

A matrix DEIM technique for model reduction of nonlinear parametrized problems in cardiac mechanics ¹

Diana Bonomi^a, Andrea Manzoni^{b,*}, Alfio Quarteroni^b

^a*MOX, Dipartimento di Matematica, Politecnico di Milano,
P.za Leonardo da Vinci 32, I-20133 Milano, Italy*

^b*CMCS-MATH-SB,
Ecole Polytechnique Fédérale de Lausanne,
Station 8, CH-1015 Lausanne, Switzerland*

Abstract

When relying on Newton iterations to solve nonlinear problems in the context of Reduced Basis (RB) methods, the assembling of the RB arrays during the online stage depends on the dimension of the underlying high-fidelity approximation. This is more of an issue 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 combined to evaluate both the residual vector and the Jacobian matrix very efficiently in the case of complex parametrized nonlinear mechanical problems. We compare this strategy with the classical DEIM approach and we derive a posteriori error estimates on the solution accounting for the contribution of DEIM/MDEIM errors. The effectiveness of the proposed framework is assessed on quasi-static nonlinear problems. In particular, we consider a nonlinear elasticity problem defined on a cube and a mechanical model describing heart contraction, for an idealized left ventricle geometry. The latter is a coupled problem, in which the activation of the heart contraction is given by the solution of an electrophysiology model. Our numerical results show that MDEIM is preferable to the classical DEIM, both in terms of efficiency and accuracy.

Keywords: reduced basis method; discrete empirical interpolation; nonlinear mechanics; cardiac mechanics; a posteriori error estimation

1. Introduction

Reducing the computational time for the numerical solution of complex parametrized mechanical problems is crucial in many engineering applications, e.g. in biomechanics, where parameters may be related to initial and/or boundary conditions, physical coefficients or forcing terms. Being able to rapidly approximate the solution of such problems allows to investigate the parameter-to-solution map and, ultimately, the impact of significant parameters on the modeled system; this is e.g. required when dealing with sensitivity analysis, uncertainty quantification, and parameter estimation. Reduced order modeling (ROM) techniques are suitable to achieve this goal, as they

¹We are grateful to Federico Negri and Luca Dedè (EPFL) for their careful remarks and to Toni Lassila (University of Sheffield) for some inspiring discussions at the initial stage of this work.

*Corresponding author. Tel.: +41 21 693 5509

Email addresses: diana.bonomi@polimi.it (Diana Bonomi), andrea.manzoni@epfl.ch (Andrea Manzoni), alfio.quarteroni@epfl.ch (Alfio Quarteroni)

provide accurate solutions at a greatly reduced computational cost. This is usually achieved by seeking the solution in a subspace of much smaller dimension N than the one, N_h , of the original finite dimensional space employed by a full order model (FOM). 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 e.g. on greedy algorithms or proper orthogonal decomposition to generate the reduced space [28, 45].

Being able to assemble and solve the reduced problem at a very reduced computational cost is possible if the ROM arrays are independent of the FOM dimension N_h ; in this case, we can 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 featuring an affine parametric dependence. In the case of more complex, nonaffine parametric dependence, assembling the ROM for any new parameter would require to assemble the FOM arrays first and 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. The construction of a reduced space and the system approximation 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 [5, 34] to approximate nonaffinely parametrized functions, as well as tensor components, appearing in full-order operators, prior to reduction; see, e.g., [26, 36] for applications in the framework of RB methods and [19] for an extension of EIM to the case where the function to be interpolated is the PDE operator itself. The discrete variant of EIM, the so-called Discrete Empirical Interpolation Method (DEIM), was originally developed in [13] to efficiently deal with nonlinear problems, but has also been applied to nonaffinely parametrized linear operators [3]. 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 [12, 55] 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 [39] to address the efficient reduction of nonaffinely parametrized steady and time-dependent linear PDEs, with applications to PDE-constrained optimization and coupled problems. In the last years, MDEIM has been exploited in a few applications for the approximation of Jacobian matrices, see e.g. [52, 55], nevertheless focusing on semilinear problems or PDEs characterized by mild polynomial nonlinearities.

In the case of parametrized nonlinear PDEs, to develop an efficient ROM, each step of the online Newton method must be independent of the dimension N_h of the underlying FOM. This goal can be achieved by means of the so called hyper-reduction or system approximation, which aim to recover an approximate affine structure of nonlinear terms to guarantee an efficient offline-online decomposition. Preliminary applications dealing with semilinear PDEs, as well as nonlinear PDEs featuring low-order polynomial nonlinearities, have usually exploited EIM and DEIM to deal with nonlinear terms, see e.g. [26]; more recent applications can be found, e.g., in [56, 53, 38]. In the finite element context, a variant of DEIM was also developed in [54] under the name of unassembled DEIM (UDEIM). A slightly different alternative, the so-called *best point interpolation* method, has been addressed in [40]. 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 necessarily using EIM/DEIM, even if this implies decomposing nonlinear terms into the sum of $O(N^2)$ terms, see e.g. [35].

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, *a priori hyper-reduction* [49, 50], *missing point estimation* [4] and *gappy POD* [21] have also been developed, e.g. within the so-called Gauss-Newton with approximated tensors (GNAT) method, see e.g. [10].

By extending a recent methodology proposed in [39] for the efficient system reduction of parametrized linear PDEs, in this paper we show how to take advantage of both DEIM and MDEIM to perform, at a purely algebraic level, hyper-reduction of three-dimensional, fully nonlinear parametrized problems arising in computational mechanics. We apply for the first time the MDEIM technique to nonlinear parametrized problems and we show how this technique can be efficiently used to reduce a mechanical problem characterized by a highly nonlinear strain energy function, defined on complex 3D configurations. The proposed framework allows to approximate both the Jacobian matrix and the residual vector and can be combined with RB methods whose reduced spaces are obtained thanks to either greedy or POD techniques. In addition, we propose a new strategy for snapshots selection based on the introduction of a semi hyper-reduced problem, able to provide a very substantial computational saving. Hereon, we will refer to the proposed DEIM/MDEIM framework (being DEIM applied on residual vectors and MDEIM on Jacobian matrices) as to the *MDEIM method* for the sake of compactness. We compare this approach to the classical *DEIM method*, where the reduced Jacobian matrix is obtained by deriving the DEIM approximation of the residual vector, in order to highlight advantages and drawbacks related to both techniques. Furthermore, we develop a computable, residual-based a posteriori error bound that also accounts for the DEIM/MDEIM approximation errors.

We apply our reduced framework to two problems arising in nonlinear mechanics: a shear test on a Saint-Venant-Kirchhoff material with time-dependent boundary conditions, and a coupled electro-mechanical problem for the simulation of the cardiac contraction. What makes the cardiac mechanical problem extremely difficult is the presence of an exponential strain energy function and the description of the structure in terms of muscular fibers and sheets, resulting in a complex model showing highly nonlinear terms. Moreover, in this work we also consider the coupling with a (time-dependent) electrical model, which describes the propagation of the signal triggering the mechanical heart contraction. This yields the solution of a *many-query* problem, since at each time step the mechanical problem can be treated as a quasi-static problem, where time can be considered as a generic parameter. The *many-query* nature of the cardiac electro-mechanics makes the proposed framework extremely appropriate for the sake of computational cost reduction. We point out that, in this work, even if we consider a full electromechanical model, our reduction technique is only applied to the mechanical subproblem. This choice is motivated by the fact that the solution of the electrical subproblem is significantly fast compared to the mechanical one.

We point out that few problems in structural mechanics have been tackled by ROMs so far, because of the complexity involved by nonlinear constitutive laws. We mention former contributions dealing with linear elasticity problems (see, e.g. [30, 37, 2]) and nonlinear elasticity of two-dimensional structures, featuring e.g. polynomial nonlinearities [58], or homogenization techniques [57]. As for three dimensional problems, preliminary applications of ROM to structural mechanics can be found in [32], where the focus is on time dependent problems, even if no hyper-reduction techniques are considered. The construction of a ROM which preserves the Lagrangian structure of a mechanical system can also be found in [33, 11]. More recent contributions to non-

linear three-dimensional structural problems can be found e.g. in [9, 31]. In particular, an energy preserving method has been applied in [22, 23] for the reduction of nonlinear structural problems. In [46] a POD-DEIM method has been recently applied to nonlinear structural problems, dealing however 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 capable to significantly reduce the computational time, still providing the accuracy of the FOM solution.

The structure of the paper is as follows. In section 2 we remind some basic notions of the RB method for parametrized nonlinear problems and in section 3 we recall the POD technique used to construct the reduced space. In section 4 we first introduce the classical DEIM approach for the approximation of the residual vectors; then we show how the MDEIM technique can be extended to nonlinear parametrized problems. Details of the reduced framework are then illustrated and a new snapshots selection strategy is proposed. A new reliable a posteriori error bound is provided in section 5. In section 6 we present two different applications to highly nonlinear mechanical problems, in order to assess the computational performance of the proposed strategy. In particular, we consider the full cardiac electromechanical model and we show how the proposed ROM is able to efficiently approximate the mechanical subproblem, on an idealized left ventricle configuration. Conclusions are drawn in section 7.

2. RB methods for nonlinear parametrized PDEs

In this work we consider nonlinear parametrized mechanical problems under the following form: find $\mathbf{u}(\boldsymbol{\mu}) \in V$ such that

$$R(\mathbf{u}(\boldsymbol{\mu}); \boldsymbol{\mu}) = 0 \quad \text{in } V'. \quad (2.1)$$

$V = V(\Omega_0)$ is a suitable Hilbert space and V' its dual; $\Omega_0 \subset \mathbb{R}^n$, $n = 2, 3$ is the (reference) spatial domain; $\boldsymbol{\mu} = (\mu_1, \dots, \mu_d) \in \mathcal{D}$ is a parameter vector and $\mathcal{D} \subset \mathbb{R}^d$ the parameter domain. To find an approximate solution of (2.1) we introduce a finite-dimensional approximation space $V_h \subset V$, defined by

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

here X_h^r denotes the finite element (FE) space 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 approximate the weak form of (2.1) by the FE method and this represents our FOM. To solve the associated nonlinear algebraic system

$$\mathbf{R}(\mathbf{u}_h(\boldsymbol{\mu}); \boldsymbol{\mu}) = \mathbf{0} \quad (2.2)$$

we apply the Newton method: given $\mathbf{u}_h^{(0)}(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$, for $k \geq 1$, find $\delta \mathbf{u}_h(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ s.t.

$$\begin{cases} \mathbf{J}(\mathbf{u}_h^{(k-1)}(\boldsymbol{\mu}); \boldsymbol{\mu}) \delta \mathbf{u}_h(\boldsymbol{\mu}) = -\mathbf{R}(\mathbf{u}_h^{(k-1)}(\boldsymbol{\mu}); \boldsymbol{\mu}), \\ \mathbf{u}_h^{(k)}(\boldsymbol{\mu}) = \mathbf{u}_h^{(k-1)}(\boldsymbol{\mu}) + \delta \mathbf{u}_h(\boldsymbol{\mu}) \end{cases} \quad (2.3)$$

and iterate until **the relative residual is such that** $\|\mathbf{R}(\mathbf{u}_h^{(k)}(\boldsymbol{\mu}); \boldsymbol{\mu})\|_2 / \|\mathbf{R}(\mathbf{u}_h^{(0)}(\boldsymbol{\mu}); \boldsymbol{\mu})\|_2 < \varepsilon$, where $\varepsilon > 0$ denotes a small, given tolerance. In (2.3), $\forall \mathbf{v}_h \in \mathbb{R}^{N_h}$

$$[\mathbf{J}(\mathbf{v}_h(\boldsymbol{\mu}); \boldsymbol{\mu})]_{ij} = \langle J(\tilde{\mathbf{v}}_h(\boldsymbol{\mu}); \boldsymbol{\mu})[\boldsymbol{\varphi}_j], \boldsymbol{\varphi}_i \rangle, \quad [\mathbf{R}(\mathbf{v}_h(\boldsymbol{\mu}); \boldsymbol{\mu})]_i = R(\tilde{\mathbf{v}}_h(\boldsymbol{\mu}); \boldsymbol{\mu})(\boldsymbol{\varphi}_i), \quad i, j = 1, \dots, N_h$$

denote the components of the Jacobian matrix $\mathbf{J}(\mathbf{v}_h(\boldsymbol{\mu}); \boldsymbol{\mu}) \in \mathbb{R}^{N_h \times N_h}$ and of the residual vector $\mathbf{R}(\mathbf{v}_h(\boldsymbol{\mu}); \boldsymbol{\mu}) \in \mathbb{R}^{N_h}$, respectively, where $J(\tilde{\mathbf{v}}_h(\boldsymbol{\mu}); \boldsymbol{\mu}) = DR(\tilde{\mathbf{v}}_h(\boldsymbol{\mu}); \boldsymbol{\mu})$ denotes the Fréchet differential of R with respect to the first argument, evaluated at $\tilde{\mathbf{v}}_h(\boldsymbol{\mu})$. Here, $\tilde{\mathbf{v}}_h(\boldsymbol{\mu}) = \sum_{i=1}^{N_h} v_{h,i}(\boldsymbol{\mu}) \boldsymbol{\varphi}_i$

is a generic element of the FE space, $\mathbf{v}_h(\boldsymbol{\mu}) = [v_{h,1}(\boldsymbol{\mu}), \dots, v_{h,N_h}(\boldsymbol{\mu})]$ is its vector counterpart and $\{\varphi_i, i = 1, \dots, N_h\}$ denote the basis functions of the FOM space V_h .

To reduce the computational complexity involved by (2.3), we introduce a projection-based ROM. To this aim, we select N vectors of \mathbb{R}^{N_h} – which we refer to as reduced basis (RB) and which form the matrix $\mathbf{V} \in \mathbb{R}^{N_h \times N}$ – and consider the subspace \mathbf{V}_N of \mathbb{R}^{N_h} generated by them. Each vector is indeed made by the components that are the nodal values of suitable finite element functions, called *snapshots*, or RB functions. Then, for any $\boldsymbol{\mu} \in \mathcal{P}$, we look for a vector $\mathbf{u}_N(\boldsymbol{\mu}) \in \mathbb{R}^N$ that satisfies

$$\mathbf{V}^T \mathbf{R}(\mathbf{V} \mathbf{u}_N(\boldsymbol{\mu}); \boldsymbol{\mu}) = \mathbf{0}, \quad (2.4)$$

that is,

$$\mathbf{R}_N(\mathbf{V} \mathbf{u}_N(\boldsymbol{\mu}); \boldsymbol{\mu}) = \mathbf{0}, \quad (2.5)$$

where $\mathbf{R}_N(\mathbf{V} \mathbf{u}_N(\boldsymbol{\mu}); \boldsymbol{\mu}) = \mathbf{V}^T \mathbf{R}(\mathbf{V} \mathbf{u}_N(\boldsymbol{\mu}); \boldsymbol{\mu})$. Problem (2.4) (or, equivalently, (2.5)) represents the Galerkin projection of (2.2) onto the subspace \mathbf{V}_N , and from now on will be referred to as the ROM; the resulting linear combination $\mathbf{V} \mathbf{u}_N(\boldsymbol{\mu})$ of the RB functions finally gives the RB approximation to $\mathbf{u}_h(\boldsymbol{\mu})$. A suitable strategy to select the RB functions is postponed to Sect. 3.

The Galerkin RB problem (2.4) can be solved by the Newton method: given $\mathbf{u}_N^{(0)}(\boldsymbol{\mu}) \in \mathbb{R}^N$, for $k \geq 1$, find $\delta \mathbf{u}_N(\boldsymbol{\mu}) \in \mathbb{R}^N$ s.t.

$$\begin{cases} \mathbf{V}^T \mathbf{J}(\mathbf{V} \mathbf{u}_N^{(k-1)}(\boldsymbol{\mu}); \boldsymbol{\mu}) \mathbf{V} \delta \mathbf{u}_N(\boldsymbol{\mu}) = -\mathbf{V}^T \mathbf{R}(\mathbf{V} \mathbf{u}_N^{(k-1)}(\boldsymbol{\mu}); \boldsymbol{\mu}), \\ \mathbf{u}_N^{(k)}(\boldsymbol{\mu}) = \mathbf{u}_N^{(k-1)}(\boldsymbol{\mu}) + \delta \mathbf{u}_N(\boldsymbol{\mu}), \end{cases} \quad (2.6)$$

and iterate until the relative reduced residual is $\|\mathbf{V}^T \mathbf{R}(\mathbf{V} \mathbf{u}_N^{(k)}(\boldsymbol{\mu}); \boldsymbol{\mu})\|_2 / \|\mathbf{V}^T \mathbf{R}(\mathbf{V} \mathbf{u}_N^{(0)}(\boldsymbol{\mu}); \boldsymbol{\mu})\|_2 < \tilde{\varepsilon}$, being $\tilde{\varepsilon} > 0$ a small, given tolerance. Note that (2.6)₁ can be equivalently rewritten as

$$\mathbf{J}_N(\mathbf{V} \mathbf{u}_N^{(k-1)}(\boldsymbol{\mu}); \boldsymbol{\mu}) \delta \mathbf{u}_N(\boldsymbol{\mu}) = -\mathbf{R}_N(\mathbf{V} \mathbf{u}_N^{(k-1)}(\boldsymbol{\mu}); \boldsymbol{\mu}) \quad (2.7)$$

where $\mathbf{J}_N(\mathbf{V} \mathbf{w}_N; \boldsymbol{\mu}) = \mathbf{V}^T \mathbf{J}(\mathbf{V} \mathbf{w}_N; \boldsymbol{\mu}) \mathbf{V}$ for any $\mathbf{w}_N \in \mathbb{R}^N$.

