



Research article

Projection-based reduced order models for parameterized nonlinear time-dependent problems arising in cardiac mechanics

Ludovica Cicci¹, Stefania Fresca¹, Stefano Pagani¹, Andrea Manzoni^{1,*} and Alfio Quarteroni^{1,2}

¹ MOX - Department of Mathematics, Politecnico di Milano, P.zza Leonardo da Vinci 32, 20133 Milano, Italy

² Institute of Mathematics, École Polytechnique Fédérale de Lausanne, Switzerland

* **Correspondence:** Email: andrea.l.manzoni@polimi.it; Tel: +390223994638;
Fax: +390223994568.

Abstract: The numerical simulation of several virtual scenarios arising in cardiac mechanics poses a computational challenge that can be alleviated if traditional full-order models (FOMs) are replaced by reduced order models (ROMs). For example, in the case of problems involving a vector of input parameters related, e.g., to material coefficients, projection-based ROMs provide mathematically rigorous physics-driven surrogate ROMs. In this work we demonstrate how, once trained, ROMs yield extremely accurate predictions (according to a prescribed tolerance) – yet cheaper than the ones provided by FOMs – of the structural deformation of the left ventricular tissue over an entire heartbeat, and of related output quantities of interest, such as the pressure-volume loop, for any desired input parameter values within a prescribed parameter range. However, the construction of ROM approximations for time-dependent cardiac mechanics is not straightforward, because of the highly nonlinear and multiscale nature of the problem, and almost never addressed. Our approach relies on the reduced basis method for parameterized partial differential equations. This technique performs a Galerkin projection onto a low-dimensional space for the displacement variable; the reduced space is built from a set of solution snapshots – obtained for different input parameter values and time instances – of the high-fidelity FOM, through the proper orthogonal decomposition technique. Then, suitable hyper-reduction techniques, such as the Discrete Empirical Interpolation Method, are exploited to efficiently handle nonlinear and parameter-dependent terms. In this work we show how a fast and reliable approximation of the time-dependent cardiac mechanical model can be achieved by a projection-based ROM, taking into account both passive and active mechanics for the left ventricle providing all the building blocks of the methodology, and highlighting those challenging aspects that are still open.

Keywords: projection-based reduced order models; proper orthogonal decomposition; discrete

1. Introduction

The heart is a hollow, muscular organ that receives blood from the veins and pumps it out through the arteries, back into the circulatory system. Its activity between two consecutive heartbeats is referred to as the cardiac cycle, which consists of a sequence of alternating contraction (*systole*) and relaxation (*diastole*) of the cardiac chambers. The active left ventricular systole is the main agent of blood ejection in the circulatory system and results, at the macroscopic level, in a longitudinal shortening, a wall thickening and a torsion around the longitudinal axis. This latter is due to the particular fiber distribution of the cardiac muscle cells, the cardiomyocytes, which highly influences the mechanical response of the heart. Although different physical phenomena occur in the cardiac function - namely, the propagation of the electrical potential, ion dynamics, active contraction of cardiomyocytes, tissue mechanics and blood circulation, see, e.g., [1, 2] - in this article we focus on the mechanical activity of the cardiac tissue. In particular, we take into account both the passive response of the tissue, and the active shortening of the muscular fibers, by adopting the finite elasticity models to describe the heart contraction and relaxation.

Describing the mechanics of the cardiac tissue requires complex constitutive laws, characterized by an exponential strain energy function and the presence of muscular fibers and sheets. For this reason, the passive myocardium is modeled as an hyperelastic orthotropic material, ultimately yielding highly nonlinear mechanical problems to be solved. The numerical approximation of these problems through high-fidelity, full order models (FOMs) such as those built on the finite element method entails severe computational costs, because of the need to account for complex geometrical configurations, and suitable spatial and temporal discretizations. Computational complexity is even more exacerbated if one is interested in going beyond a single, direct simulation. Indeed, when simulating cardiac mechanics, several input data – ultimately depending on a set of parameters – affect the problem under investigation, often varying within a broad range, and possibly hampered by uncertainty.

Addressing the impact of input parameters variation on mechanical outputs of clinical interest thus plays a key role in order to explore the tissue response in many different conditions [3, 4], to calibrate the numerical solver and, ultimately, to obtain personalized models taking into account inter-patient variability [5]. From a numerical perspective, those tasks yield the solution of parameter estimation, sensitivity analysis, and uncertainty quantification problems; all these tasks share the need of multiple queries of the parameter-to-solution map, in accurate (and possibly very fast) way. However, the construction of a surrogate model for the input-output map, which directly approximates the quantity of interest as a function of the input (possibly, random) parameters, is often preferred to a reduced order approximation of the field variables. For instance, the effect of uncertainty of both global myocardial material properties and the local variability of the microstructure orientation on the left ventricular function has been considered in [6] using meta-models built through a polynomial chaos expansion; similarly, in [4] both sensitivity analysis and forward UQ methods have been applied to investigate the left ventricular function during the full cardiac cycle also involving a circulatory model, to assess the effect of regional wall thickness, fiber orientation, passive material parameters, active stress and the

circulatory model, again using polynomial chaos expansion surrogate models. However, despite being more intrusive to implement, ROMs obtained through a projection process on the equations governing the FOM yield more accurate approximations than data fits and usually generate more significant computational gains than lower-fidelity models, requiring less training data [7].

