mathbf{F}^T(\mathbf{Vu}_{N,m}) - \mathbf{F}(\mathbf{v})\nabla \mathbf{w}\mathbf{F}^T(\mathbf{v})) : \nabla \mathbf{z})d\Omega_0$$
$$\leq (2\lambda + 3\mu)(2 + 2C + r(\boldsymbol{\mu}))||\mathbf{Vu}_{N,m} - \mathbf{v}||_{\mathbf{X}_h}||\mathbf{w}||_{\mathbf{X}_h}||\mathbf{z}||_{\mathbf{X}_h}. \tag{7.1}$$

Exploiting the definition of the $||\cdot||_{\mathbf{X}_h, \mathbf{X}_h^{-1}}$ norm, it is possible to show that (7.1) finally implies

$$K_h^N(\boldsymbol{\mu}) = (2\lambda + 3\mu)(2 + 2C + r(\boldsymbol{\mu})).$$

References

[1] D. Amsallem, J. Cortial, K. Carlberg, and C. Farhat. A method for interpolating on manifolds structural dynamics reduced-order models. *International Journal for Numerical Methods in Engineering*, 80(9):1241–1258, 2009.

[2] H. Antil, M. Heinkenschloss, and D. C. Sorensen. Application of the discrete empirical interpolation method to reduced order modeling of nonlinear and parametric systems. In A. Quarteroni and G. Rozza, editors, *Reduced Order Methods for Modeling and Computational Reduction*, volume 9 of *Modeling, Simulation and Applications*, pages 101–136. Springer, Switzerland, 2014.

[3] P. Astrid, S. Weiland, K. Willcox, and T. Backx. Missing point estimation in models described by proper orthogonal decomposition. *IEEE Trans. Automat. Control*, 53:2237–2251, 2008.

[4] 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. *Comptes Rendus Mathematique*, 339(9):667–672, 2004.

[5] M. Boel, O.J. Abilez, A.N. Assar, C.K. Zarins, and E. Kuhl. in Vitro/in Silico Characterization of Active and Passive Stresses in Cardiac Muscle. *International Journal for Multiscale Computational Engineering*, 10(2):171–188, 2012.

[6] A. Bueno-Orovio, E.M. Cherry, and F.H. Fenton. Minimal model for human ventricular action potentials in tissue. *Journal of Theoretical Biology*, 253(3):544–560, 2008.

[7] K. Carlberg, C. Bou-Mosleh, and C. Farhat. Efficient non-linear model reduction via a least-squares Petrov-Galerkin projection and compressive tensor approximations. *International Journal for Numerical Methods in Engineering*, 86:155–181, 2011.

[8] K. Carlberg, C. Farhat, J. Cortial, and D. Amsallem. The GNAT method for nonlinear model reduction: effective implementation and application to computational fluid dynamics and turbulent flows. *J. Comp. Phys.*, 242:623–647, 2013.

[9] K. Carlberg, R. Tuminaro, and P. Boggs. Efficient structure-preserving model reduction for nonlinear mechanical systems with application to structural dynamics. *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, pages 1–16, 2012.

[10] K. Carlberg, R. Tuminaro, and P. Boggs. Preserving Lagrangian structure in nonlinear model reduction with application to structural dynamics. *SIAM J. Sci. Comput.*, 37(2):B153–B184, 2015.

[11] S. Chaturantabut and D.C. Sorensen. Nonlinear Model Reduction via Discrete Empirical Interpolation. *SIAM Journal on Scientific Computing*, 32(5):2737–2764, 2010.

[12] P.G. Ciarlet. *Linear and nonlinear functional analysis with applications*, volume 130. SIAM, 2013.

[13] P. Colli Franzone, L.F. Pavarino, and S. Scacchi. *Mathematical cardiac electrophysiology*, volume 13 of *Modeling, Simulation and Applications (MS&A) Series*. Springer-Verlag Italia, Milano, 2014.

[14] T.S.E. Eriksson, A.J. Prassl, G. Plank, and G.A. Holzapfel. Influence of myocardial fiber/sheet orientations on left ventricular mechanical contraction. *Mathematics and Mechanics of Solids*, page 1081286513485779, 2013.