Assuming that the FOM (2.2) admits a solution $\mathbf{u}_h(\boldsymbol{\mu})$ for any $\boldsymbol{\mu} \in \mathcal{P}$, existence and uniqueness of the solution to the ROM (2.5) follow by the classical Brezzi-Rappaz-Raviart theory; see, e.g., [8, Theorem 7.1]. In particular, we assume that $\mathbf{J}(\mathbf{u}_h(\boldsymbol{\mu}))$ is invertible for any $\boldsymbol{\mu} \in \mathcal{P}$, Lipschitz continuous at $\mathbf{u}_h(\boldsymbol{\mu})$, and positive (semi)definite, and that $\lim_{N \rightarrow N_h} \inf_{\mathbf{w}_N \in \mathbb{R}^N} \|\mathbf{u}_h(\boldsymbol{\mu}) - \mathbf{V} \mathbf{w}_N\|_2 = 0$. Regarding instead the solution of problem (2.7), the Jacobian matrix has to be non-singular at each iteration, in order to guarantee the well-posedness of the linear system to be solved (see e.g. [16, 59] for rigorous analysis on the convergence of Newton method). We observe that the mere non-singularity of \mathbf{J} does not warrant a priori the invertibility of $\mathbf{J}_N = \mathbf{V}^T \mathbf{J} \mathbf{V}$. Further assumptions on \mathbf{J} , e.g. that it is a symmetric positive definite matrix, would make the job. However, even when dealing with problems for which \mathbf{J} is not symmetric positive definite, we have never numerically incurred in a non-singular $\mathbf{J}_N = \mathbf{V}^T \mathbf{J} \mathbf{V}$.

Remark 1. Suitable globalization techniques (see e.g. [41]) can enhance the chance of convergence of the Newton method when initial solutions are not close enough to the limit solution. Although employing these techniques is feasible also at the reduced-order level, so far the standard Newton algorithm has performed well in all our numerical tests.

3. Solution-space reduction: POD technique

In this work, the POD technique is used to compute the reduced basis \mathbf{V} through the so-called method of snapshots, as well as for the construction of both DEIM and MDEIM bases. We now sketch a general description of the POD technique. Let \mathbf{z} be a map defined on the parameter

space \mathcal{D} with values in \mathbb{R}^{N_h} and let $\mathcal{M}_{\mathbf{z}} = \{\mathbf{z}(\boldsymbol{\mu}) \in \mathbb{R}^{N_h} \mid \boldsymbol{\mu} \in \mathcal{D}\}$ the manifold identified by the image of \mathbf{z} . The goal of POD is to approximate $\mathcal{M}_{\mathbf{z}}$ with a low-dimensional optimal linear space. To do so, starting from a set of n_s snapshots $\mathbf{z}_i = \mathbf{z}(\boldsymbol{\mu}_i)$, with $\boldsymbol{\mu}_i \in \mathcal{D}$, $i = 1, \dots, n_s$, POD constructs a low-dimensional subspace of \mathbb{R}^{N_h} (spanned by the so-called POD basis) retaining as much as possible of the snapshots content of information. Here $\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_{n_s}$ are randomly sampled points of \mathcal{D} ; other strategies, such as e.g. *latin hypercube sampling* or *sparse grid* techniques, could be exploited especially for high-dimensional parameter spaces. The POD basis is obtained performing the singular value decomposition (SVD) of the snapshot matrix

$$\mathbf{Z} = [\mathbf{z}_1 \quad \mathbf{z}_2 \quad \dots \quad \mathbf{z}_{n_s}] \in \mathbb{R}^{N_h \times n_s},$$

thus yielding the factorization

$$\mathbf{Z} = \boldsymbol{\Phi} \boldsymbol{\Sigma} \boldsymbol{\Lambda}^T,$$

where $\boldsymbol{\Phi} \in \mathbb{R}^{N_h \times n_s}$, $\boldsymbol{\Lambda} \in \mathbb{R}^{n_s \times N_h}$ and $\boldsymbol{\Sigma} = \text{diag}(\sigma_1, \sigma_2, \dots)$, where $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{n_s} \geq 0$ are the n_s singular values of \mathbf{Z} . The so-called POD basis $\boldsymbol{\Phi}_N$ of dimension $N_h \leq N$ is obtained by collecting the first N columns of $\boldsymbol{\Phi}$ (i.e. the first N left singular vectors of \mathbf{Z}), corresponding to the first N (largest) singular values; we can set the basis dimension N as the minimum integer such that

$$\frac{\sum_{i=1}^N \sigma_i^2}{\sum_{i=1}^{n_s} \sigma_i^2} \geq 1 - \varepsilon_{POD}, \quad (3.1)$$

for a given small tolerance $\varepsilon_{POD} > 0$. The reduced basis provided by POD is optimal, in the sense that that $\boldsymbol{\Phi}_N$ satisfies

$$\boldsymbol{\Phi}_N = \underset{\substack{\{\mathbf{y}_1, \dots, \mathbf{y}_N\} \\ \text{orthonormal in } \mathbb{R}^{N_h}}}]{\text{argmin}} \sum_{i=1}^{n_s} \|\mathbf{z}_i - \Pi_{Y_N} \mathbf{z}_i\|_2^2,$$

where $\Pi_{Y_N} \mathbf{z}$ denotes the orthogonal projection of $\mathbf{z} \in \mathbb{R}^{N_h}$ onto $Y_N = \text{span}\{\mathbf{y}_1, \dots, \mathbf{y}_N\}$ 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. Note that no a posteriori error estimation (like in the greedy case, see, e.g., [45, Chapter 7] for further details) is used to select the snapshots. POD is based on a purely spectral argument and it is efficient provided that a sufficiently rich set of snapshots has been selected in order to "cover" the manifold $\mathcal{M}_{\mathbf{z}}$ as much as possible. We summarize the POD technique in Algorithm 3.1; note that different norms can also be used, see e.g. [45, Chapter 6] for further details.

Algorithm 3.1 POD algorithm

INPUT: Snapshots matrix $\mathbf{Z} \in \mathbb{R}^{N_h \times n_s}$, tolerance ε_{POD}

OUTPUT: $\boldsymbol{\Phi}_N$

- 1: Perform the SVD of \mathbf{Z} : $\mathbf{Z} = \boldsymbol{\Phi} \boldsymbol{\Sigma} \boldsymbol{\Lambda}^T$
 - 2: Set the basis dimension N as the minimum integer such that condition (3.1) is fulfilled
 - 3: Construct $\boldsymbol{\Phi}_N = [\boldsymbol{\phi}_1, \dots, \boldsymbol{\phi}_N]$ selecting the first N columns of the matrix $\boldsymbol{\Phi}$.
-

4. System approximation: DEIM/MDEIM techniques

The reduced Newton problem (2.6) still depends on the high fidelity dimension N_h . Indeed, both 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, and then 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 efficiency. Several works [42, 6, 53, 14, 32] showed that the DEIM technique originally proposed in [13], properly combined with POD, allows to meet this goal. In this paper, we propose a MDEIM technique and a suitable snapshots selection strategy, which allow to obtain a better computational speedup when considering highly nonlinear, possibly coupled, problems characterized by a complex parameters dependence. In this section we first shortly describe the classical DEIM, then we introduce our MDEIM alternative technique, in order to highlight advantages and drawbacks related to the different approaches.

4.1. Discrete Empirical Interpolation Method

For the problem at hand, at each Newton step, DEIM [13] allows to efficiently express the residual vector as a linear combination of (possibly few) $\boldsymbol{\mu}$ -independent terms so that, the $\boldsymbol{\mu}$ -dependent weights of this combination can be efficiently computed by solving an interpolation problem. 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, $\forall k \geq 1$

$$\mathbf{R}(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}); \boldsymbol{\mu}) \approx \mathbf{R}_m(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}); \boldsymbol{\mu}) = \boldsymbol{\Phi}_R \boldsymbol{\theta}_R(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}), \boldsymbol{\mu}) = \sum_{i=1}^{m_R} \theta_R^i(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}), \boldsymbol{\mu}) \phi_R^i, \quad (4.1)$$

where $\boldsymbol{\theta}_R(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}), \boldsymbol{\mu}) \in \mathbb{R}^{m_R}$ 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 $\{\mathbf{R}(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}_i); \boldsymbol{\mu}_i), i = 1, \dots, n_s\}$. Here we have decided to take a (large) number of snapshots equal to n_s , although it is not required to compute as many snapshots as in the case of the problem solution. The same consideration holds for the evaluation of Jacobian matrix snapshots, too. Since we are dealing with a nonlinear problem, in order to obtain the residual snapshots $\mathbf{R}(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}_i); \boldsymbol{\mu}_i)$, we need to solve the reduced problem (2.6) for different values of $\boldsymbol{\mu}$ and, at each Newton iteration, to store the computed residual vectors.

The coefficient vector $\boldsymbol{\theta}_R(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}), \boldsymbol{\mu})$ can be evaluated for each new value of $\boldsymbol{\mu}$ by imposing m_R interpolation constraints on a subset $\wp = [\wp_1, \dots, \wp_{m_R}]$ of entries of $\mathbf{R}(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}); \boldsymbol{\mu})$ (the so-called magic points, see e.g. [34]), selected by the DEIM algorithm, see Algorithm 4.1. For ease of notation, we introduce the matrix

$$\mathbf{P} = [\mathbf{e}_{\wp_1}, \dots, \mathbf{e}_{\wp_{m_R}}] \in \mathbb{R}^{N_h \times m_R},$$

where $\mathbf{e}_{\wp_i} = [0, \dots, 0, 1, 0, \dots, 0]^T \in \mathbb{R}^{N_h}$ is the \wp_i -th column of the identity matrix $I \in \mathbb{R}^{N_h \times N_h}$, for $i = 1, \dots, m_R$. The coefficient vector $\boldsymbol{\theta}_R(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}), \boldsymbol{\mu})$ is then obtained as the solution of the system arising from the imposition of the interpolation constraints

$$\mathbf{P}^T \boldsymbol{\Phi}_R \boldsymbol{\theta}_R(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}), \boldsymbol{\mu}) = \mathbf{P}^T \mathbf{R}(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}); \boldsymbol{\mu}), \quad (4.2)$$

where $\mathbf{P}^T \boldsymbol{\Phi}_R$ and $\mathbf{P}^T \mathbf{R}(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}); \boldsymbol{\mu})$ are the restrictions of $\boldsymbol{\Phi}_R$ and $\mathbf{R}(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}); \boldsymbol{\mu})$ to the subset of indices \wp , respectively.

Remark 2. The interpolation condition (4.2) can be generalized to the case where more magic points ($\#col(\mathbf{P}) > m_R$) than basis functions are considered, yielding the so-called gappy POD reconstruction [4, 9]. In this case the coefficient vector is obtained by solving a least-squares problem under the form

$$\boldsymbol{\theta}_R(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}), \boldsymbol{\mu}) = \arg \min_{\mathbf{y} \in \mathbb{R}^{m_R}} \|\mathbf{P}^T \mathbf{R}(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}); \boldsymbol{\mu}) - \mathbf{P}^T \boldsymbol{\Phi}_R \mathbf{y}\|^2.$$

In both cases the subset of entries of the vector \mathbf{R} to be evaluated has to be very small compared to the vector dimension N_h ; the remaining entries are reconstructed either by interpolation or by a least-square strategy. DEIM traditionally hinges upon the solution of an interpolation problem, and we have pursued this strategy, too, although a gappy POD strategy can also be employed. Performing least-squares gappy reconstruction as opposed to interpolation often improves accuracy without significantly affecting the CPU performance provided the function being approximated is easy to evaluate; see, e.g., [9] for a detailed comparison. However, as it will be shown below, for the cases at hand the cost of assembling the residual vector (and, more importantly, the Jacobian matrix) has a stronger impact on the CPU performance, thus motivating the use of interpolation.

The approximation of the reduced residual vector in (2.6) can be obtained by projecting (4.1) onto the reduced space yielding

$$\mathbf{V}^T \mathbf{R}(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}); \boldsymbol{\mu}) \approx \mathbf{V}^T \boldsymbol{\Phi}_R (\mathbf{P}^T \boldsymbol{\Phi}_R)^{-1} \mathbf{P}^T \mathbf{R}(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}); \boldsymbol{\mu}) := \mathbf{R}_{N,m}(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}); \boldsymbol{\mu}). \quad (4.3)$$

All the quantities which do not depend on $\boldsymbol{\mu}$ can be precomputed offline; in the online stage we only need to assemble $\mathbf{P}^T \mathbf{R}(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}); \boldsymbol{\mu})$, which is the restriction of the residual to the subset of DEIM nodes. Hereon, the subscript m denotes a *generic* hyper-reduced quantity, similarly to the subscript N indicating a reduced quantity. Depending on the case at hand, m_R and m_J will be used to denote the DEIM reduction and the separate reduction associated with MDEIM, respectively; however, in order not to make notation heavier, this additional subscript is omitted whenever it is clear which quantity is referred to.

The classical DEIM approach to deal with nonlinear problems (see e.g. [6, 14, 32, 42]) then would approximate the reduced Jacobian $\mathbf{J}_N(\mathbf{V}\mathbf{w}_N(\boldsymbol{\mu}); \boldsymbol{\mu})$, $\forall \mathbf{w}_N \in \mathbb{R}^N$ by the derivative of the reduced approximated residual vector $\mathbf{R}_{N,m}(\mathbf{V}\mathbf{w}_N; \boldsymbol{\mu})$ (appearing at the right-hand side of (4.3)), yielding

$$\begin{aligned} \tilde{\mathbf{J}}_{N,m}(\mathbf{V}\mathbf{w}_N(\boldsymbol{\mu}); \boldsymbol{\mu}) &= \frac{\partial \mathbf{R}_{N,m}(\mathbf{V}\mathbf{w}_N(\boldsymbol{\mu}); \boldsymbol{\mu})}{\partial \mathbf{w}_N} \\ &= \mathbf{V}^T \boldsymbol{\Phi}_R (\mathbf{P}^T \boldsymbol{\Phi}_R)^{-1} \mathbf{P}^T \frac{\partial \mathbf{R}(\mathbf{V}\mathbf{w}_N(\boldsymbol{\mu}); \boldsymbol{\mu})}{\partial \mathbf{w}_N} \\ &= \mathbf{V}^T \boldsymbol{\Phi}_R (\mathbf{P}^T \boldsymbol{\Phi}_R)^{-1} \mathbf{P}^T \mathbf{J}(\mathbf{V}\mathbf{w}_N(\boldsymbol{\mu}); \boldsymbol{\mu}) \mathbf{V}. \end{aligned} \quad (4.4)$$

As for the residual vector, we can precompute the $\boldsymbol{\mu}$ -independent quantities offline, while online we have to assemble $\mathbf{P}^T \mathbf{J}(\mathbf{V}\mathbf{w}_N(\boldsymbol{\mu}); \boldsymbol{\mu}) \in \mathbb{R}^{m_R \times N_h}$, that is the restriction of the Jacobian matrix to the rows which correspond to the indices contained in \wp . Consequently, we need to assemble online, at each Newton step, a matrix of dimension $m_R \times N_h$, an operation which turns to be quite expensive, above all when m_R becomes large. We point out that, since the reduced Jacobian matrix is obtained as the derivative of the reduced residual, the classical DEIM can be regarded as an exact Newton method, although on the following approximated version of problem (2.6),

$$\mathbf{R}_{N,m}(\mathbf{V}\mathbf{u}_N(\boldsymbol{\mu}); \boldsymbol{\mu}) = \mathbf{0}, \quad (4.5)$$

where $\mathbf{R}_{N,m}(\mathbf{V}\mathbf{w}_N; \boldsymbol{\mu})$ denotes the reduced approximated residual vector defined in (4.3). As a result, the k -th Newton iteration for problem (4.5) reads as follows:

$$\tilde{\mathbf{J}}_{N,m}(\mathbf{V}\mathbf{u}_{N,m}^{(k-1)}(\boldsymbol{\mu}); \boldsymbol{\mu}) \delta \mathbf{u}_N(\boldsymbol{\mu}) = -\mathbf{R}_{N,m}(\mathbf{V}\mathbf{u}_{N,m}^{(k-1)}(\boldsymbol{\mu}); \boldsymbol{\mu}).$$

Remark 3. Constructing a basis able to retain as much as possible of the content of information related to the residual vectors is a challenging task. Indeed, the entries of the residual vectors

usually vary over a wide range of values as they approach the tolerance imposed in the stopping criterion of the Newton method, while assuming large values at the first Newton steps. Thus, a large number m_R of DEIM terms can be needed to accurately approximate the residuals in the ROM. Using normalized or rescaled residuals does not seem to be the cure. Rather, we can 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 viable alternative regarding the number of terms m_R . Even more importantly, when a large number of DEIM terms is obtained, the computational time required by the assembling of the reduced Jacobian matrix also increases substantially, since it depends on m_R . Moreover, since assembling the Jacobian matrix is by far the most expensive operation of each Newton iteration, large assembling times can negatively affect the speedup provided by the ROM.