In this paper we adapt ROMs developed in the last decade [8, 9] for parameterized PDEs to the case of time-dependent nonlinear problems arising in cardiac mechanics. We consider projection-based ROMs built through the reduced basis method, suitably equipped with hyper-reduction strategies to assemble nonlinear terms efficiently. Exploiting proper orthogonal decomposition (POD) for the construction of the reduced spaces, we can rely on completely physics-based ROMs, rather than on data-driven strategies such as the ones recently proposed and exploiting, e.g., a fully-automatic deep learning (DL) workflow to generate both volumetric parameters and strain measures from cine-MRI data [10], Gaussian process interpolations [11–13], or physics-informed neural networks [14]. Indeed, despite their nonlinear nature, parameterized problems in elastodynamics such as those arising from cardiac modeling do not pose serious issues regarding the dimensionality of the reduced spaces, the slow decay of singular values when considering POD, or the assumption stated by the reduced basis (RB) method entailing a linear superimposition of modes.

The construction of projection-based reduced order models addressing nonlinear unsteady problems related with cardiac mechanics has been considered in very few contributions so far. A quasi-static model was employed in [15] to describe the contraction of an idealized left ventricle geometry with few degrees of freedom, where a POD-Galerkin ROM was used at each time step, also exploiting further hyper-reduction techniques such as the discrete empirical interpolation (DEIM) and its matrix version (MDEIM) to efficiently assemble the residual vectors and the Jacobian matrices, respectively, required by Newton iterations at the ROM level. In this case, the activation of the heart contraction was given by the solution of a high-fidelity electrophysiology model, whilst the pressure caused by the presence of blood in the ventricular chamber was neglected. An extension to quasi-Newton methods can be found in [16]. A POD-Galerkin ROM for a patient-specific biventricular cardiac model has been introduced and analyzed in [17], for the sake of parameter estimation based on medical images. More recently, the inverse analysis of large nonlinear cardiac simulations has been considered in [18], where POD-Galerkin ROMs have been built for the contraction of a four-chambers cardiac model built over a patient-specific cardiac geometry, involving a monolithic coupling between a POD-reduced three-dimensional structural model – however treating the passive myocardium as isotropic – and a zero-dimensional Windkessel model for the blood circulation. In this latter case, a speedup ranging from 5 to 13 times compared to the FOM is achieved by the ROM, when dealing with the evaluation of cardiac outputs as functions of the contractility, the myofiber activation/deactivation rate, the onset of ventricular systole/diastole. No hyper-reduction strategies are considered, in this latter examples, to enhance the construction of the ROM.

In this paper we build POD-Galerkin-DEIM reduced order models, relying on *(i)* POD for finding a low-dimensional trial subspace, *(ii)* Galerkin projection to generate, in a physically consistent way, a reduced order model, and *(iii)* hyper-reduction techniques, through the discrete empirical interpolation method (DEIM), to accelerate the assembling of nonlinear (with respect to the solution, input parameters, or both) terms. Such hyper-reduced ROMs are thus exploited for the efficient and accurate solution to the time-dependent cardiac mechanics problem, on idealized and patient-specific left ventricle geometries, albeit using a relative low number of degrees of freedom for practical reasons.

Our mechanical problem is uncoupled from both cardiac electrophysiology and blood dynamics. However, we employ suitable analytical time-dependent functions to mimic the influence of both the active stress and the blood circulation, to assess the performance of the proposed methodology when the whole cardiac cycle is taken into account. Treating the passive myocardium as transversely orthotropic makes the problem much more challenging compared to the case where the material is treated as isotropic, concerning the efficiency of the hyper-reduction stage. Indeed, if the construction of a reduced subspace to approximate the problem solution does not pose serious issues, resulting in extremely low dimensional spaces even for complex material laws, the bottleneck is represented by the assembling of reduced operators, and the projection of the approximated operators through DEIM.

The structure of this paper is as follows. After a brief recall (Section 2) on the mathematical modeling of cardiac mechanics, we describe the high-fidelity finite element (FE) approximation we start from (Section 3). We then introduce the key tools of the proposed ROM technique (Section 4): the POD-Galerkin method and suitable hyper-reduction strategies, to tackle nonlinear time-dependent problems. Numerical results dealing with a benchmark prolate spheroid geometry – concerning the passive inflation without and with the active contraction – and the full cardiac cycle of a patient-specific left ventricle geometry are then shown (Section 5). Finally, open critical issues and future perspectives are outlined (Section 6).