[15] R. Everson and L. Sirovich. Karhunen-Loeve procedure for gappy data. *Journal of the Optical Society of America A*, 12(8):1657, 1995.

[16] S. Goktepe and E. Kuhl. Electromechanics of the heart: a unified approach to the strongly coupled excitation-contraction problem. *Computational Mechanics*, 45(2-3):227–243, November 2009.

[17] G.H. Golub and C.F. Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.

[18] M. Grepl, Y. Maday, N. Nguyen, and A.T. Patera. Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. *Esaim Math. Model. Numer. Anal.*, 41(3):575–605, 2007.

[19] J. Helfenstein, M. Jabareen, E. Mazza, and S. Govindjee. On non-physical response in models for fiber-reinforced hyperelastic materials. *International Journal of Solids and Structures*, 47(16):2056–2061, 2010.

[20] G. Holzapfel and R.W. Ogden. Constitutive modelling of passive myocardium: a structurally based framework for material characterization. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 367(1902):3445–3475, 2009.

[21] D.B.P. Hyunh and A.T. Patera. Reduced-basis approximation and a posteriori error estimation for stress intensity factors. *Archives of Computational Methods in Engineering*, 72(10):341–368, 1219-1259.

[22] P. Kerfriden, P. Gosselet, S. Adhikari, and S. Bordas. Bridging Proper Orthogonal Decomposition methods and augmented Newton-Krylov algorithms: an adaptive model order reduction for highly nonlinear mechanical problems. *Computer Methods in Applied Mechanics and Engineering*, 200:850–866, 2011.

[23] P. Krysl, S. Lall, and J.E. Marsden. Dimensional model reduction in non-linear finite element dynamics of solids and structures. *International Journal for numerical methods in engineering*, 51(4):479–504, 2001.

[24] 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.

[25] A. Manzoni. An efficient computational framework for reduced basis approximation and a posteriori error estimation of parametrized Navier-Stokes flows. *ESAIM Math. Modelling Numer. Anal.*, 48:1199–1226, 2014.

[26] A. Manzoni, A. Quarteroni, and G. Rozza. Shape optimization of cardiovascular geometries by reduced basis methods and free-form deformation techniques. *Int. J. Numer. Methods Fluids*, 70(5):646–670, 2012.

[27] R. Milani, A. Quarteroni, and G. Rozza. Reduced basis method for linear elasticity problems with many parameters. *Comput. Methods Appl. Mech. Engrg.*, 197:4812–4829, 2008.

[28] M.P. Nash and P.J. Hunter. Computational Mechanics of the Heart. *Journal of Elasticity*, 61:113–141, 2001.

[29] F. Negri. *Efficient reduction techniques for the simulation and optimization of parametrized systems: analysis and applications*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne, 2015.

[30] F. Negri, A. Manzoni, and D. Amsallem. Efficient model reduction of parametrized systems by matrix discrete empirical interpolation. *J. Comp. Phys.*, 303:431–454, 2015.

[31] N.C. Nguyen and J. Peraire. An efficient reduced-order modeling approach for non-linear parametrized partial differential equations. *Int. J. Numer. Meth. Engrg.*, 76(1):27–55, 2008.

[32] P. Pathmanathan, S.J. Chapman, D.J. Gavaghan, and J.P. Whiteley. Cardiac electromechanics: The effect of contraction model on the mathematical problem and accuracy of the numerical scheme. *Quarterly Journal of Mechanics and Applied Mathematics*, 63(3):375–399, 2010.

[33] S. Pezzuto. *Mechanics of the Heart - Constitutive Issues and Numerical Experiments*. PhD thesis, Politecnico di Milano, 2013.

[34] A. Quarteroni, T. Lassila, S. Rossi, and R. Ruiz-baier. Integrated heart - coupling multiscale and multiphysics models for the simulation of the cardiac function. *Mox Report*, 05/2016.