Algorithm 4.1 DEIM algorithm (as originally proposed in [13])

INPUT: $\Phi = [\phi_1, \dots, \phi_m] \in \mathbb{R}^{n \times m}$ made by linearly independent columns
OUTPUT: $\wp = [\wp_1, \dots, \wp_m] \in \mathbb{R}^m$

- 1: $\wp_1 = \text{maxpos}\{\phi_1\}$
- 2: $\Phi = [\phi_1]$, $\mathbf{P} = [\mathbf{e}_{\wp_1}]$
- 3: **for** $k = 2, \dots, m$ **do**
- 4: Solve $(\mathbf{P}^T \Phi) \mathbf{c} = (\mathbf{P}^T \phi_k)$
- 5: $\mathbf{r} = \phi_k - \Phi \mathbf{c}$
- 6: $\wp_k = \text{maxpos}\{\mathbf{r}\}$
- 7: $\Phi \leftarrow [\Phi \ \phi_k]$, $\mathbf{P} \leftarrow [\mathbf{P} \ \mathbf{e}_{\wp_k}]$
- 8: **end for**

4.2. MDEIM for Jacobian approximation

In this section we show how to exploit a matrix version of DEIM (MDEIM) as alternative to the classical DEIM approach, in order to perform hyper-reduction of the Jacobian matrices arising in (2.6). The idea is to directly approximate the reduced Jacobian matrix $\mathbf{V}^T \mathbf{J}(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}); \boldsymbol{\mu}) \mathbf{V}$ exploiting a variant of the DEIM technique, where the approximation of the residual and the Jacobian rely on two different DEIM basis. Hence, MDEIM offers the advantage to choose a different number of DEIM indices for the residual vector and the Jacobian matrix, so that even if m_R is large, a small number of DEIM indices associated with the Jacobian matrix can be selected, ultimately yielding a greater reduction of the computational cost. Indeed, the $\boldsymbol{\mu}$ -dependent Jacobian matrix usually varies in a significantly smaller range of scales compared to the residual vector², so that a small number of terms is usually sufficient to obtain an accurate approximation.

MDEIM provides an approximation of the Jacobian matrix $\mathbf{J}(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}); \boldsymbol{\mu}) \in \mathbb{R}^{N_h \times N_h}$ under the form

$$\mathbf{J}(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}); \boldsymbol{\mu}) \approx \mathbf{J}_m(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}); \boldsymbol{\mu}) = \sum_{i=1}^{m_J} \theta_J^i(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}), \boldsymbol{\mu}) \mathbf{J}^i, \quad (4.6)$$

²Otherwise said, from step to step when executing Newton method, the Jacobian matrix experiences much less relevant variations with respect to the ones shown by the residual vector. Indeed, the magnitude of this latter ranges across different scales, finally required to be under a desired tolerance; on the other hand, linearizing the problem around the solutions computed at different Newton iterates yield matrices that are closer and closer as Newton method fulfills the stopping criterion.

where $\{\mathbf{J}^i \in \mathbb{R}^{N_h \times N_h}, i = 1, \dots, m_J\}$ denotes a set of $\boldsymbol{\mu}$ -independent matrices that can be computed once for all and $\boldsymbol{\theta}_J(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}), \boldsymbol{\mu}) = [\theta_J^1(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}), \boldsymbol{\mu}), \dots, \theta_J^{m_J}(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}), \boldsymbol{\mu})]$ a coefficient vector. This approximation is obtained by defining

$$j(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}); \boldsymbol{\mu}) = \text{vec}(\mathbf{J}(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}); \boldsymbol{\mu})) \in \mathbb{R}^{N_h^2}$$

as the vector obtained by stacking the columns of $\mathbf{J}(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}); \boldsymbol{\mu})$, and approximating $j(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}); \boldsymbol{\mu})$ by its DEIM counterpart

$$j(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}); \boldsymbol{\mu}) \approx j_m(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}); \boldsymbol{\mu}) = \boldsymbol{\Phi}_J \boldsymbol{\theta}_J(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}), \boldsymbol{\mu}), \quad \boldsymbol{\Phi}_J = [\phi_1, \dots, \phi_n] \in \mathbb{R}^{N_h^2 \times m_J}.$$

Then, the matrices \mathbf{J}^i can be computed transforming each column $\phi_i \in \mathbb{R}^{N_h^2}$ of $\boldsymbol{\Phi}_J$ into a matrix $\mathbf{J}^i \in \mathbb{R}^{N_h \times N_h}$ by reverting the vec operation, as $\mathbf{J}^i = \text{vec}^{-1}(\phi_i)$, so that $\mathbf{J}_m(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}); \boldsymbol{\mu}) = \text{vec}^{-1}(j_m(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}); \boldsymbol{\mu}))$. The basis $\boldsymbol{\Phi}_J$ and the coefficient vector $\boldsymbol{\theta}_J(\boldsymbol{\mu})$ are determined following the same procedure used for the residual vectors, relying on a set of snapshots $\{\mathbf{J}(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}_i); \boldsymbol{\mu}_i), i = 1, \dots, n_s\}$, evaluated on the reduced solution.

The reduced Jacobian matrix in (2.6) can thus be approximated in the following way:

$$\mathbf{V}^T \mathbf{J}(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}); \boldsymbol{\mu}) \mathbf{V} \approx \mathbf{V}^T \mathbf{J}_m(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}); \boldsymbol{\mu}) \mathbf{V} =: \mathbf{J}_{N,m}(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}); \boldsymbol{\mu}). \quad (4.7)$$

Remark 4. Note that all matrices are stored in a sparse format. As a result, the actual dimension of the vectorized matrices is n_{nx} rather than N_h^2 , where n_{nx} denotes the number of nonzero entries of the matrix. See e.g. [39, 52] for further details.

Remark 5. For nonsingular matrices, the Bauer-Fike theorem³ guarantees that, as m_J increases, the singular values of the approximated matrix $\mathbf{J}_m(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}); \boldsymbol{\mu})$ become closer and closer to the singular values of the original matrix $\mathbf{J}(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}); \boldsymbol{\mu})$; see e.g. [25]. This property ensures that the approximate matrix $\mathbf{J}_m(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}); \boldsymbol{\mu})$ is nonsingular, and thus invertible at each Newton step, provided that a sufficiently large number of basis functions is chosen and that $\mathbf{J}(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}); \boldsymbol{\mu})$ is nonsingular too. Further details on the applicability of the Bauer-Fike theorem in the MDEIM context can be found in [39].

Since we use two different independently computed bases to approximate the residual vector and the Jacobian matrix, the reduced Jacobian matrix (4.7) arising from the MDEIM approximation is not the exact Jacobian corresponding to the reduced DEIM residual $\mathbf{R}_{N,m}(\mathbf{V}\mathbf{u}_N(\boldsymbol{\mu}); \boldsymbol{\mu})$ defined in (4.5). Relying on the MDEIM approximation for the Jacobian matrix thus yields a quasi-Newton method for problem (4.5), for which at the k -th step the following system

$$\mathbf{J}_{N,m}(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}); \boldsymbol{\mu}) \delta \mathbf{u}_{N,m}(\boldsymbol{\mu}) = -\mathbf{R}_{N,m}(\mathbf{V}\mathbf{u}_N(\boldsymbol{\mu}); \boldsymbol{\mu}) \quad (4.8)$$

is solved instead than (4.8). Using a quasi-Newton method might require, in principle, a higher number of iterations to get convergence with respect to an exact Newton algorithm; nevertheless, our numerical tests show that having $m_J \ll m_R$ yields substantial computational savings.

³The Bauer-Fike theorem (see, e.g., [25, Theorem 7.2.2]) concerns stability of the eigenvalues of a matrix. If η is an eigenvalue of $A + E \in \mathbb{R}^{n \times n}$ and $X^{-1}AX = \text{diag}(\lambda_1, \dots, \lambda_n)$, where $\lambda_1, \dots, \lambda_n$ denotes the spectrum of A , $\lambda(A)$, and the columns of X are the corresponding eigenvectors, and E is a perturbation matrix, then

$$\min_{\lambda \in \lambda(A)} |\lambda - \eta| \leq \kappa_p(X) \|E\|_p$$

where $\|\cdot\|_p$ denotes any of the p -norms and $\kappa_p(X)$ is the p -condition number of X .

4.3. Efficient assembling on a reduced mesh

The DEIM algorithm selects a small subset of grid nodes of the original mesh. 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}_φ can be computed by simply integrating the residual only on the quadrature points belonging to those mesh elements which provide a non-zero contribution to the entries φ ; this set of elements is usually referred to as reduced mesh [10]. Hence, to compute the coefficient vector $\boldsymbol{\theta}_R(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}), \boldsymbol{\mu})$, we only need to evaluate \mathbf{R} on the reduced mesh. When using the classical DEIM approach, also the Jacobian matrix (4.4) has to be assembled on the same reduced mesh elements. Instead, by using the MDEIM approach, we need to use two different reduced meshes, associated to the approximation of the residual vector and the Jacobian matrix, respectively, the latter being coarser than the former, as already pointed in Remark 3. Indeed, if we assume (4.1) and (4.6), the reduced problem (2.6) can be replaced by the following hyper-reduced problem: given $\mathbf{u}_{N,m}^{(0)}(\boldsymbol{\mu}) \in \mathbb{R}^N$, at each Newton step $k \geq 1$ we seek $\delta\mathbf{u}_{N,m}^{(k)}(\boldsymbol{\mu}) \in \mathbb{R}^N$, such that

$$\begin{cases} \sum_{i=1}^{m_J} \theta_J^i(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}), \boldsymbol{\mu}) \mathbf{V}^T \mathbf{J}^i \mathbf{V} \delta\mathbf{u}_{N,m}(\boldsymbol{\mu}) = - \sum_{i=1}^{m_R} \theta_R^i(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}), \boldsymbol{\mu}) \mathbf{V}^T \phi_R^i \\ \mathbf{u}_{N,m}^{(k)}(\boldsymbol{\mu}) = \mathbf{u}_{N,m}^{(k-1)}(\boldsymbol{\mu}) + \delta\mathbf{u}_{N,m}(\boldsymbol{\mu}), \end{cases} \quad (4.9)$$

and iterate until $\|\sum_{i=1}^{m_R} \theta_R^i(\boldsymbol{\mu}) \mathbf{V}^T \phi_R^i\|_2 < \tilde{\varepsilon}$. We remark that, in order to correctly integrate the Jacobian matrix and the residual on the reduced mesh, at each Newton step we need the solution entries corresponding to the subsets $\tilde{\varphi}_R$ and $\tilde{\varphi}_J$ which contain all the nodes belonging to the elements of the two reduced meshes. We report in figure 1 the sketch of a reduced mesh in the case of linear finite elements; DEIM nodes belonging to φ are highlighted as red squares, while the nodes belonging to $\tilde{\varphi}$ are those marked with both a red square and a blue circle.

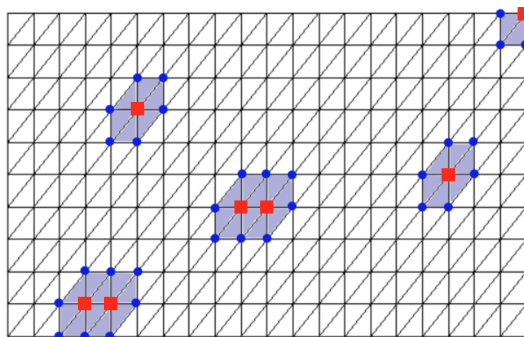


Figure 1: Sketch of a reduced mesh in a simple two-dimensional case (using linear finite elements)

On the other hand, during the online stage, we need to evaluate $\mathbf{u}|_{\tilde{\varphi}_R}$ and $\mathbf{u}|_{\tilde{\varphi}_J}$, that are the restriction of the solution to the subsets $\tilde{\varphi}_R \in \mathbb{N}^{\tilde{m}_R}$ and $\tilde{\varphi}_J \in \mathbb{N}^{\tilde{m}_J}$, respectively; to efficiently obtain these quantities we rely on some useful matrices preassembled offline, see Sect. 4.5.

4.4. A new snapshots selection strategy

We recall that, in order to construct the POD bases Φ_R and Φ_J for the DEIM and the MDEIM approximations of the residual and the Jacobian matrix, we need in principle to solve the reduced problem (2.6) n_s times to form the snapshot sets $\{\mathbf{R}(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}_i); \boldsymbol{\mu}_i), i = 1, \dots, n_s\}$ and $\{\mathbf{J}(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}_i); \boldsymbol{\mu}_i), i = 1, \dots, n_s\}$, both depending on the reduced solution. This latter has to be computed by solving (2.6) for $\boldsymbol{\mu} = \boldsymbol{\mu}_i$, which unfortunately requires to reassemble $\mathbf{R}(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}); \boldsymbol{\mu})$

and $\mathbf{J}(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}); \boldsymbol{\mu})$ at each Newton step, and then to project them onto the reduced space to obtain the corresponding ROM arrays $\mathbf{V}^T \mathbf{R}(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}); \boldsymbol{\mu})$ and $\mathbf{V}^T \mathbf{J}(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}); \boldsymbol{\mu}) \mathbf{V}$.

We propose a new strategy to collect the snapshots (see Algorithm 4.2), based on the introduction of a semi hyper-reduced problem involving only a MDEIM approximation of the Jacobian matrix, still relying on an exact residual vector: given $\mathbf{u}_{N,m}^{(0)}(\boldsymbol{\mu}) \in \mathbb{R}^N$, for $k \geq 1$, we search $\delta \mathbf{u}_{N,m}^{(k)}(\boldsymbol{\mu}) \in \mathbb{R}^N$ such that

$$\begin{cases} \sum_{i=1}^{m_J} \boldsymbol{\theta}_J^i(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}), \boldsymbol{\mu}) \mathbf{V}^T \mathbf{J}^i \mathbf{V} \delta \mathbf{u}_{N,m}(\boldsymbol{\mu}) = -\mathbf{V}^T \mathbf{R}(\mathbf{V}\mathbf{u}_{N,m}^{(k-1)}(\boldsymbol{\mu})) \\ \mathbf{u}_{N,m}^{(k)}(\boldsymbol{\mu}) = \mathbf{u}_{N,m}^{(k-1)}(\boldsymbol{\mu}) + \delta \mathbf{u}_{N,m}(\boldsymbol{\mu}), \end{cases} \quad (4.10)$$

and iterate until $\|\mathbf{V}^T \mathbf{R}(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}); \boldsymbol{\mu})\| < \tilde{\varepsilon}$. Solving this problem for any $\boldsymbol{\mu} = \boldsymbol{\mu}_i$ is significantly faster than solving problem (2.6), since \mathbf{J} is assembled only onto the reduced mesh. Not only, problem (4.10) requires almost the same effort of the full hyper-reduced problem (4.9), since the assembling of the Jacobian matrix takes almost the full time of a Newton iteration. Generating the residual snapshots according to this method then allows to correctly approximate the residual vectors using a small number of bases, still with low offline computational costs.