2. Mathematical models for cardiac mechanics

After introducing a basic setting for solid mechanics, in this section we provide the formulation of the mathematical problems we are interested in. Given a continuum body \mathcal{B} embedded in a three-dimensional Euclidean space, let us denote by $\Omega_0 \subset \mathbb{R}^3$ its reference configuration at time $t = 0$ and by $\Omega_t \subset \mathbb{R}^3$ its current configuration at time $t > 0$. The motion of the body $\chi : \Omega_0 \times \mathbb{R}^+ \rightarrow \mathbb{R}^3$ is defined as a function which takes a generic material point $\mathbf{X} \in \Omega_0$ and maps it onto the corresponding spatial point $\mathbf{x} = \chi(\mathbf{X}, t) \in \Omega$, for all times $t > 0$. The displacement field is defined as

$$\mathbf{u}(\mathbf{X}, t) = \mathbf{x}(\mathbf{X}, t) - \mathbf{X},$$

for all times $t \geq 0$, and represents the unknown of the problem we are interested in. A central quantity in finite deformations theory is the so-called deformation gradient,

$$\mathbf{F}(\mathbf{X}, t) = \nabla_0 \chi(\mathbf{X}, t) = \frac{\partial \mathbf{x}(\mathbf{X}, t)}{\partial \mathbf{X}}, \quad \mathbf{F}_{ij} = \frac{\partial x_i(\mathbf{X}, t)}{\partial X_j},$$

which describes the relationship between quantities in the undeformed and the deformed configurations of the body, and can be expressed in terms of the displacement field as

$$\mathbf{F}(\mathbf{X}, t) = \mathbf{I} + \nabla_0 \mathbf{u}(\mathbf{X}, t),$$

where \mathbf{I} is the identity matrix and ∇_0 denotes the material gradient. The change in volume between the reference and the current configurations at time $t > 0$ is given by the determinant of the deformation tensor,

$$J(\mathbf{X}, t) = \det \mathbf{F}(\mathbf{X}, t),$$

also known as volume ratio. If there is no motion, i.e., $\mathbf{x} = \mathbf{X}$, we obtain the consistency condition $J = 1$. In general, a motion for which this condition holds is said to be *isochoric*.

Another measure of deformation is the right Cauchy-Green deformation tensor $\mathbf{C} = \mathbf{F}^T \mathbf{F}$, whilst a common measure of the strain is the Green-Lagrange strain tensor

$$\mathbf{E} = \frac{1}{2}(\mathbf{C} - \mathbf{I}),$$

which is defined in the reference configuration and is zero in absence of motion. All these kinematic quantities are commonly used to express constitutive equations describing the relationship between mechanical forces and the material motion. We will restrict ourselves to the case of hyperelastic materials, for which we can assume that there exists a strain energy density function \mathcal{W} such that

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

gives the functional stress-strain relationships, where \mathbf{P} is the first Piola-Kirchhoff stress tensor.

2.1. Passive and active mechanics

We assume the myocardium to be hyperelastic, orthotropic, with nonlinear passive behavior, as experiments have shown higher material stiffness and mechanical response along the cardiac fibers. Many constitutive laws have been derived for both its passive and active description – see, e.g., [19–24] – taking into account varying material symmetries. In this work, we adopt the relation proposed in [19] and commonly referred to as the *Guccione* law, which assumes the material to be transversely isotropic, with its primary material axis oriented along the local fiber direction. The corresponding strain-energy density function is given by

$$\mathcal{W}(\mathbf{F}) = \frac{C}{2}(e^{Q(\mathbf{F})} - 1), \quad (2.1)$$

where the three-dimensional transverse isotropy with respect to the fiber coordinate system is accounted by choosing

$$Q(\mathbf{F}) = b_f E_{ff}^2 + b_s E_{ss}^2 + b_n E_{nn}^2 + b_{fs}(E_{fs}^2 + E_{sf}^2) + b_{fn}(E_{fn}^2 + E_{nf}^2) + b_{sn}(E_{sn}^2 + E_{ns}^2).$$

Here, E_{ij} , $i, j \in \{f, s, n\}$ are the components of the Green-Lagrange strain tensor \mathbf{E} , the material constant C scales the stresses and the coefficients b_f , b_s , b_n are related to the material stiffness in the fiber, sheet and cross-fiber directions, respectively.

Since biological tissues are mostly composed of water, the material density ρ_0 is often taken constant in time [23] so that the conservation of mass implies $J = 1$ and an incompressible formulation is adopted. However, as often done in cardiac mechanics [25, 26], incompressibility is weakly imposed, and an isochoric-volumetric decoupling of the strain energy function is employed, yielding

$$\mathcal{W}(\mathbf{F}) = \frac{C}{2}(e^{Q(\mathbf{F})} - 1) + \mathcal{W}_{vol}(J).$$