[35] A. Quarteroni, A. Manzoni, and F. Negri. *Reduced Basis Methods for Partial Differential Equations. An Introduction*, volume 92 of *Unitext*. Springer, 2016.

[36] A. Radermacher and S. Reese. POD-based model reduction with empirical interpolation applied to nonlinear elasticity. *Int. J. Numer. Meth. Engrg.*, 2015. in press.

[37] S. Rossi, T. Lassila, R. Ruiz-Baier, A. Sequeira, and A. Quarteroni. Thermodynamically consistent orthotropic activation model capturing ventricular systolic wall thickening in cardiac electromechanics. *European Journal of Mechanics, A/Solids*, 48, 2013.

[38] C. Sansour. On the physical assumptions underlying the volumetric-isochoric split and the case of anisotropy. *European Journal of Mechanics, A/Solids*, 27(1):28–39, 2008.

[39] N.P. Smith, D.P. Nickerson, E.J. Crampin, and P.J. Hunter. Multiscale computational modelling of the heart. *Acta Numerica*, 13:371, 2004.

[40] R. Stefanescu and A. Sandu. Efficient approximation of sparse jacobians for time-implicit reduced order models. Technical report, Virginia Polytechnic Institute and State University, CSTR-19/2015, 2014.

[41] R. Stefanescu, A. Sandu, and I.M. Navon. POD/DEIM reduced-order strategies for efficient four dimensional variational data assimilation. *J. Comput. Phys.*, 295:569–595, 2015.

[42] D. Wirtz, D.C. Sorensen, and B. Haasdonk. A posteriori error estimation for deim reduced nonlinear dynamical systems. *SIAM J. Sci. Comput.*, 36(2):A311–A338, 2014.

[43] D. Xiao, F. Fang, A.G. Buchan, C.C. Pain, I.M. Navon, J. Du, and G. Hu. Non-linear model reduction for the Navier-Stokes equations using residual DEIM method. *J. Comput. Phys.*, 263:1–18, 2014.

[44] L. Zanon and K. Veroy-Grepl. The reduced basis method applied to a finite deformation problem in elasticity. Proceeding of 3rd ECCOMAS Young Investigators Conference, 2015.

MOX Technical Reports, last issues

13/2016   Guerciotti, B; Vergara, C; Ippolito, S; Quarteroni, A; Antona, C; Scrofani, R.
Computational study of the risk of restenosis in coronary bypasses

12/2016   Bartezzaghi, A.; Dedè, L.; Quarteroni, A.
Isogeometric Analysis of Geometric Partial Differential Equations

11/2016   Zhu, S.; Dedè, L.; Quarteroni, A.
Isogeometric Analysis and proper orthogonal decomposition for the acoustic wave equation

10/2016   Flemisch, B.; Fumagalli, A.; Scotti, A.
A review of the XFEM-based approximation of flow in fractured porous media

08/2016   Dassi, F.; Perotto, S.; Si, H.; Streckenbach, T.
A priori anisotropic mesh adaptation driven by a higher dimensional embedding

09/2016   Rizzo, C.B.; de Barros, F.P.J.; Perotto, S.; Oldani, L.; Guadagnini, A.
Relative impact of advective and dispersive processes on the efficiency of POD-based model reduction for solute transport in porous media

07/2016   Pacciarini, P.; Gervasio, P.; Quarteroni, A.
Spectral Based Discontinuous Galerkin Reduced Basis Element Method for Parametrized Stokes Problems

06/2016   Micheletti, S.; Perotto, S.; Signorini, M.
Anisotropic mesh adaptation for the generalized Ambrosio-Tortorelli functional with application to brittle fracture

05/2016   Alfio Quarteroni, A.; Lassila, T.; Rossi, S.; Ruiz-Baier, R.
Integrated Heart - Coupling multiscale and multiphysics models for the simulation of the cardiac function

01/2016   Domanin, M.; Buora, A.; Scardulla, F.; Guerciotti, B.; Forzenigo, L.; Biondetti, P.; Vergara, C.
Computational fluid-dynamic analysis of carotid bifurcations after endarterectomy: closure with patch graft versus direct suture