Algorithm 4.2 Basis Construction algorithm

INPUT: n_s combinations of parameters $\{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_{n_s}\}$
OUTPUT: \mathbf{V} , Φ_J , Φ_R

- 1: **for** $i = 1, \dots, n_s$ **do**
- 2: Solve problem (2.3) for $\boldsymbol{\mu}_i$
- 3: At each Newton iteration k :
 - $\mathbf{U} \leftarrow [\mathbf{U} \quad \mathbf{u}_{h,i}^k(\boldsymbol{\mu}_i)]$, $\mathbf{U}_J \leftarrow [\mathbf{U}_J \quad \text{vec}(\mathbf{J}(\mathbf{u}_{h,i}^{k-1}(\boldsymbol{\mu}_i), \boldsymbol{\mu}_i))]$
- 4: **end for**
- 5: $\mathbf{V} = \text{POD}(\mathbf{U}, \varepsilon)$, $\Phi_J = \text{POD}(\mathbf{U}_J, \varepsilon_J)$
- 6: **for** $i = 1, \dots, n_s$ **do**
- 7: Solve problem (4.10) for $\boldsymbol{\mu}_i$
- 8: At each Newton iteration k :
 - $\mathbf{U}_R \leftarrow [\mathbf{U}_R \quad \mathbf{R}(\mathbf{V}\mathbf{u}_N^{k-1}(\boldsymbol{\mu}_i), \boldsymbol{\mu}_i)]$
- 9: **end for**
- 10: $\Phi_R = \text{POD}(\mathbf{U}_R, \varepsilon_R)$.

Remark 6. We point out that the proposed MDEIM strategy requires an additional effort with respect to the classical DEIM approach, in order to store the Jacobian snapshots and compute Φ_J . Nevertheless, using MDEIM allows to save offline computational costs by considering the basis construction strategy proposed in Algorithm 4.2. Therefore, the offline cost of the MDEIM strategy is comparable to the cost associated to the classical DEIM approach.

Remark 7. A possible way to reduce the offline cost could be obtained by performing the solution-space approximation and the system reduction simultaneously, by storing the snapshots of the residual vectors and, eventually, of the Jacobian matrices when computing the snapshot solution of each high-fidelity problem for constructing the POD basis \mathbf{V} . This strategy has been widely used in the RB literature, see e.g. [13, 46]; however, in some cases, this method can provide a worse 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. An alternative option could be to compute the reduced Jacobian using finite differences of the reduced residual, as shown, e.g., in [46]. The associated quasi-Newton method for the is no longer a projection-based ROM; nevertheless, a possible investigation of this approach within the proposed framework will make the object of a further work.

4.5. Offline/Online algorithms

The entire computational procedure related to the MDEIM approach is summarized in Algorithms 4.3 and 4.4.

Algorithm 4.3 Offline stage

INPUT: n_s combinations of parameters $\{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_{n_s}\}$

OUTPUT: \mathbf{V} , $\boldsymbol{\Phi}_J$, $\boldsymbol{\Phi}_R$, $\boldsymbol{\wp}_R$, $\boldsymbol{\wp}_J$, \mathbf{S}_R , \mathbf{S}_J

1: Compute basis \mathbf{V} , $\boldsymbol{\Phi}_R$, $\boldsymbol{\Phi}_J$ using Algorithm 4.2

2: Assemble offline structures:

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

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

3: $\boldsymbol{\wp}_R = \text{DEIM}(\boldsymbol{\Phi}_R)$, $\boldsymbol{\wp}_J = \text{DEIM}(\boldsymbol{\Phi}_J)$

4: Construct the reduced meshes and compute the restriction matrices

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

5: Compute $\mathbf{S}_R = \tilde{\mathbf{P}}_R^T \mathbf{V} \in \mathbb{R}^{\tilde{m}_R \times N}$ and $\mathbf{S}_J = \tilde{\mathbf{P}}_J^T \mathbf{V} \in \mathbb{R}^{\tilde{m}_J \times N}$.

Algorithm 4.4 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

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

4: Compute $\mathbf{R}_N^{(k)}(\boldsymbol{\mu}) = \sum_{i=1}^{m_R} \boldsymbol{\theta}_R(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}), \boldsymbol{\mu}) \mathbf{R}_N^i$, $\mathbf{J}_N^{(k)}(\boldsymbol{\mu}) = \sum_{i=1}^{m_J} \boldsymbol{\theta}_J^i(\mathbf{V}\mathbf{u}_N^{(k)}(\boldsymbol{\mu}), \boldsymbol{\mu}) \mathbf{J}_N^i$

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: $\mathbf{u}|_{\tilde{\wp}_R}(\boldsymbol{\mu}) = \mathbf{S}_R \mathbf{u}_{N,m}^{(k+1)}(\boldsymbol{\mu})$, $\mathbf{u}|_{\tilde{\wp}_J}(\boldsymbol{\mu}) = \mathbf{S}_J \mathbf{u}_{N,m}^{(k+1)}(\boldsymbol{\mu})$

8: **end while**

5. A posteriori error estimation

We equip our hyper-reduction strategy with an a posteriori estimator for the error $e_{N,m}(\boldsymbol{\mu}) = \mathbf{u}_h(\boldsymbol{\mu}) - \mathbf{V}\mathbf{u}_{N,m}(\boldsymbol{\mu})$ between the high-fidelity solution and the reduced solution. Our estimator accounts for the errors related to both the solution-space reduction and the DEIM and MDEIM approximations of the high-fidelity residual and Jacobian matrix. Let us denote by \mathbf{X}_h the matrix associated to a discrete norm in V_h ; moreover, let us 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

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

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

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

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

Theorem 5.1. *Assume that $\mathbf{J}(\mathbf{V}\mathbf{u}_{N,m}(\boldsymbol{\mu}); \boldsymbol{\mu})$ is locally Lipschitz continuous at $\mathbf{V}\mathbf{u}_{N,m}(\boldsymbol{\mu})$, i.e., there exists $K_h^N(\boldsymbol{\mu}) > 0$ such that*

$$\|\mathbf{J}(\mathbf{V}\mathbf{u}_{N,m}(\boldsymbol{\mu}); \boldsymbol{\mu}) - \mathbf{J}(\mathbf{v}; \boldsymbol{\mu})\|_{\mathbf{X}_h, \mathbf{X}_h^{-1}} \leq K_h^N(\boldsymbol{\mu}) \|\mathbf{V}\mathbf{u}_{N,m}(\boldsymbol{\mu}) - \mathbf{v}\|_{\mathbf{X}_h} \quad \forall \mathbf{v} \in \overline{B}_r(\boldsymbol{\mu})(\mathbf{V}\mathbf{u}_{N,m}(\boldsymbol{\mu})). \quad (5.1)$$

Moreover, denote by

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

and

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

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

$$\begin{aligned} \|\mathbf{u}_h(\boldsymbol{\mu}) - \mathbf{V}\mathbf{u}_{N,m}(\boldsymbol{\mu})\|_{\mathbf{X}_h} &\leq \frac{2}{\beta_m(\boldsymbol{\mu})} \left(\|\mathbf{R}(\mathbf{V}\mathbf{u}_{N,m}(\boldsymbol{\mu}); \boldsymbol{\mu})\|_{\mathbf{X}_h^{-1}} \right. \\ &\quad \left. + \|\mathbf{J}(\mathbf{V}\mathbf{u}_{N,m}(\boldsymbol{\mu}); \boldsymbol{\mu}) - \mathbf{J}_m(\mathbf{V}\mathbf{u}_{N,m}(\boldsymbol{\mu}); \boldsymbol{\mu})\|_{\mathbf{X}_h, \mathbf{X}_h^{-1}}^2 \right). \end{aligned} \quad (5.2)$$

The proof of this theorem is an application of the classical Brezzi-Rappaz-Raviart theory (see, e.g., [8, Theorem 2.1]) and is reported in Appendix 7. Note that this theorem does not require that $\mathbf{V}\mathbf{u}_{N,m}(\boldsymbol{\mu})$ be the limit of the sequence of Newton steps (4.8), neither a solution of problem (4.5). The only requirement is that the Jacobian matrix be Lipschitz continuous in a neighborhood of the element $\mathbf{V}\mathbf{u}_{N,m}(\boldsymbol{\mu})$ about which we are interested in estimating the distance from $\mathbf{u}_h(\boldsymbol{\mu})$. Indeed, this theorem only provides a characterization of a residual-based a posteriori error estimation; note that this latter would be valid for any arbitrary vector \mathbf{w} provided $\mathbf{J}(\mathbf{w}; \boldsymbol{\mu})$ is locally Lipschitz continuous at \mathbf{w} . Furthermore, by assuming that there exists $\tilde{N}(\boldsymbol{\mu})$, such that $\tau_{N,1}(\boldsymbol{\mu}) \leq 1$ and $\tau_{N,2}(\boldsymbol{\mu}) \leq 1$ as soon as $N \geq \tilde{N}(\boldsymbol{\mu})$, we require that a *non-dimensional* measure of the FOM residual and of the MDEIM approximation error of the Jacobian matrix, respectively, are small when evaluated at $\mathbf{V}\mathbf{u}_{N,m}(\boldsymbol{\mu})$ – that is, $\mathbf{V}\mathbf{u}_{N,m}(\boldsymbol{\mu})$ nearly satisfies the FOM. In other words, the a posteriori error bound is rigorous as soon as the RB space and the MDEIM approximation have already been trained.

For the sake of computation, we express the error bound (5.2) under the following, slightly different form, obtained by simply applying the triangular inequality on the first term appearing at the left-hand side of (5.2),

$$\begin{aligned}
\|\mathbf{u}_h(\boldsymbol{\mu}) - \mathbf{V}\mathbf{u}_{N,m}(\boldsymbol{\mu})\|_{\mathbf{X}_h} &\leq \frac{2}{\beta_m(\boldsymbol{\mu})} \left(\|\mathbf{R}_m(\mathbf{V}\mathbf{u}_{N,m}(\boldsymbol{\mu}); \boldsymbol{\mu})\|_{\mathbf{X}_h^{-1}} \right. \\
&\quad + \|\mathbf{R}(\mathbf{V}\mathbf{u}_{N,m}(\boldsymbol{\mu}); \boldsymbol{\mu}) - \mathbf{R}_m(\mathbf{V}\mathbf{u}_{N,m}(\boldsymbol{\mu}); \boldsymbol{\mu})\|_{\mathbf{X}_h^{-1}} \\
&\quad \left. + \|\mathbf{J}(\mathbf{V}\mathbf{u}_{N,m}(\boldsymbol{\mu}); \boldsymbol{\mu}) - \mathbf{J}_m(\mathbf{V}\mathbf{u}_{N,m}(\boldsymbol{\mu}); \boldsymbol{\mu})\|_{\mathbf{X}_h, \mathbf{X}_h^{-1}}^2 \right). \tag{5.3}
\end{aligned}$$

In this way, all the quantities appearing in the error bound (5.3) are easily computable, as they exclusively require the approximations of the residual vector and of the Jacobian matrix based on DEIM and MDEIM respectively.

We remark that the a posteriori error bound (5.3) is not exploited for the sake of basis construction since we entirely rely on the POD technique. However, if a weak greedy algorithm were used to select the snapshots, the right hand side of (5.3) would serve as a criterion to be maximized at each step, see e.g. [35] for an example of application to nonlinear problems. In this paper, it will be solely used in order to certify the numerical results provided by the proposed ROM. Indeed, we rely on POD for the sake of basis construction because of its higher flexibility, also in view of more involved (e.g., coupled electromechanical) problems, and the fact that it does not require the evaluation of error bounds or indicators for the selection of basis functions, as instead a greedy algorithm would do. We remark that the whole reduction and hyper-reduction workflow would still feature the same structure even if a greedy algorithm were used.

6. Numerical results

In order to show the effectiveness of our approach, we present two different applications related to nonlinear mechanical problems: (i) a structural test on a simple geometry and (ii) a cardiac electromechanical model for an idealized left ventricle geometry. Numerical simulations have been performed using the parallel finite element library LifeV (see www.lifev.org).

We start by recalling the formulation our mechanical problems. We consider a reference configuration Ω_0 and an actual configuration Ω . A deformation is a map $\varphi : \Omega_0 \rightarrow \Omega$ from the reference to the actual configuration, such that $\mathbf{x} = \varphi(\mathbf{X})$ for any $\mathbf{X} \in \Omega_0$, $\mathbf{x} \in \Omega$. The deformation gradient tensor $\mathbf{F} : \Omega_0 \rightarrow \Omega$ is defined as

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

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

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

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 $\mathbf{C} : \Omega_0 \rightarrow \Omega$, $\mathbf{C} = \mathbf{F}^T \mathbf{F}$ 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

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

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:

$$\operatorname{div}(\mathbf{P}(\mathbf{u}(\boldsymbol{\mu}); \boldsymbol{\mu})) = \mathbf{f}(\boldsymbol{\mu}) \quad \text{in } \Omega_0,$$

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 $R(\mathbf{u}(\boldsymbol{\mu}); \boldsymbol{\mu}) = \text{div}(\mathbf{P}(\mathbf{u}(\boldsymbol{\mu}); \boldsymbol{\mu})) - \mathbf{f}(\boldsymbol{\mu})$; the high-fidelity solution is then computed relying on the Newton method (2.3).

Although the presented mechanical framework is stationary, it may happen that the mechanical problem actually depends also on time $t \in [0, T_f]$, for instance through time-dependent boundary conditions and forcing terms or through the coupling with time-dependent problems, as we will see in the next sections. In these cases we obtain a quasi-static problem, that is a sequence of stationary problems that are independent of each other, except from the choice of the starting point of the Newton method. Indeed, Newton iterations at t_{n+1} start from the solution computed at t_n . For this kind of problems, time can be considered as an additional parameter, although with peculiar features. Indeed, as we are not interested in solving the problem for generic values of t decided online, we use the same time step Δt in both the offline and online stages; in particular, snapshots are selected only at time instants of the form $t_n = n \Delta t$ where $n = 1, \dots, T_f/\Delta t$.

6.1. Shear test for a cubic domain

We first perform a shear test on a cubic domain $\Omega_0 = [0, 1]^3$. We consider a Saint Venant-Kirchhoff material, which is characterized by the following strain energy function

$$\mathcal{W}(\mathbf{E}) = \frac{\lambda_1}{2} [\text{tr}(\mathbf{E})]^2 + \lambda_2 \text{tr}(\mathbf{E}^2),$$

where λ_1 and λ_2 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 relations

$$\lambda_1 = \frac{\nu E}{(1 + \nu)(1 - 2\nu)} \quad \lambda_2 = \frac{E}{2(1 + \nu)}.$$

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:

$$\mathbf{P} = \lambda_1 \text{tr}(\mathbf{E})\mathbf{F} + 2\lambda_2 \mathbf{F}\mathbf{E},$$

where $\text{tr}(\mathbf{E})$ denotes the trace operator of the tensor \mathbf{E} .

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

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

Although the problem is quasi-static, its solution also depends on time, as so does the Neumann boundary condition. The Dirichlet and the Neumann boundaries are reported in Figure 2 together with the computational mesh. The FOM is built on a conforming mesh with 3072 elements and 729 vertices, resulting in a high-fidelity space V_h of dimension $N_h = 2187$.

We consider the following parameters: time $t \in [0, 10s]$, Young modulus $E \in [6 \cdot 10^4, 7 \cdot 10^4]$, Poisson coefficient $\nu \in [0.3, 0.4]$, external load parameter $\alpha \in [10, 35]$, where α affects the Neumann boundary condition $g(t, \boldsymbol{\mu}) = (\alpha^2 + 5\alpha)(t + 1)$. For the case at hand, we choose a time step $\Delta t = 1s$.

In Figure 3 we report the FOM and the ROM solutions at three time instants, computed for two different values of the parameters; we can observe that the ROM accuracy is comparable with that of the FOM. The reduced solutions have been computed using the MDEIM approach,

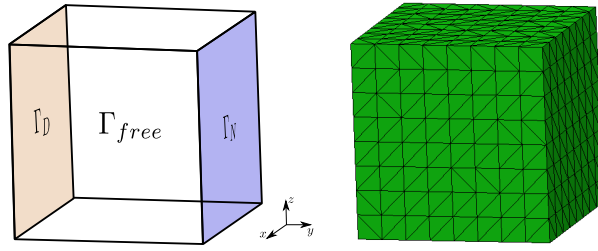


Figure 2: Test case 1: domain and computational grid

with $N = 10$, $m_R = 212$ and $m_J = 38$. The choice $N = 10$ corresponds to a POD tolerance $\varepsilon_{POD} = 10^{-4}$. To compute the POD bases we have chosen $n_s = 300$ snapshots. In Figure 4 we report the difference between the FOM and the ROM solution at the final time instants, for the same values of the parameters chosen above.