The volumetric term $\mathcal{W}_{vol}(J)$ is chosen to penalize large volume variations, so that it is usually a convex function with global minimum in $J = 1$, e.g.,

$$\mathcal{W}_{vol}(J) = \frac{K}{2}(J - 1) \ln(J).$$

Another crucial aspect is the inclusion of the active contractile forces in the constitutive equation. Active properties are time-dependent and anisotropic, with more active stress generated along the local muscle fiber direction [27]. Active tension can be integrated into the passive stress tensor in different ways, among which the active stress and the active strain represent the most common approaches [28]. Here, we adopt the former, and add to the passive first Piola-Kirchhoff stress tensor a time-dependent active tension $T_a(t; \boldsymbol{\mu})$, which is assumed to act only in the fiber direction,

$$\mathbf{P}(\mathbf{F}) = \frac{\partial}{\partial \mathbf{F}} \left(\frac{C}{2}(e^\varrho - 1) + \frac{K}{2}(J - 1)\log(J) \right) + T_a(t; \boldsymbol{\mu})(\mathbf{F}\mathbf{f}_0 \otimes \mathbf{f}_0), \quad (2.2)$$

where $\mathbf{f}_0 \in \mathbb{R}^3$ denotes the reference unit vector in the fiber direction. Since \mathbf{F} can be written in terms of the displacement field \mathbf{u} , in the following we will write $\mathbf{P} = \mathbf{P}(\mathbf{u})$.

2.2. Parameterized problems in cardiac mechanics

We have now all the ingredients required to formulate the mathematical model to describe the motion of a body, specifically the cardiac left ventricle, in terms of an initial boundary-value problem (IBVP) for the elastodynamics equation, that is, the equation of motion for continuum mechanics.

To account for inter-patient variability, different scenarios can be described by assuming the dependence of the mechanical displacement on a set of input parameters, collectively denoted as $\boldsymbol{\mu} \in \mathcal{P} \subset \mathbb{R}^P$, where \mathcal{P} is a suitable compact set. For the application at hand, possible parameters of interest are the coefficients of the constitutive law, such as those related to the stiffness of the cardiac muscle in different direction or the bulk modulus K related to the material incompressibility; fibers orientation is highly patient-specific and have a crucial impact on the torsion and shortening of the ventricle; moreover, the pressure exerting on the endocardium or the active tension greatly affects the mechanical behavior; see Section 5.

The parameterized IBVPs of interest in cardiac mechanics can be stated in general form as follows: given $\boldsymbol{\mu} \in \mathcal{P}$, a body force field $\mathbf{b}_0 = \mathbf{b}_0(\mathbf{X}, t; \boldsymbol{\mu})$, a prescribed displacement $\bar{\mathbf{u}} = \bar{\mathbf{u}}(\mathbf{X}, t; \boldsymbol{\mu})$ and surface traction $\bar{\mathbf{T}} = \bar{\mathbf{T}}(\mathbf{X}, t, \mathbf{N}; \boldsymbol{\mu})$, find the displacement field $\mathbf{u}(\boldsymbol{\mu}): \Omega_0 \times [0, T) \rightarrow \mathbb{R}^3$ such that

$$\begin{cases} \rho_0 \partial_t^2 \mathbf{u}(\mathbf{X}, t; \boldsymbol{\mu}) - \nabla_0 \cdot \mathbf{P}(\mathbf{u}(\mathbf{X}, t; \boldsymbol{\mu})) = \mathbf{b}_0(\mathbf{X}, t; \boldsymbol{\mu}) & \text{in } \Omega_0 \times (0, T) \\ \mathbf{u}(\mathbf{X}, t; \boldsymbol{\mu}) = \bar{\mathbf{u}}(\mathbf{X}, t; \boldsymbol{\mu}) & \text{on } \Gamma_0^D \times (0, T) \\ \mathbf{P}(\mathbf{u}(\mathbf{X}, t; \boldsymbol{\mu}))\mathbf{N} = \bar{\mathbf{T}}(\mathbf{X}, t, \mathbf{N}; \boldsymbol{\mu}) & \text{on } \Gamma_0^N \times (0, T) \\ \mathbf{P}(\mathbf{u}(\mathbf{X}, t; \boldsymbol{\mu}))\mathbf{N} + \alpha \mathbf{u}(\mathbf{X}, t; \boldsymbol{\mu}) + \beta \partial_t \mathbf{u}(\mathbf{X}, t; \boldsymbol{\mu}) = \mathbf{0} & \text{on } \Gamma_0^R \times (0, T) \\ \mathbf{u}(\mathbf{X}, 0; \boldsymbol{\mu}) = \mathbf{u}_0(\mathbf{X}; \boldsymbol{\mu}), \quad \partial_t \mathbf{u}(\mathbf{X}, 0; \boldsymbol{\mu}) = \dot{\mathbf{u}}_0(\mathbf{X}; \boldsymbol{\mu}) & \text{in } \Omega_0 \times \{0\}, \end{cases} \quad (2.3)$$

where $\alpha, \beta \in \mathbb{R}$, $\Gamma_0^D \cup \Gamma_0^N \cup \Gamma_0^R = \partial\Omega_0$ and $\Gamma_0^i \cap \Gamma_0^j = \emptyset$ for $i, j \in \{D, N, R\}$. These equations are inherently nonlinear and an additional source of (exponential) nonlinearity is introduced in the material law (2.2). Hence, suitable discretization techniques are required for the solution of the IBVP.