In Figure 5 we compare the error $\|\mathbf{u}_h - \mathbf{V}\mathbf{u}_{N,m}\|_{L^2(0,T,H^1(\Omega))}$ as a function of the number of selected basis functions for the classical DEIM and the MDEIM approaches, respectively. The error has been computed over a testing set of 50 randomly chosen parameters over \mathcal{D} . Less than 20 basis functions are needed in order to correctly reproduce the high fidelity solution, for both the

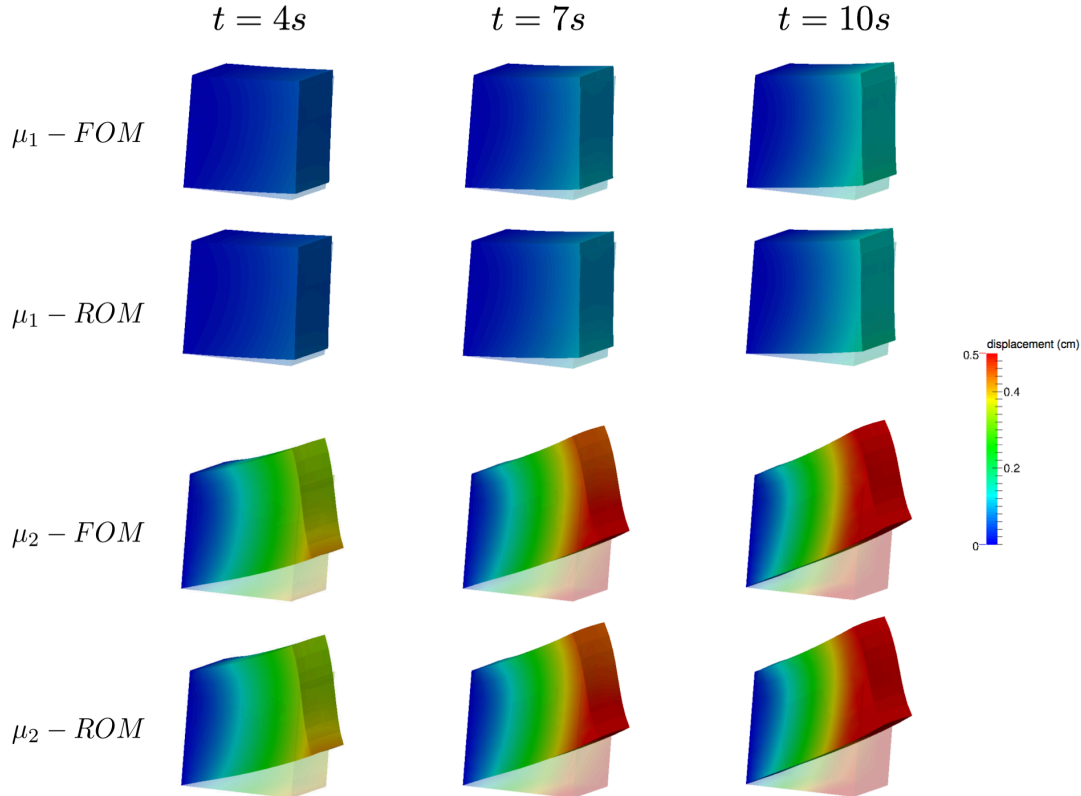


Figure 3: Test case 1, comparison between FOM and ROM solutions. Both solutions are computed at $t = 4s$ (left), $t = 7s$ (center), $t = 10s$ (right) for $\boldsymbol{\mu}_1 = [6.4 \cdot 10^4, 0.31, 10]$ (plots in the upper half part) and $\boldsymbol{\mu}_2 = [6.8 \cdot 10^4, 0.38, 32]$ (plots in the lower half part). FOM solutions are reported on lines 1 and 3, ROM solutions on lines 2 and 4, respectively

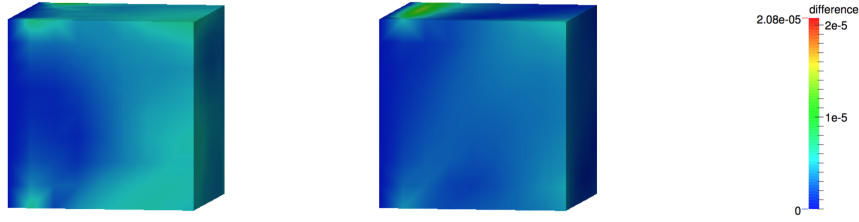


Figure 4: Test case 1, errors between FOM and ROM solutions at $t = 10s$ for $\boldsymbol{\mu}_1 = [6.4 \cdot 10^4, 0.31, 10]$ (left) and $\boldsymbol{\mu}_2 = [6.8 \cdot 10^4, 0.38, 32]$ (right). Errors are reported on the reference, undeformed configuration

considered approaches. The error related to the DEIM technique has been reported for $m_R = 95$ and $m_R = 212$, while the one for the MDEIM technique only for $m_R = 212$. These values of m_R correspond to an imposed tolerance of $\varepsilon_{DEIM} = 10^{-6}$ and $\varepsilon_{DEIM} = 10^{-8}$, respectively, where ε_{DEIM} is the tolerance used in the POD algorithm to construct the DEIM basis Φ_R . In the case of the MDEIM approach, we adopt a lower tolerance because the choice $\varepsilon_{DEIM} = 10^{-6}$ on the DEIM approximation of the residual was not sufficient to guarantee the convergence of the reduced Newton problem for all the parameter combinations considered. Then, we notice that the DEIM approach allows to consider a smaller number of terms m_R for approximating the residual in the online phase. This difference between the two techniques can be explained by considering that the MDEIM approach is a quasi-Newton method, while in the DEIM approach the Jacobian matrix is the exact derivative of the residual. However, the MDEIM technique allows to select a tolerance $\varepsilon_{MDEIM} = 10^{-2}$ which corresponds to $m_J = 38$ terms, while the DEIM approach requires $m_J = 95$ to approximate the Jacobian matrix (that is, the number of terms used to approximate the residual vector). On the other hand, the number of terms required to approximate the residual is higher if MDEIM is used for the Jacobian approximation.

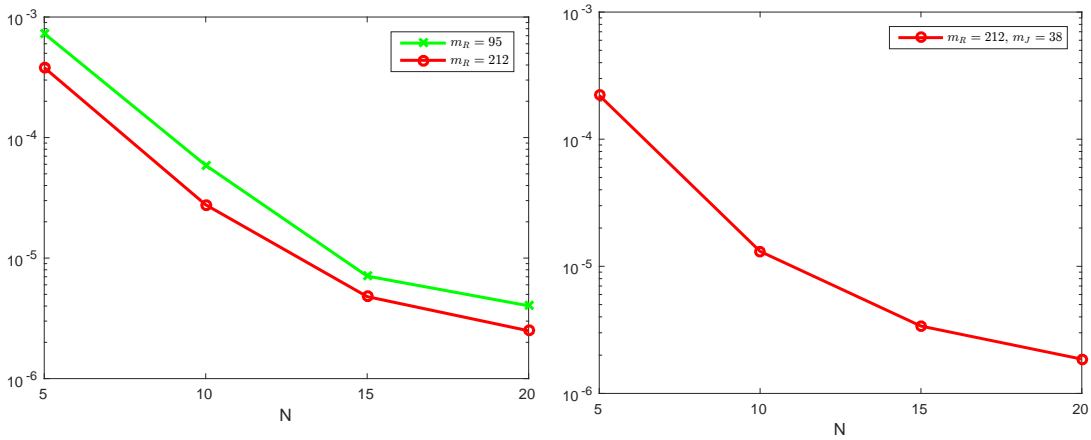


Figure 5: Test case 1, ROM convergence, comparison between DEIM and MDEIM. Average $L^2(0, T_f; H^1(\Omega_0))$ error computed over a testing set of 50 parameters using DEIM (left) and MDEIM (right) (in the MDEIM case choosing $m_R = 95$ does not ensure the convergence of the Newton algorithm)

Number of FE dofs	2187	Relative Newton FE/RB tolerance	10^{-5}
FE time Jacobian assembling	1s	FE system solution time	0.005s
FE time residual assembling	0.18s	FE time for each time step	5s

Table 1: Test case 1, numerical data

Remark 8. Based on our experience, DEIM tolerance ε_{DEIM} is chosen smaller than the one, ε_{POD} imposed to determine the basis dimension N when performing POD; this is motivated by the need to prevent hyper-reduction errors on residuals from dominating over reduction errors. The tolerance $\tilde{\varepsilon}$ required by the stopping criterion in the Newton method on the reduced problem is chosen even smaller than ε_{POD} to ensure the accuracy of the solution of the RB problem. The tolerance ε_{MDEIM} used to stop the selection of terms in the MDEIM approximation of the Jacobian matrix – whenever relying on a MDEIM, instead than DEIM, approximation of the Jacobian – can be chosen much higher than ε_{DEIM} . Indeed, when using MDEIM, inexact Jacobian matrices can be employed without sacrificing the accuracy of the procedure, provided very accurate residuals approximations are used. This is also a compromise between accuracy and efficiency, yielding smaller dimension (in terms of basis functions) of the MDEIM approximation of the Jacobian, and thus a faster assembling.

A detailed comparison of the two techniques is shown in Table 2. We can observe that the computational time required to assemble the residual is noticeably smaller than the time needed for the Jacobian matrix construction. For this reason, even if DEIM allows in principle to consider a smaller number of terms for the residual approximation, MDEIM yields a greater computational speedup, since the MDEIM Jacobian matrix can be assembled on a smaller reduced mesh. In particular, with MDEIM the online reduced problem is solved in about 0.8 s per time step, while DEIM requires 1.2 s for each time step. All the data shown in Table 2 are computed using $N = 10$ and a Newton tolerance of 10^{-5} , as reported in Table 1.

	DEIM	MDEIM
Residual DEIM terms m_R	95	212
Residual Reduced Mesh elements	666	1205
RB time Residual assembling	0.012	0.02s
Jacobian MDEIM terms m_J	-	38
Jacobian reduced Mesh elements	666	463
RB time Jacobian assembling	0.2s	0.13s
RB system solution time	$2 \cdot 10^{-5} s$	$2 \cdot 10^{-5} s$
Mean number of Newton iteration	4	4
Offline time	3h	3h
Online RB time for each time step	1.2s	0.8s
Computational speedup (wrt FOM)	4	7

Table 2: Test case 1, computational data related to the DEIM and the MDEIM approaches. Online RB times are evaluated considering a state space of dimension $N = 10$, obtained by imposing a POD tolerance $\varepsilon_{POD} = 10^{-2}$; the same comparison in the online RB times and computational speedups between DEIM and MDEIM holds for larger RB dimensions $N \leq 20$

In conclusion, we report in Figure 6 the error $\|\mathbf{u}_h(t) - \mathbf{V}\mathbf{u}_{N,m}(t)\|_{\mathbf{X}_h}$ versus the number of selected basis functions for $t = 5s$ and $t = 10s$ and the associated error bound. The matrix \mathbf{X}_h is associated with the norm induced in the discrete space by the continuous $H^1(\Omega_0)$ -norm. The proposed error bound has an effectivity of about $O(10)$ for the largest value of N , thus yielding reliable error estimations. The error behavior in this case may suggest that hyper-reduction errors become comparable (and, later, dominant) showing the *plateau* typical of those cases where the

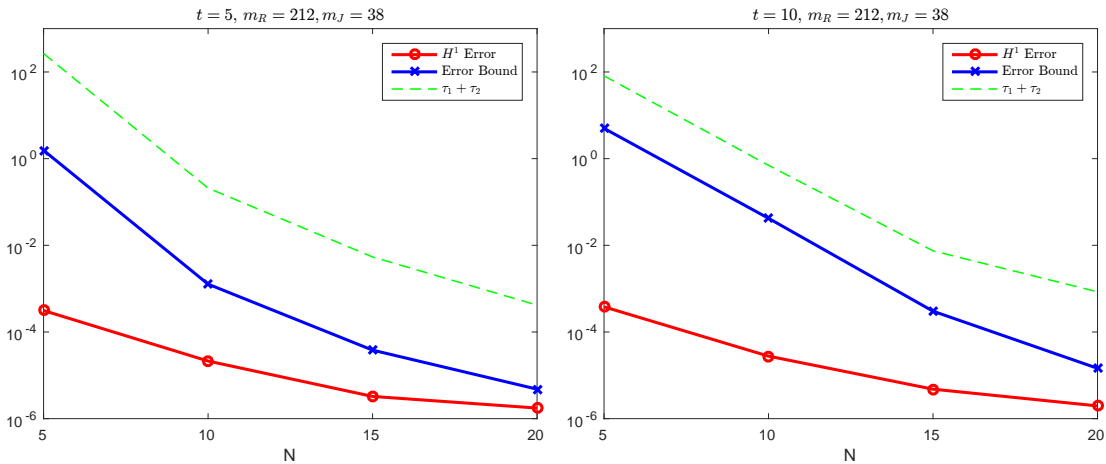


Figure 6: Test case 1, a posteriori error estimation, MDEIM case. Average $H^1(\Omega_0)$ error (red) and a posteriori error bound (blue) computed over a testing set of 50 parameters at $t = 5s$ and $t = 10s$. Also the quantity $\tau_{N,1} + \tau_{N,2}$ is reported (dotted line, in green), since the error bound holds when $\tau_{N,1}$ and $\tau_{N,2}$ are smaller than 1, according to Theorem 5.1

precision in the hyper-reduced solution is limited by the hyper-reduction accuracy⁴. For the case at hand, it is possible to evaluate the Lipschitz constant $K_h^N(\boldsymbol{\mu})$ explicitly (see Appendix B), obtaining

$$K_h^N(\boldsymbol{\mu}) = (2\lambda + 3\mu)(2 + 2C + r(\boldsymbol{\mu})), \quad (6.1)$$

where $C = \sup_{\boldsymbol{\mu} \in \mathcal{D}} \|\mathbf{V}\mathbf{u}_{N,m}(\boldsymbol{\mu})\|_{\mathbf{x}_h}$. For the case at hand, we obtain $K_h^N \leq 10^6$, where C has been estimated numerically as the average value of $\|\mathbf{V}\mathbf{u}_{N,m}(\boldsymbol{\mu})\|_{\mathbf{x}_h}$, computed over a test sample of 100 parameters. We report in Figure 6 also the quantity $\tau_{N,1} + \tau_{N,2}$, as we proved that the error bound holds when $\tau_{N,1}$ and $\tau_{N,2}$ are smaller than 1. For the case at hand, this condition is verified as soon as $N \geq 10$.

6.2. Cardiac electromechanics in an idealized left ventricle

In this section we apply the same approach for the solution of the cardiac electromechanical problem. The cardiac electromechanical problem comprises an electrophysiology problem, which describes the propagation of the electrical signal triggering the heart contraction, and a mechanical problem, 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 numerical solution of the electrical model is considerably faster than that of the mechanical one. We point out that our method allows to consider parameters which are related with both mechanics (e.g. physical properties of the myocardium as the Young modulus) and electrophysiology (e.g. conductivity velocities), which thus affect the mechanics through the solution of the electrical problem. For the sake of simplicity, we consider the case of an idealized left ventricle, although the proposed technique can be applied in principle to more complex or even subject-specific geometries.

⁴Similarly, bounds on hyper reduction errors become smaller and smaller until the desired tolerance is reached; then, we expect that the contribution carried by the norm of the residual keeps on decreasing, while the two additional contributions in the error bound carried by hyper-reduced quantities are limited to the tolerance imposed on the DEIM and MDEIM reconstruction of the residual and the Jacobian, respectively.

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. [44] for a general introduction. Regarding cardiac electrophysiology, a ionic model describing the evolution of ions concentrations and ionic currents in the cell, is coupled 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 [7], coupled with the monodomain model (see e.g.[18] for a complete derivation). To describe cardiac muscle displacements, we assume an orthotropic mechanical constitutive law that accounts for two preferred directions of muscular fibers and sheets, respectively; fibers and sheets are essential to determine the ability of the ventricle to twist and swell correctly during the blood filling phase. For the case at hand, we consider the hyperelastic model proposed in [29], 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 unit vectors in the preferred directions (fibers and sheets, respectively) 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

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

respectively. The coefficients of the Holzapfel-Ogden constitutive law are taken from literature, see e.g. [20]. Fibers and sheets vectors are computed using the algorithm proposed in [48] and exploiting the fact 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 our idealized human ventricle we obtain the fibers distribution shown in Figure 7.