3. High-fidelity approximation

Before addressing the construction of ROMs for the mechanical problem (2.3), we introduce its full-order approximation, upon which the ROMs will be built. In this work we rely on the finite element method (FEM) for the space discretization, and on backward differentiation formula (BDF) schemes

for time discretization. In particular, we neglect the external body forces \mathbf{b}_0 , assume homogeneous Dirichlet boundary conditions, i.e., $\bar{\mathbf{u}} = \mathbf{0}$, and set $\alpha = \beta = 0$, thus obtaining homogeneous Neumann boundary conditions on Γ_0^R . Moreover, the boundaries Γ_0^D , Γ_0^N and Γ_0^R represent the base (representing the artificial boundary resulting from truncation of the heart below the valves in a short axis plane), the endocardium and the epicardium, respectively.

First of all, let us introduce the weak formulation for (2.3) in material description [29]. By setting

$$V(\Omega_0) := [H_{\Gamma_0^D}^1(\Omega_0)]^3 = \left\{ \boldsymbol{\eta} \in [H^1(\Omega_0)]^3 : \boldsymbol{\eta} = \mathbf{0} \text{ on } \Gamma_0^D \right\},$$

multiplying the equation of motion (2.3)₁ by a test function $\boldsymbol{\eta} \in V(\Omega_0)$ and integrating over Ω_0 , we obtain the weak formulation of (2.3) as follows

$$\int_{\Omega_0} \rho_0 \partial_t^2 \mathbf{u}(t; \boldsymbol{\mu}) \cdot \boldsymbol{\eta} d\Omega + \int_{\Omega_0} \mathbf{P}(\mathbf{u}(t; \boldsymbol{\mu})) : \nabla \boldsymbol{\eta} d\Omega - \int_{\Gamma_N} \bar{\mathbf{T}}(t, \mathbf{N}; \boldsymbol{\mu}) \cdot \boldsymbol{\eta} d\Gamma = 0 \quad \forall \boldsymbol{\eta} \in V(\Omega_0). \quad (3.1)$$

From (3.1), we can set the finite element (FE) discretization of the problem. Let us denote by \mathcal{T}_h a hexahedral mesh of Ω_0 such that $\bigcup_{\tau \in \mathcal{T}_h} \tau$ is an approximation of the domain Ω_0 , where $h > 0$ denotes the grid size, and define the FE space of dimension $r \geq 1$

$$\mathcal{X}_h^r = \left\{ v \in C^0(\bar{\Omega}_0) : v|_{\tau} \in \mathbb{Q}_r(\tau) \quad \forall \tau \in \mathcal{T}_h \right\},$$

where $\mathbb{Q}_r(\tau)$ is the set of polynomials of degree smaller than or equal to r on each element τ and $\dim(\mathcal{X}_h^r) = N_{h,r}^{dof}$ denotes the total number of degrees of freedom (dofs). The FE space of vector-valued functions is defined as

$$V_h = V(\Omega_0) \cap [\mathcal{X}_h^r]^3 = \text{span} \{ \boldsymbol{\varphi}_i \}_{i=1}^{N_h}$$

whose dimension $N_h = 3N_{h,r}^{dof}$ corresponds to the total number of structural dofs, including those associated with the Dirichlet boundary conditions, and $\boldsymbol{\varphi}_i : \Omega_0 \rightarrow \mathbb{R}^3$, for $i = 1, \dots, N_h$, are the vector-valued basis functions. Given $\mathbf{w}(t) \in V(\Omega_0)$, its approximation $\tilde{\mathbf{w}}_h(t) \approx \mathbf{w}(t)$ in V_h can be expressed as a linear combination of the basis functions $\{ \boldsymbol{\varphi}_i \}_{i=1}^{N_h}$ as

$$\tilde{\mathbf{w}}_h(t) = \sum_{i=1}^{N_h} \mathbf{w}_{h,i}(t) \boldsymbol{\varphi}_i,$$

where $\mathbf{w}_h(t) = [\mathbf{w}_{h,i}(t)]_{i=1}^{N_h} \in \mathbb{R}^{N_h}$ denotes the corresponding vector of coefficients in the expansion with respect to the FE basis, that is, the unknown of our high-fidelity FOM.