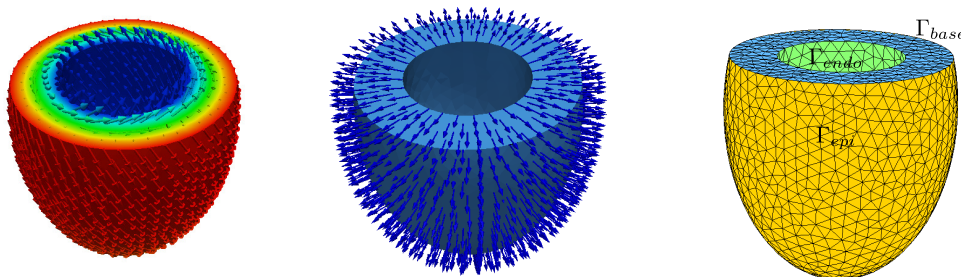


Figure 7: Muscular fibers with $\theta_{max} = 60^\circ$ (left) and sheets (center) and computational grid (right)

In order to properly describe myocardium deformations, we consider a quasi-incompressible formulation [51, 27], which offers several advantages with respect to a full incompressible one, from both a modeling and a numerical viewpoint [43]. Indeed, taking into account very limited volumetric changes is possible according to experimental evidence and the quasi-incompressible formulation leads to a simpler numerical problem.

Moreover, when modeling 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 strain approach [15, 1], which is based on a multiplicative decomposition of the deformation gradient tensor, of the form

$$\mathbf{F}(\mathbf{u}, t) = \mathbf{F}_e(\mathbf{u})\mathbf{F}_a(c(t)),$$

where \mathbf{F}_e describes the elastic deformation of the myocardium while

$$\mathbf{F}_a(c(t)) = I + \gamma_f(c(t))\mathbf{f}_0 \otimes \mathbf{f}_0 + \gamma_s(c(t))\mathbf{s}_0 \otimes \mathbf{s}_0 + \gamma_n(c(t))\mathbf{n}_0 \otimes \mathbf{n}_0$$

is the anelastic deformation due to fibers contraction. Here $c(t)$ is the calcium concentration (obtained from the solution of the ionic model), \mathbf{n}_0 is a unit vector normal to \mathbf{f}_0 and \mathbf{s}_0 , γ_f , γ_s and γ_n are time-dependent coefficients describing the cell shortening respectively in the \mathbf{f}_0 , \mathbf{s}_0 and \mathbf{n}_0 directions. These latter are computed from the following dynamical law

$$\begin{cases} \mu_A \dot{\gamma}_f(t) = (c(t) - c_0)^2 f(\mathcal{I}_{4,f}) + \sum_{j=1}^5 -1^j (j+1)(j+2) \gamma_f^j(t) \\ \gamma_s(t) = \kappa \gamma_f(t), \quad \gamma_n(t) = \frac{1}{(1 + \gamma_f(t))(1 + \gamma_s(t))} - 1, \end{cases}$$

where c_0 is the calcium concentration at the end of the diastolic phase. In this work, we choose $\gamma_f(0) = 0$. Since the electrophysiology problem is time dependent, the solution of our mechanical model depends on time via the solution $c(t)$ of the electrical problem. The Piola tensor takes the following form

$$\mathbf{P}(t) = \det(\mathbf{F}_a(c(t))) \frac{\partial \mathcal{W}(\mathbf{F}_e)}{\partial \mathbf{F}_e} \mathbf{F}_a^{-T}(c(t))$$

so that the full mechanical problem reads as follows

$$\begin{cases} \operatorname{div}(\mathbf{P}(\mathbf{u}(t, \boldsymbol{\mu}); \boldsymbol{\mu}, c(t))) = \mathbf{f} & \text{in } \Omega \\ \mathbf{P}(\mathbf{u}(t, \boldsymbol{\mu}); \boldsymbol{\mu}, c(t)) \mathbf{n} = g & \text{on } \Gamma_{endo} \\ \mathbf{P}(\mathbf{u}(t, \boldsymbol{\mu}); \boldsymbol{\mu}, c(t)) \mathbf{n} + \alpha \mathbf{u}(t, \boldsymbol{\mu}) = 0 & \text{on } \Gamma_{epi} \cup \Gamma_{base}, \end{cases}$$

where we impose homogeneous Neumann boundary conditions on the endocardium Γ_{endo} and Robin boundary conditions on the epicardium Γ_{epi} and on the base Γ_{base} in order to avoid rigid motions. We point out that homogeneous 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. As usual in cardiac mechanics literature (see e.g. [24, 47, 17]), we neglect inertial forces thus obtaining a quasi-static problem. For a detailed discussion on this topic we refer to [44]. We remark that the full Piola tensor depends on the solution of the electrical problem, which acts as a forcing term for the system. For the case at hand, we can consider two different time steps for the electrical and the mechanical problem: indeed, electrophysiology requires a significantly small time step in order to correctly capture the propagation of the signal. Instead, the mechanical displacement is slower than the signal propagation; it is thus sufficient to consider a time step small enough to guarantee the convergence of the Newton method.

6.2.2. Numerical results

We perform a test on an idealized left ventricle geometry, using a computational mesh (see Figure 7) with 8292 elements and 2099 vertices, resulting in an high-fidelity space V_h of dimension $N_h = 6297$. We consider a time interval $t \in [0, 100ms]$, which covers all the systolic phase of the cardiac cycle. We consider a time step Δt_m associated to the mechanical problem which varies through the systole; we choose $\Delta t_m = 5ms$ for $t \in [0, 90)$ and $\Delta t_m = 1ms$ for $t \in [90, 100]$. As for the electrophysiology we use a time step $\Delta t_e = 0.02ms$. In this case we consider as parameters: time $t \in [0, 100ms]$ and the electrical conductivities $\sigma_f \in [15, 40]$ and $\sigma_s = \sigma_n \in [3, 20]$. These parameters have a significant effect on the heart contraction. Indeed, the electrical

conductivities of the myocardium can be noticeably different from one patient to another and they affect the propagation of the electrical signal and, consequently, the correct heart contraction. Other parameters are represented by the fibers orientation, which can have a crucial impact on the correct torsion and shortening of the ventricle; the Bulk modulus $\kappa > 0$, which measures the material resistance to a uniform compression; the isotropic coefficient a , which is related to the stiffness of the cardiac muscle and thus affects the ejected quantity of blood. For the case at hand, since the electrical conductivities affect also the electrophysiology problem, this latter 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. This task does not require, at the moment, to introduce a further ROM since solving the electrical problem by means of a FE approximation for the case at hand requires about 3 minutes for all the systolic phase, on a standard laptop, as reported in Table 3.

Number of FE dofs	6297	Number of RB dofs	12
Offline time	200h	FE electrophysiology solution time	<i>3min</i>
FE time Jacobian assembling	50s	FE time residual assembling	2.5s
FE system solution time	0.08s	FE mechanics time for each time step	<i>6min</i>

Table 3: Test case 2, numerical data

In Figures 8 and 9 we show the displacement of the cardiac muscle at different time instants, obtained for two parameter values with the FOM and the ROM, respectively. The ROM simulations have been performed using $N = 12$ basis functions, obtained by imposing a POD tolerance $\varepsilon_{POD} = 10^{-2}$, $m_R = 294$ and $m_J = 21$. The POD bases are computed from a set of $n_s = 400$ snapshots. We observe that the proposed ROM accurately captures the solution obtained with an high-fidelity model, for the whole systolic phase of the cardiac cycle. From these figures, we can also detect how the solution is affected by parameters; specifically, when increasing the conductivities the shortening of the ventricle increases, together with the wall thickening. In Figure 10, we report instead the displacement field obtained with the MDEIM approach for three different parameter values on the central section of the ventricle. We can observe that large values of the conductivities are associated with a greater reduction of the internal volume of the ventricle; thus highlighting the significant impact of the electrical conductivities on the quantity of blood ejected.

Moreover, we evaluate the average relative error $\|\mathbf{u}_h - \mathbf{V}\mathbf{u}_{N,m}\|_{L^2(0,T;H^1(\Omega_0))}$ using the classical DEIM method and the MDEIM method (Figure 11). The error decreases when increasing the number of basis functions, and it becomes smaller than 10^{-2} when considering at least 8 basis functions. For this test case DEIM converges when $\varepsilon_{DEIM} \leq 10^{-5}$, while MDEIM requires $\varepsilon_{DEIM} = 10^{-7}$. These values correspond to $m_R = 108$ and $m_R = 294$, respectively. However, the MDEIM technique allows to approximate the Jacobian matrix choosing $\varepsilon_{MDEIM} = 10^{-2}$ which corresponds to $m_J = 21$. Moreover, we notice that the error obtained with DEIM increases when considering $m_R = 108$, thus leading to a less accurate ROM solution. Data related to the computational performances of the two approaches are reported in Table 4. For this test case the online RB time needed for a serial computation of each time step of the electromechanical solution is about 2 minutes on a standard laptop using the DEIM approach, even if considering $m_R = 108$, and reduces to 40 seconds in the MDEIM case. Hence, even if MDEIM relies on a quasi-Newton method and thus requires more iterations than the Newton method employed by DEIM, assembling the Jacobian on the MDEIM reduced mesh takes about the 10% of the CPU time required by the same operation performed on the DEIM reduced mesh. This explains the overall efficiency of the MDEIM approach. The FOM simulation requires instead about 6 minutes,

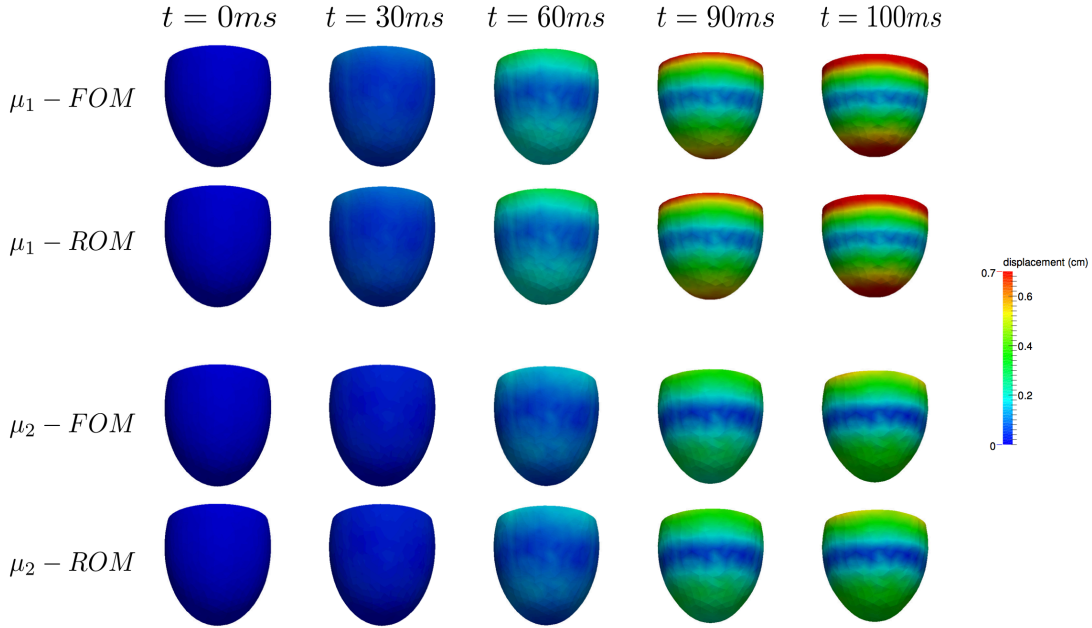


Figure 8: Test case 2, comparison between FOM and ROM solutions (displacements). Contraction of the ventricle is computed at different time instants for $\mu_1 = [39, 19]$ (plots in the upper half part) and $\mu_2 = [16, 4]$ (plots in the lower half part). FOM solutions are reported on lines 1 and 3, ROM solutions on lines 2 and 4, respectively

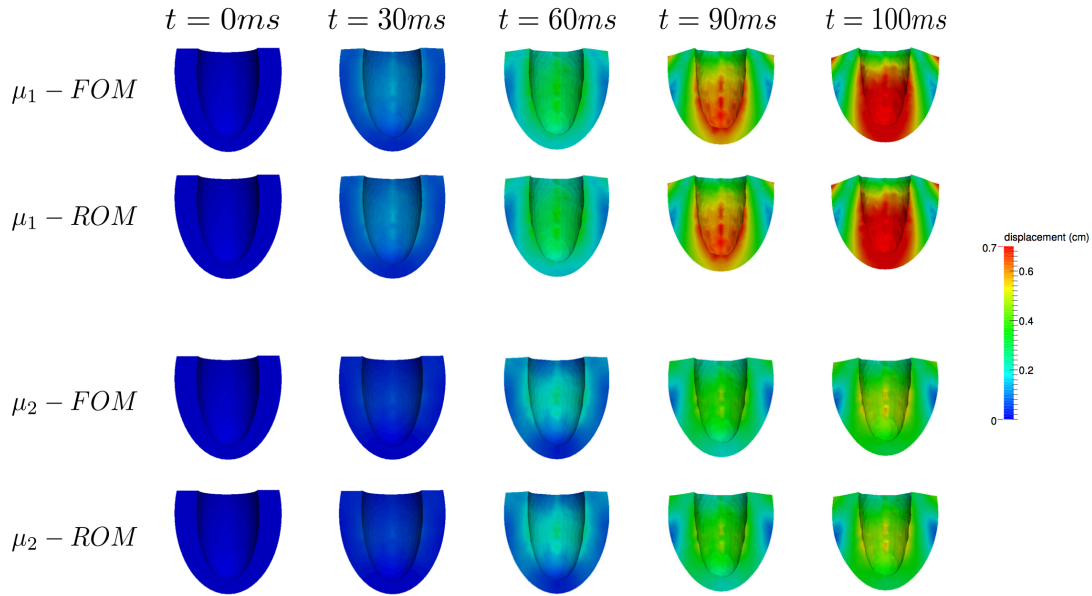


Figure 9: Test case 2, comparison between FOM and ROM solutions (displacements). A section of the ventricle is visualized at different time instants for $\mu_1 = [39, 19]$ (plots in the upper half part) and $\mu_2 = [16, 4]$ (plots in the lower half part). FOM solutions are reported on lines 1 and 3, ROM solutions on lines 2 and 4, respectively

running in parallel on 16 cores of an HPC machine. This computational speedup can be explained by considering the reduced mesh over with we assemble the Jacobian matrices; indeed the amount of elements of the DEIM reduced mesh is significantly larger than the one selected by the MDEIM procedure, see Figure 12. For the case at hand, we can conclude that MDEIM has to be preferred to DEIM in terms of both accuracy and efficiency.

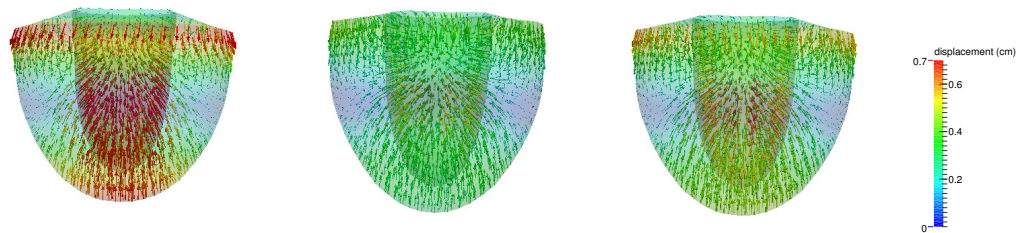


Figure 10: Test case 2, solution variability with respect to parameter values. Displacement fields obtained with the MDEIM approach are visualized on a section of the ventricle at $t = 100ms$ for $\mu_1 = [37, 14]$ (left), $\mu_2 = [15, 3]$ (middle), $\mu_3 = [20, 20]$ (right)

Finally, we also show the computed errors and corresponding a posteriori error bounds, computed at $t = 30$, $t = 60$, $t = 90$ and $t = 100$ over a testing set of 50 parameters (see Figure 13); in this case the effectivity is between 10 and 10^2 , slightly larger than in the test case of Sect. 6.1.

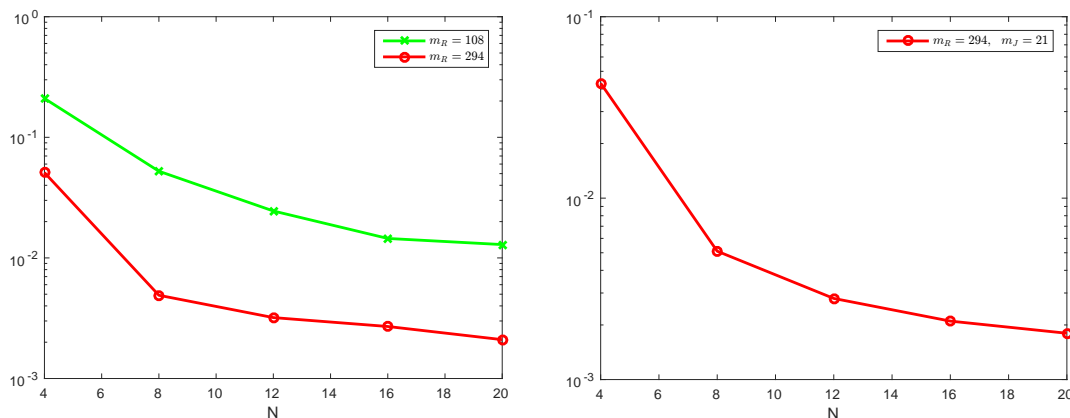


Figure 11: Average H^1 relative error computed over a testing set of 50 parameters using DEIM (left) and MDEIM (right) (in the MDEIM case choosing $m_R = 108$ does not ensure the convergence of the Newton algorithm)

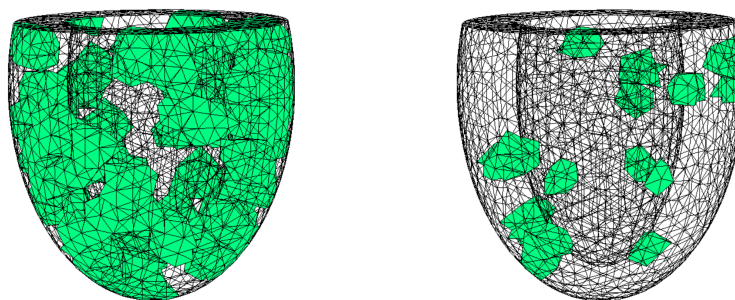


Figure 12: Test case 2. Reduced meshes for the approximation of the Jacobian matrix obtained using DEIM (left) and MDEIM (right). MDEIM allows to select a significantly smaller much number of elements compared to DEIM, thus making the assembling of the Jacobian matrix much more efficient

	DEIM	MDEIM
Residual DEIM terms m_R	108	294
Residual Reduced Mesh elements	2328	4827
RB time Residual assembling	0.5s	1s
Jacobian MDEIM terms m_J	-	21
Jacobian reduced Mesh elements	2328	329
RB time Jacobian assembling	22s	3s
RB system solution time	$2 \cdot 10^{-5} s$	$2 \cdot 10^{-5} s$
Mean number of Newton iteration	7	13
Offline time	200h	200h
Online RB time for each time step	2min	40s
Computational speedup (wrt FOM)	3	9

Table 4: Test case 2, computational data related to the DEIM and the MDEIM approaches. Online RB times are evaluated considering a state space of dimension $N = 12$, obtained by imposing a POD tolerance $\varepsilon_{POD} = 10^{-2}$; the same comparison in the online RB times and computational speedups between DEIM and MDEIM holds for larger dimensions $N \leq 20$

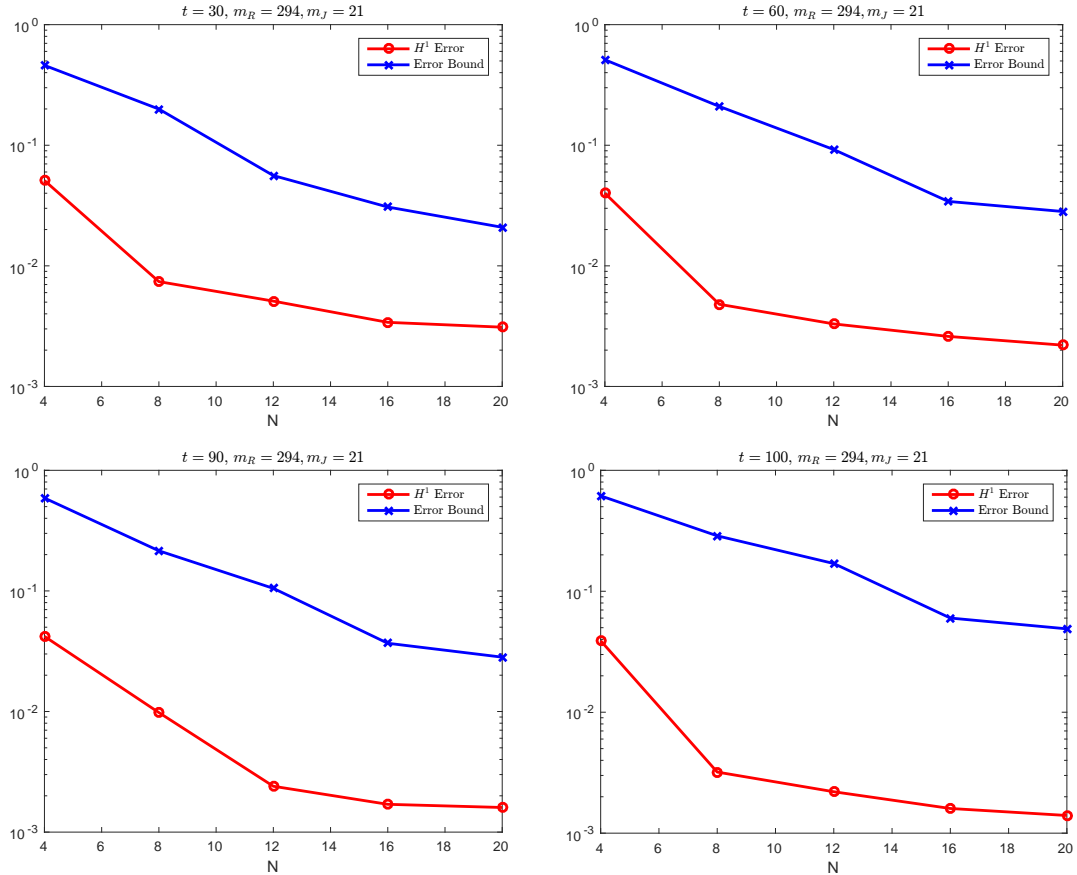


Figure 13: Test case 2, a posteriori error estimation. Average $H^1(\Omega_0)$ error (red) and a posteriori error bound (blue) computed over a testing set of 50 parameters at $t = 30, 60, 90, 100$ ms

7. Conclusions

In this paper we have proposed a reduction strategy for highly nonlinear parametrized mechanical problems, which combines proper orthogonal decomposition for the selection of basis functions, Galerkin projection over a low dimensional subspace, DEIM for the efficient assembling of the residual vectors and a matrix discrete empirical interpolation (MDEIM) technique for efficiently handling the Jacobian matrices. At our knowledge, this is the first time that the MDEIM technique is used to reduce three dimensional highly nonlinear parametrized problem. The resulting ROM allows to evaluate the problem solution at a very reduced computational cost, but entailing the same accuracy of the high fidelity model. We have proposed a new snapshots selection strategy able to provide accurate and fast online solutions, still retaining low offline computational times. A detailed comparison with the classical DEIM approach has been carried out, in order to highlight advantages and disadvantages of the proposed method.

We have proposed a new reliable a posteriori error bound, that takes 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 our approach has been proven on two different test cases. In the former, a shear test on a cubic domain for a Saint Venant-Kirchhoff material characterized by a polynomial nonlinearity has been considered. In the latter, our method has been applied to the cardiac electromechanical problem on an idealized left ventricle geometry. This application demonstrates that our reduction strategy can be efficiently used also on problems with highly nonlinear constitutive laws and parametric dependence induced by the coupling with other problems.

Appendix A. Proof of theorem 5.1

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 $\overline{B}_r(\mathbf{V}\mathbf{u}_{N,m})$, let us define the map $\mathbf{H} : \mathbb{R}^{N_h} \rightarrow \mathbb{R}^{N_h}$

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

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

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

Using the mean value theorem, we obtain

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

so that

$$\begin{aligned} \mathbf{H}(\mathbf{v}) - \mathbf{V}\mathbf{u}_{N,m} &= \mathbf{J}_m(\mathbf{V}\mathbf{u}_{N,m})^{-1} \left[(\mathbf{J}_m(\mathbf{V}\mathbf{u}_{N,m}) - \mathbf{J}(\mathbf{V}\mathbf{u}_{N,m}))(\mathbf{v} - \mathbf{V}\mathbf{u}_{N,m}) \right. \\ &\quad \left. + \int_0^1 (\mathbf{J}(\mathbf{V}\mathbf{u}_{N,m}) - \mathbf{J}(\mathbf{V}\mathbf{u}_{N,m} + s(\mathbf{v} - \mathbf{V}\mathbf{u}_{N,m})))(\mathbf{v} - \mathbf{V}\mathbf{u}_{N,m})ds - \mathbf{R}(\mathbf{V}\mathbf{u}_{N,m}) \right]. \end{aligned}$$

Then,

$$\begin{aligned} \|\mathbf{H}(\mathbf{v}) - \mathbf{V}\mathbf{u}_{N,m}\|_{\mathbf{x}_h} &\leq \frac{1}{\beta_m^N} \left(\|\mathbf{v} - \mathbf{V}\mathbf{u}_{N,m}\|_{\mathbf{x}_h} \|\mathbf{J}(\mathbf{V}\mathbf{u}_{N,m}) - \mathbf{J}_m(\mathbf{V}\mathbf{u}_{N,m})\|_{\mathbf{x}_h, \mathbf{x}_h^{-1}} \right. \\ &\quad + \|\mathbf{v} - \mathbf{V}\mathbf{u}_{N,m}\|_{\mathbf{x}_h} \int_0^1 \|\mathbf{J}(\mathbf{V}\mathbf{u}_{N,m}) - \mathbf{J}(\mathbf{V}\mathbf{u}_{N,m} + s(\mathbf{v} - \mathbf{V}\mathbf{u}_{N,m}))\|_{\mathbf{x}_h, \mathbf{x}_h^{-1}} ds \\ &\quad \left. + \|\mathbf{R}(\mathbf{V}\mathbf{u}_{N,m})\|_{\mathbf{x}_h^{-1}} \right). \end{aligned}$$

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

$$\begin{aligned} \|\mathbf{H}(\mathbf{v}) - \mathbf{V}\mathbf{u}_{N,m}\|_{\mathbf{x}_h} &\leq \frac{1}{\beta_m^N} \left(\frac{1}{2K_h^N} \|\mathbf{J}(\mathbf{V}\mathbf{u}_{N,m}) - \mathbf{J}_m(\mathbf{V}\mathbf{u}_{N,m})\|_{\mathbf{x}_h, \mathbf{x}_h^{-1}}^2 \right. \\ &\quad \left. + \frac{3K_h^N}{2} \|\mathbf{v} - \mathbf{V}\mathbf{u}_{N,m}\|_{\mathbf{x}_h}^2 + \|\mathbf{R}(\mathbf{V}\mathbf{u}_{N,m})\|_{\mathbf{x}_h^{-1}} \right). \end{aligned}$$

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

$$\begin{aligned} \|\mathbf{H}(\mathbf{v}) - \mathbf{V}\mathbf{u}_{N,m}\|_{\mathbf{x}_h} &\leq \frac{1}{\beta_m^N} \left[\frac{1}{2K_h^N} \|\mathbf{J}(\mathbf{V}\mathbf{u}_{N,m}) - \mathbf{J}_m(\mathbf{V}\mathbf{u}_{N,m})\|_{\mathbf{x}_h, \mathbf{x}_h^{-1}}^2 \right. \\ &\quad + \frac{3K_h^N}{2} \left(\frac{4}{(\beta_m^N)^2} \|\mathbf{J}(\mathbf{V}\mathbf{u}_{N,m}) - \mathbf{J}_m(\mathbf{V}\mathbf{u}_{N,m})\|_{\mathbf{x}_h, \mathbf{x}_h^{-1}}^4 \right. \\ &\quad + \frac{4}{(\beta_m^N)^2} \|\mathbf{R}(\mathbf{V}\mathbf{u}_{N,m})\|_{\mathbf{x}_h^{-1}} \|\mathbf{J}(\mathbf{V}\mathbf{u}_{N,m}) - \mathbf{J}_m(\mathbf{V}\mathbf{u}_{N,m})\|_{\mathbf{x}_h, \mathbf{x}_h^{-1}}^2 \\ &\quad \left. \left. + \frac{4}{(\beta_m^N)^2} \|\mathbf{R}(\mathbf{V}\mathbf{u}_{N,m})\|_{\mathbf{x}_h^{-1}}^2 \right) + \|\mathbf{R}(\mathbf{V}\mathbf{u}_{N,m})\|_{\mathbf{x}_h^{-1}} \right], \end{aligned}$$

so that

$$\|\mathbf{H}(\mathbf{v}) - \mathbf{V}\mathbf{u}_{N,m}\|_{\mathbf{x}_h} \leq \frac{1}{\beta_m^N} \left(2\|\mathbf{J}(\mathbf{V}\mathbf{u}_{N,m}) - \mathbf{J}_m(\mathbf{V}\mathbf{u}_{N,m})\|_{\mathbf{x}_h, \mathbf{x}_h^{-1}}^2 + 2\|\mathbf{R}(\mathbf{V}\mathbf{u}_{N,m})\|_{\mathbf{x}_h^{-1}} \right) = r. \quad (7.1)$$

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

$$\begin{aligned} \mathbf{H}(\mathbf{v}_1) - \mathbf{H}(\mathbf{v}_2) &= \mathbf{v}_1 - \mathbf{J}_m(\mathbf{V}\mathbf{u}_{N,m})^{-1} \mathbf{R}(\mathbf{v}_1) - \mathbf{v}_2 + \mathbf{J}_m(\mathbf{V}\mathbf{u}_{N,m})^{-1} \mathbf{R}(\mathbf{v}_2) \\ &= \mathbf{J}_m(\mathbf{V}\mathbf{u}_{N,m})^{-1} \left[(\mathbf{v}_1 - \mathbf{v}_2) (\mathbf{J}_m(\mathbf{V}\mathbf{u}_{N,m}) - \mathbf{J}(\mathbf{V}\mathbf{u}_{N,m})) \right. \\ &\quad \left. + \int_0^1 \mathbf{J}(\mathbf{V}\mathbf{u}_{N,m}) - \mathbf{J}(\mathbf{v}_2 + s(\mathbf{v}_1 - \mathbf{v}_2)) (\mathbf{v}_1 - \mathbf{v}_2) ds \right]. \end{aligned}$$

Thus, we obtain

$$\begin{aligned} \|\mathbf{H}(\mathbf{v}_1) - \mathbf{H}(\mathbf{v}_2)\|_{\mathbf{x}_h} &\leq \frac{1}{\beta_m^N} \left(\|\mathbf{v}_1 - \mathbf{v}_2\|_{\mathbf{x}_h} \|\mathbf{J}(\mathbf{V}\mathbf{u}_{N,m}) - \mathbf{J}_m(\mathbf{V}\mathbf{u}_{N,m})\|_{\mathbf{x}_h, \mathbf{x}_h^{-1}} + K_h^N \|\mathbf{v}_1 - \mathbf{v}_2\|_{\mathbf{x}_h}^2 \right) \\ &\leq \frac{1}{\beta_m^N} \left(\|\mathbf{J}(\mathbf{V}\mathbf{u}_{N,m}) - \mathbf{J}_m(\mathbf{V}\mathbf{u}_{N,m})\|_{\mathbf{x}_h, \mathbf{x}_h^{-1}} + K_h^N R \right) \|\mathbf{v}_1 - \mathbf{v}_2\|_{\mathbf{x}_h} \\ &\leq \left(\frac{1}{\beta_m^N} \|\mathbf{J}(\mathbf{V}\mathbf{u}_{N,m}) - \mathbf{J}_m(\mathbf{V}\mathbf{u}_{N,m})\|_{\mathbf{x}_h, \mathbf{x}_h^{-1}} + \frac{2K_h^N}{(\beta_m^N)^2} \|\mathbf{J}(\mathbf{V}\mathbf{u}_{N,m}) - \mathbf{J}_m(\mathbf{V}\mathbf{u}_{N,m})\|_{\mathbf{x}_h, \mathbf{x}_h^{-1}}^2 \right. \\ &\quad \left. + \frac{2K_h^N}{(\beta_m^N)^2} \|\mathbf{R}(\mathbf{V}\mathbf{u}_{N,m})\|_{\mathbf{x}_h^{-1}} \right) \|\mathbf{v}_1 - \mathbf{v}_2\|_{\mathbf{x}_h}. \end{aligned}$$