We can now introduce the semi-discrete Galerkin-FE approximation of the problem, which takes the form of a second-order dynamical system: for each $t \in (0, T)$, find $\mathbf{u}_h(t; \boldsymbol{\mu}) \in V(\Omega_0)$ such that

$$\begin{cases} \rho_0 \mathcal{M} \partial_t^2 \mathbf{u}_h(t; \boldsymbol{\mu}) + \mathcal{S}(\mathbf{u}_h(t)) = \mathcal{F}^{ext}(t) \\ \mathbf{u}_h(0) = \mathbf{u}_{h,0}, \quad \partial_t \mathbf{u}_h(0) = \dot{\mathbf{u}}_{h,0} \end{cases} \quad (3.2)$$

where $\mathbf{u}_{h,0} = [(\mathbf{u}_0, \boldsymbol{\varphi}_i)_{[L^2(\Omega_0)]^3}]_{i=1}^{N_h}$, $\partial_t \mathbf{u}_{h,0} = [(\dot{\mathbf{u}}_0, \boldsymbol{\varphi}_i)_{[L^2(\Omega_0)]^3}]_{i=1}^{N_h}$ and

$$\begin{aligned} [\mathcal{M}]_{ij} &= \int_{\Omega_0} \boldsymbol{\varphi}_j \cdot \boldsymbol{\varphi}_i d\Omega, \quad \forall i, j = 1, \dots, N_h, \\ [\mathcal{S}(\mathbf{u}_h(t; \boldsymbol{\mu}))]_i &= \int_{\Omega_0} \mathbf{P}(\mathbf{u}_h(t; \boldsymbol{\mu})) : \nabla \boldsymbol{\varphi}_i d\Omega, \quad \forall i = 1, \dots, N_h, \\ [\mathcal{F}^{ext}(t; \boldsymbol{\mu})]_i &= \int_{\Gamma_0^N} \bar{\mathbf{T}}(t, \mathbf{N}; \boldsymbol{\mu}) \cdot \boldsymbol{\varphi}_i d\Gamma, \quad \forall i = 1, \dots, N_h. \end{aligned}$$

For the time discretization of the dynamical system (3.2), we consider a uniform partition $\{t^n = n\Delta t, n = 0, \dots, N_t\}$ of the interval $(0, T)$, where $\Delta t = \frac{T}{N_t}$ is the time step length. To approximate the first and the second derivative at time t^n , we employ the following BDF schemes of order 1

$$\dot{\mathbf{u}}_h(t^n) \approx \frac{\mathbf{u}_h^n - \mathbf{u}_h^{n-1}}{\Delta t}, \quad \ddot{\mathbf{u}}_h(t^n) \approx \frac{\mathbf{u}_h^n - 2\mathbf{u}_h^{n-1} + \mathbf{u}_h^{n-2}}{\Delta t^2},$$

respectively, where the superscripts n , $n-1$ and $n-2$ denote the solution \mathbf{u}_h computed at time t^n , t^{n-1} and t^{n-2} , e.g., $\mathbf{u}_h(t^n) = \mathbf{u}_h^n$. From now on, we indicate all the quantities computed at time t^n with the superscript n , for $n = 0, \dots, N_t$. Note that we employ implicit time integration schemes in order to avoid restrictions on the time step due to the highly nonlinear terms of the strain energy density function considered in (2.2).

Finally, we obtain the following fully-discrete approximation of (2.3): for each $n = 1, \dots, N_t$, find $\mathbf{u}_h^n \in \mathbb{R}^{N_h}$ such that

$$\mathbf{R}(\mathbf{u}_h^n(\boldsymbol{\mu}), t^n; \boldsymbol{\mu}) = \mathbf{0}, \quad (3.3)$$

where \mathbf{u}_h^{n-1} and \mathbf{u}_h^0 are given, and

$$\mathbf{R}(\mathbf{u}_h^n(\boldsymbol{\mu}), t^n; \boldsymbol{\mu}) := \frac{\rho_0}{\Delta t^2} \mathcal{M} \mathbf{u}_h^n(\boldsymbol{\mu}) + \mathcal{S}(\mathbf{u}_h^n(\boldsymbol{\mu})) - \frac{2\rho_0}{\Delta t^2} \mathcal{M} \mathbf{u}_h^{n-1}(\boldsymbol{\mu}) + \frac{\rho_0}{\Delta t^2} \mathcal{M} \mathbf{u}_h^{n-2}(\boldsymbol{\mu}) - \mathcal{F}^{ext,n}(\boldsymbol{\mu}).$$

For the solution of the nonlinear problems arising at each time step, we use the Newton method, turning the nonlinear problem (3.3) into a sequence of linear problems of the following form: given $\boldsymbol{\mu} \in \mathcal{P}$, for each $n = 1, \dots, N_t$, given an initial guess $\mathbf{u}_h^{n,(0)}(\boldsymbol{\mu})$, for $k \geq 0$, find $\delta \mathbf{u}_h^{n,(k)}(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ such that

$$\begin{cases} \mathbf{J}(\mathbf{u}_h^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu}) \delta \mathbf{u}_h^{n,(k)}(\boldsymbol{\mu}) = -\mathbf{R}(\mathbf{u}_h^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu}), \\ \mathbf{u}_h^{n,(k+1)}(\boldsymbol{\mu}) = \mathbf{u}_h^{n,(k)}(\boldsymbol{\mu}) + \delta \mathbf{u}_h^{n,(k)}(\boldsymbol{\mu}), \end{cases} \quad (3.4)$$

until $\|\mathbf{R}(\mathbf{u}_h^{n,(k+1)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu})\|_2 / \|\mathbf{R}(\mathbf{u}_h^{n,(0)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu})\|_2 < \varepsilon$, where $\varepsilon > 0$ is a prescribed tolerance. Here, $\mathbf{J} \in \mathbb{R}^{N_h \times N_h}$ denotes the Jacobian matrix, whose components are given by

$$\mathbf{J}(\mathbf{u}_h^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu}) = \frac{\partial}{\partial \mathbf{u}_h} \mathbf{R}(\mathbf{u}_h^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu}).$$

The initial guess $\mathbf{u}_h^{n,(0)}(\boldsymbol{\mu})$ is set equal to the initial guess $\mathbf{u}_{h,0}(\boldsymbol{\mu})$, when $n = 1$, and is equal to the solution at previous time iteration $\mathbf{u}_h^{n-1}(\boldsymbol{\mu})$, for $n = 2, \dots, N_t$. From now on, we will refer to (3.4) as to the FOM for our problem.

Remark 1. As an alternative to Newton iterative scheme is, e.g., Broyden's quasi-Newton method [30], as done, e.g., in [16, 31] for the construction of ROMs in nonlinear elasticity problems. In this way we can avoid the computation of the Jacobian matrix at each iteration $k \geq 0$ by replacing it with rank-one updates, based on residuals computed at previous iterations.

4. Projection-based model order reduction

To mitigate the computational cost associated with the FOM solution, we introduce a projection-based ROM, by relying on the reduced basis (RB) method [8]. To make this paper self-contained, in this section we describe the construction of a POD-Galerkin-DEIM ROM for nonlinear time-dependent problems arising in cardiac mechanics, generalizing the strategies proposed in [15]. Details related to both POD and DEIM methods are reported in the Appendix.

4.1. POD-Galerkin method for solution-space reduction

The goal of the RB method for parameterized PDEs is to approximate the N_h -dimensional solution manifold

$$\mathcal{M}_h = \{\mathbf{u}_h^n(\boldsymbol{\mu}) \in \mathbb{R}^{N_h} \mid \boldsymbol{\mu} \in \mathcal{P}, n = 1, \dots, N_t\} \subset \mathbb{R}^{N_h}$$

with a low number N (less than a few dozens, or hundreds at most) of basis functions, forming the so-called *reduced basis*, whose nodal values are collected column-wise in the RB matrix $\mathbf{V} \in \mathbb{R}^{N_h \times N}$. This is usually done by performing a Galerkin projection of the high-fidelity problem onto the N -dimensional subspace spanned by the reduced basis functions, obtaining a reduced problem facing a much lower computational complexity, still respecting the structure of the underlying PDE and retaining the essential features of the parameter-to-solution map. More precisely, the reduction procedure consists of:

- 1) the construction of a reduced basis $\mathbf{V} = [\xi_1 \mid \dots \mid \xi_N] \in \mathbb{R}^{N_h \times N}$ from FOM solution snapshots

$$\mathbf{S}_u = \left[\mathbf{u}_h^1(\boldsymbol{\mu}_1) \mid \dots \mid \mathbf{u}_h^{N_t}(\boldsymbol{\mu}_1) \mid \dots \mid \mathbf{u}_h^1(\boldsymbol{\mu}_{n_s}) \mid \dots \mid \mathbf{u}_h^{N_t}(\boldsymbol{\mu}_{n_s}) \right] \quad (4.1)$$

obtained for different parameter values $\boldsymbol{\mu}_\ell \in \mathcal{P}$, $\ell = 1, \dots, n_s$, suitably sampled over \mathcal{P} ;

- 2) the definition of the ROM by forcing the high-fidelity residual vector $\mathbf{R} \in \mathbb{R}^{N_h}$ computed over the RB solution to be orthogonal to the subspace spanned by the columns of \mathbf{V} , that is

$$\mathbf{V}^T \mathbf{R}(\mathbf{V}\mathbf{u}_N^n(\boldsymbol{\mu}), t^n; \boldsymbol{\mu}) = \mathbf{0}, \quad n = 1, \dots, N_t; \quad (4.2)$$

- 3) the solution of (4.2) for a given $\boldsymbol{\mu} \in \mathcal{P}$, so that, at the end, $\mathbf{u}_h^n(\boldsymbol{\mu}) \approx \mathbf{V}\mathbf{u}_N^n(\boldsymbol{\mu})$, $n = 1, \dots, N_t$.

Two popular methods for the construction of the RB basis are the greedy algorithm [32], based on an a posteriori error estimator, and the proper orthogonal decomposition (POD), based on a singular value decomposition (SVD) of the snapshots matrix. Since for time-dependent, nonlinear problems error bounds are usually extremely difficult to obtain, in this work we rely on the POD method, which Section 4.2 is devoted to.