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

$$\begin{aligned} \|\mathbf{H}(\mathbf{v}_1) - \mathbf{H}(\mathbf{v}_2)\|_{\mathbf{x}_h} &\leq \left(\frac{1}{\sqrt{12K_h^N}} + \frac{1}{6} + \frac{1}{3} \right) \|\mathbf{v}_1 - \mathbf{v}_2\|_{\mathbf{x}_h} \\ &\leq \left(\frac{1}{\sqrt{12}} + \frac{1}{2} \right) \|\mathbf{v}_1 - \mathbf{v}_2\|_{\mathbf{x}_h} < \|\mathbf{v}_1 - \mathbf{v}_2\|_{\mathbf{x}_h}. \end{aligned}$$

Thanks to the Banach fixed-point theorem (see e.g. [16]), there exists a unique fixed point $\mathbf{u}_h \in \overline{B}_r(\mathbf{V}\mathbf{u}_{N,m})$ of \mathbf{H} , i.e. $\mathbf{H}(\mathbf{u}_h) = \mathbf{u}_h$, hence $\mathbf{R}(\mathbf{u}_h) = 0$. In conclusion, using (7.1) we have

$$\begin{aligned} \|\mathbf{u}_h - \mathbf{V}\mathbf{u}_{N,m}\|_{\mathbf{x}_h} &\leq \frac{2}{\beta_m(\boldsymbol{\mu})} \left(\|\mathbf{R}(\mathbf{V}\mathbf{u}_{N,m})\|_{\mathbf{x}_h^{-1}} + \|\mathbf{J}(\mathbf{V}\mathbf{u}_{N,m}) - \mathbf{J}_m(\mathbf{V}\mathbf{u}_{N,m})\|_{\mathbf{x}_h, \mathbf{x}_h^{-1}}^2 \right) \\ &\leq \frac{2}{\beta_m(\boldsymbol{\mu})} \left(\|\mathbf{R}_m(\mathbf{V}\mathbf{u}_{N,m})\|_{\mathbf{x}_h^{-1}} + \|\mathbf{R}(\mathbf{V}\mathbf{u}_{N,m}) - \mathbf{R}_m(\mathbf{V}\mathbf{u}_{N,m})\|_{\mathbf{x}_h^{-1}} \right. \\ &\quad \left. + \|\mathbf{J}(\mathbf{V}\mathbf{u}_{N,m}) - \mathbf{J}_m(\mathbf{V}\mathbf{u}_{N,m})\|_{\mathbf{x}_h, \mathbf{x}_h^{-1}}^2 \right). \end{aligned}$$

Appendix B. Lipschitz constant derivation for test case 1

In this section, we detail the derivation of the Lipschitz constant (6.1). We recall that the Piola tensor for a Saint Venant-Kirchoff material can be written under the following form

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

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

$$\begin{aligned} \langle J_{\mathbf{u}}(\mathbf{w}), \mathbf{z} \rangle &= \lambda \int_{\Omega_0} \left[(\mathbf{F} : \nabla \mathbf{w})(\mathbf{F} : \nabla \mathbf{z}) + \frac{1}{2}(\mathcal{I}_1 - 3)(\nabla \mathbf{w} : \nabla \mathbf{z}) \right] d\Omega_0 \\ &\quad + \mu \int_{\Omega_0} \left[(\nabla \mathbf{w} \mathbf{C} : \nabla \mathbf{z}) + (\mathbf{F}\mathbf{F}^T \nabla \mathbf{w} : \nabla \mathbf{z}) - (\nabla \mathbf{w} : \nabla \mathbf{z}) + (\mathbf{F}\nabla \mathbf{w}^T \mathbf{F} : \nabla \mathbf{z}) \right] d\Omega_0. \end{aligned}$$

We want to show that $\mathbf{J}(\mathbf{V}\mathbf{u}_{N,m}(\boldsymbol{\mu}); \boldsymbol{\mu})$ 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 $\mathbf{v} \in \overline{B}_r(\boldsymbol{\mu})(\mathbf{V}\mathbf{u}_{N,m}(\boldsymbol{\mu}))$

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

Upon defining

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

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 we omit the $\boldsymbol{\mu}$ dependence hereon.

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

$$\begin{aligned} &|\mathbf{F}^T(\mathbf{V}\mathbf{u}_{N,m})\mathbf{F}(\mathbf{V}\mathbf{u}_{N,m}) - \mathbf{F}^T(\mathbf{v})\mathbf{F}(\mathbf{v})| \\ &= |\mathbf{F}^T(\mathbf{V}\mathbf{u}_{N,m})(\mathbf{F}(\mathbf{V}\mathbf{u}_{N,m}) - \mathbf{F}(\mathbf{v})) + (\mathbf{F}^T(\mathbf{V}\mathbf{u}_{N,m}) - \mathbf{F}^T(\mathbf{v}))\mathbf{F}(\mathbf{v})| \\ &\leq (2 + 2C + r) \|\mathbf{V}\mathbf{u}_{N,m}(\boldsymbol{\mu}) - \mathbf{v}\|_{\mathbf{x}_h}. \end{aligned}$$

Then, it holds that

$$\begin{aligned}
& |\langle J_{\mathbf{V}\mathbf{u}_{N,m}}(\mathbf{w}), \mathbf{z} \rangle - \langle J_{\mathbf{v}}(\mathbf{w}), \mathbf{z} \rangle| \\
&= \lambda \int_{\Omega_0} (\mathbf{F}(\mathbf{V}\mathbf{u}_{N,m}) : \nabla \mathbf{w})(\mathbf{F}(\mathbf{V}\mathbf{u}_{N,m}) : \nabla \mathbf{z}) - (\mathbf{F}(\mathbf{v}) : \nabla \mathbf{w})(\mathbf{F}(\mathbf{v}) : \nabla \mathbf{z}) d\Omega_0 \\
&\quad + \frac{1}{2} \lambda \int_{\Omega_0} (\mathcal{I}_1(\mathbf{V}\mathbf{u}_{N,m}) - \mathcal{I}_1(\mathbf{v}))(\nabla \mathbf{w} : \nabla \mathbf{z}) d\Omega_0 \\
&\quad + \mu \int_{\Omega_0} (\nabla \mathbf{w}(\mathbf{F}^T(\mathbf{V}\mathbf{u}_{N,m})\mathbf{F}(\mathbf{V}\mathbf{u}_{N,m}) - \mathbf{F}^T(\mathbf{v})\mathbf{F}(\mathbf{v})) : \nabla \mathbf{z}) d\Omega_0 \\
&\quad + \mu \int_{\Omega_0} (\nabla \mathbf{w}(\mathbf{F}(\mathbf{V}\mathbf{u}_{N,m})\mathbf{F}^T(\mathbf{V}\mathbf{u}_{N,m}) - \mathbf{F}(\mathbf{v})\mathbf{F}^T(\mathbf{v})) : \nabla \mathbf{z}) d\Omega_0 \\
&\quad + \mu \int_{\Omega_0} ((\mathbf{F}(\mathbf{V}\mathbf{u}_{N,m})\nabla \mathbf{w}\mathbf{F}^T(\mathbf{V}\mathbf{u}_{N,m}) - \mathbf{F}(\mathbf{v})\nabla \mathbf{w}\mathbf{F}^T(\mathbf{v})) : \nabla \mathbf{z}) d\Omega_0 \\
&\leq (2\lambda + 3\mu)(2 + 2C + r(\mu)) \|\mathbf{V}\mathbf{u}_{N,m} - \mathbf{v}\|_{\mathbf{x}_h} \|\mathbf{w}\|_{\mathbf{x}_h} \|\mathbf{z}\|_{\mathbf{x}_h}.
\end{aligned}$$

Exploiting the definition of the $\|\cdot\|_{\mathbf{x}_h, \mathbf{x}_h^{-1}}$ norm, (6.1) automatically follows.

References

- [1] D. Ambrosi, G. Arioli, F. Nobile, and A. Quarteroni. Electromechanical Coupling in Cardiac Dynamics: The Active Strain Approach. *SIAM J. Appl. Math.*, 71(2):605–621, 2011.
- [2] D. Amsallem, J. Cortial, K. Carlberg, and C. Farhat. A method for interpolating on manifolds structural dynamics reduced-order models. *Int. J. Numer. Meth. Engng*, 80(9):1241–1258, 2009.
- [3] 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, (MS&A) series*, pages 101–136. Springer, Switzerland, 2014.
- [4] 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.
- [5] M. Barrault, Y. Maday, N.C. Nguyen, and A.T. Patera. An empirical interpolation method: application to efficient reduced-basis discretization of partial differential equations. *C. R. Math.*, 339(9):667–672, 2004.
- [6] M.M. Baumann. *Nonlinear Model Order Reduction using POD/DEIM for Optimal Control of Burgers’ Equation*. PhD thesis, TU Delft, Delft University of Technology, 2013.
- [7] A. Bueno-Orovio, E.M. Cherry, and F.H. Fenton. Minimal model for human ventricular action potentials in tissue. *J. Theor. Biol.*, 253(3):544–560, 2008.
- [8] G. Caloz and J. Rappaz. Numerical analysis for nonlinear and bifurcation problems. In P. G. Ciarlet and J.L. Lions, editors, *Handbook of Numerical Analysis*, volume V, pages 487–637. North-Holland, Amsterdam, 1997.
- [9] K. Carlberg, C. Bou-Mosleh, and C. Farhat. Efficient non-linear model reduction via a least-squares Petrov-Galerkin projection and compressive tensor approximations. *Int. J. Numer. Meth. Engng*, 86:155–181, 2011.
- [10] 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. Comput. Phys.*, 242:623–647, 2013.

- [11] 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.
- [12] 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.
- [13] S. Chaturantabut and D.C. Sorensen. Nonlinear Model Reduction via Discrete Empirical Interpolation. *SIAM J. Sci. Comput.*, 32(5):2737–2764, 2010.
- [14] S. Chaturantabut and D.C. Sorensen. Application of pod and deim on dimension reduction of nonlinear miscible viscous fingering in porous media. *Math. Comp. Model. Dyn.*, 17(4):337–353, 2011.
- [15] C. Cherubini, S. Filippi, P. Nardinocchi, and L. Teresi. An electromechanical model of cardiac tissue: constitutive issues and electrophysiological effects. *Progr. Biophys. Mol. Bio.*, 97(2-3):562–73, 2008.
- [16] P.G. Ciarlet. *Linear and nonlinear functional analysis with applications*, volume 130. SIAM, 2013.
- [17] P. Colli Franzone, L. Pavarino, and S. Scacchi. Parallel multilevel solvers for the cardiac electro-mechanical coupling. *Appl. Numer. Math.*, 95:140–153, 2015.
- [18] 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.
- [19] M. Drohmann, B. Haasdonk, and M. Ohlberger. Reduced basis approximation for nonlinear parametrized evolution equations based on empirical operator interpolation. *SIAM J. Sci. Comput.*, 34(2):A937–A969, 2012.
- [20] T.S.E. Eriksson, A.J. Prassel, G. Plank, and G.A. Holzapfel. Influence of myocardial fiber/sheet orientations on left ventricular mechanical contraction. *Math. Mech. Solids*, 2013.
- [21] R. Everson and L. Sirovich. Karhunen-Loeve procedure for gappy data. *J. Opt. Soc. Am. A*, 12(8):1657, 1995.
- [22] C. Farhat, P. Avery, T. Chapman, and J. Cortial. Dimensional reduction of nonlinear finite element dynamic models with finite rotations and energy-based mesh sampling and weighting for computational efficiency. *Int. J. Numer. Methods Engng.*, 98(9):625–662, 2014.
- [23] C. Farhat, T. Chapman, and P. Avery. Structure-preserving, stability, and accuracy properties of the energy-conserving sampling and weighting method for the hyper reduction of nonlinear finite element dynamic models. *Int. J. Numer. Methods Engng.*, 102(5):1077–1110, 2015.
- [24] S. Göktepe and E. Kuhl. Electromechanics of the heart: a unified approach to the strongly coupled excitation–contraction problem. *Comput. Mech.*, 45(2-3):227–243, 2010.
- [25] G.H. Golub and C.F. Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.
- [26] 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.
- [27] J. Helfenstein, M. Jabareen, E. Mazza, and S. Govindjee. On non-physical response in models for fiber-reinforced hyperelastic materials. *Int. J. Solids Struct.*, 47(16):2056–2061, 2010.
- [28] J.S. Hesthaven, G. Rozza, and B. Stamm. Certified reduced basis methods for parametrized partial differential equations. *SpringerBriefs in Mathematics*, 2016.
- [29] G. Holzapfel and R.W. Ogden. Constitutive modelling of passive myocardium: a structurally based framework for material characterization. *Philos. T. Roy. Soc. A*, 367(1902):3445–3475, 2009.

- [30] D.B.P. Hyunh and A.T. Patera. Reduced-basis approximation and a posteriori error estimation for stress intensity factors. *Arch. Comput. Meth. Eng.*, 72(10):341–368, 1219–1259.
- [31] 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. *Comput. Meth. Appl. Mech. Eng.*, 200:850–866, 2011.
- [32] P. Krysl, S. Lall, and J.E. Marsden. Dimensional model reduction in non-linear finite element dynamics of solids and structures. *Int. J. Numer. Meth. Eng.*, 51(4):479–504, 2001.
- [33] S. Lall, P. Krysl, and J.E. Marsden. Structure-preserving model reduction for mechanical systems. *Physica D: Nonlinear Phenomena*, 184(1):304–318, 2003.
- [34] 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.
- [35] 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.
- [36] 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.
- [37] 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.
- [38] F. Negri. *Efficient reduction techniques for the simulation and optimization of parametrized systems: analysis and applications*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne, 2015.
- [39] F. Negri, A. Manzoni, and D. Amsallem. Efficient model reduction of parametrized systems by matrix discrete empirical interpolation. *J. Comput. Phys.*, 303:431–454, 2015.
- [40] 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.
- [41] R.P. Pawlowski, J.N. Shadid, J.P. Simonis, and H.F. Walker. Globalization techniques for newton-krylov methods and applications to the fully coupled solution of the navier-stokes equations. *SIAM review*, 48(4):700–721, 2006.
- [42] B. Peherstorfer, D. Butnaru, K. Willcox, and H. Bungartz. Localized discrete empirical interpolation method. *SIAM J. Sci. Comput.*, 36(1):A168–A192, 2014.
- [43] S. Pezzuto. *Mechanics of the Heart - Constitutive Issues and Numerical Experiments*. PhD thesis, Politecnico di Milano, 2013.
- [44] A. Quarteroni, T. Lassila, S. Rossi, and R. Ruiz-Baier. Integrated heart - coupling multiscale and multiphysics models for the simulation of the cardiac function. *Comput. Method Appl. Mech. Eng.*, 2016, in press.
- [45] A. Quarteroni, A. Manzoni, and F. Negri. *Reduced Basis Methods for Partial Differential Equations. An Introduction*, volume 92 of *Unitext Series*. Springer, 2016.
- [46] A. Radermacher and S. Reese. POD-based model reduction with empirical interpolation applied to nonlinear elasticity. *Int. J. Numer. Meth. Engrg.*, 107(6):477–495, 2015.
- [47] S. Rossi. *Anisotropic Modeling of Cardiac Mechanical Activation*. PhD thesis, Ecole Polytechnique Federale de Lausanne, 2014.

- [48] 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. *Eur. J. Mech. A-Solid*, 48, 2013.
- [49] D. Ryckelynck. A priori hyperreduction method: an adaptive approach. *J. Comput. Phys.*, 202(1):346–366, 2005.
- [50] D. Ryckelynck, F. Chinesta, E. Cueto, and A. Ammar. On the a priori model reduction: Overview and recent developments. *Arch. Comput. Meth. Eng.*, 13(1):91–128, 2006.
- [51] C. Sansour. On the physical assumptions underlying the volumetric-isochoric split and the case of anisotropy. *Eur. J. Mech. A-Solid*, 27(1):28–39, 2008.
- [52] 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, 2016.
- [53] 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.
- [54] P. Tiso and D.J. Rixen. Discrete empirical interpolation method for finite element structural dynamics. In *Topics in Nonlinear Dynamics, Volume 1*, pages 203–212. Springer, 2013.
- [55] 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.
- [56] 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.
- [57] J. Yvonnet and Q.C. He. The reduced model multiscale method (R3M) for the non-linear homogenization of hyperelastic media at finite strains. *J. Comput. Phys.*, 223:341–368, 2007.
- [58] 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.
- [59] E. Zeidler, A.B. Nemeth, and C.I. Gheorghiu. Nonlinear functional analysis and its applications. *Acta Applicandae Mathematicae*, 36(3):304–305, 1994.