The solution to (4.2) can be found by employing Newton method, obtaining a sequence of N -dimensional linear systems: given $\boldsymbol{\mu} \in \mathcal{P}$ and, for $n = 1, \dots, N_t$, the initial guess $\mathbf{u}_N^{n,(0)}(\boldsymbol{\mu}) = \mathbf{u}_N^{n-1}(\boldsymbol{\mu})$, find $\delta\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}) \in \mathbb{R}^N$ such that, for $k \geq 0$,

$$\begin{cases} \mathbf{V}^T \mathbf{J}(\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu}) \mathbf{V} \delta\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}) = -\mathbf{V}^T \mathbf{R}(\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu}), \\ \mathbf{u}_N^{n,(k+1)}(\boldsymbol{\mu}) = \mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}) + \delta\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}), \end{cases} \quad (4.3)$$

until $\|\mathbf{V}^T \mathbf{R}(\mathbf{V}\mathbf{u}_N^{n,(k+1)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu})\|_2 / \|\mathbf{V}^T \mathbf{R}(\mathbf{V}\mathbf{u}_N^{n,(0)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu})\|_2 < \varepsilon_{nwt}$, where $\varepsilon_{nwt} > 0$ is a chosen tolerance.

The efficiency of the RB method relies, mainly, on two assumptions. First, that the solution manifold \mathcal{M}_h has low-dimension; second, that the reduction procedure can be split into *offline* and *online* stages, where the latter is completely independent of the high-fidelity dimension [33]. The first hypothesis concerns the approximability of the solution set, and is verified in the case of the problems we are focusing on, as shown in Section 5; the second assumption, unfortunately, does not hold for nonlinear problems as the ones we are considering. In fact, since both the residual vector \mathbf{R} and the Jacobian matrix \mathbf{J} in (4.3) depend on the solution at the previous iteration $\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu})$, they need to be assembled at each Newton step. This means that, in order to set up the reduced system (4.3)₁, we need to assemble

the high-dimensional arrays before projecting them onto the reduced space spanned by the columns of \mathbf{V} , entailing a computational complexity that still depends on (suitable powers of) N_h . To overcome this issue, a further level of approximation, known as hyper-reduction, must be introduced, thus pursuing an *approximate-then-reduce* strategy.

4.2. Reduced basis construction

For the construction of the reduced basis $\mathbf{V} \in \mathbb{R}^{N_h \times N}$, introduced in Section 4.1, we use the POD. Due to its ease of implementation, and its deep mathematical root (related to the analysis of compact operators, to matrix SVD, and to dimensionality reduction in data analysis, just to mention a few links), POD has been applied in a broad range of engineering fields to reduce the dimension of a given data set, in an optimal sense. In this work, POD is used to build the RB basis \mathbf{V} , as well as for the construction of the DEIM basis $\mathbf{\Phi}_R$ for the nonlinear terms (see Section 4.3).

Given the snapshot matrix \mathbf{S}_u defined in (4.1), with $n_s < N_h$, POD aims at approximating the solution manifold with a low-dimensional linear subspace, retaining as much as possible of the information gathered in the snapshots. In particular, the N -dimensional POD basis is obtained by computing the SVD $\mathbf{S}_u = \mathbf{U}\mathbf{\Sigma}\mathbf{Z}^T$ of \mathbf{S}_u , and then collecting the first N columns of $\mathbf{U} \in \mathbb{R}^{N_h \times N_h}$ corresponding to the left singular vectors, i.e.,

$$\mathbf{V} = [\boldsymbol{\xi}_1 \mid \dots \mid \boldsymbol{\xi}_N].$$

The singular values $\sigma_1 \geq \dots \geq \sigma_r > 0$, where $r \leq \min(N_h, n_s)$ is the rank of \mathbf{S} , provide a heuristic criteria for choosing the basis dimension N

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

where $\varepsilon_{POD} > 0$ is a given tolerance (see also Appendix A). A rapid decay of the singular values means that a limited number of POD modes is potentially sufficient to represent the entire manifold, so that the problem is reducible. This is exactly the kind of situation faced when dealing with the reduction of the solution space related to a problem in nonlinear elastodynamics. Moreover, an efficient, non-deterministic version of POD can be obtained by relying on the so called randomized-SVD (see Appendix B), which offers a powerful tool for performing low-rank matrix approximation, especially when dealing with massive data sets, as in the cases we have considered.

4.3. DEIM for hyper-reduction/system approximation

In the case of PDEs featuring nonaffine dependence on the parameters and/or nonlinear (high-order polynomial, or non-polynomial) dependence on the field variable, a further level of reduction must be introduced to guarantee the offline-online decoupling in the ROM construction [33]. To recover the ROM efficiency, state-of-the-art methods, such as the empirical interpolation method (EIM) [34–36], the discrete empirical interpolation method (DEIM) [37], its variant matrix DEIM [38–40], the missing point estimation [41] and the Gauss-Newton with approximated tensors [42], aim at recover an affine expansion of the nonlinear operators by computing only a few entries of the nonlinear terms.

Although originally developed in the context of nonaffine operators, DEIM represents a valid hyper-reduction technique also for nonlinear parameterized PDEs (see, e.g., [15, 31, 43–46]), employing an interpolation scheme for the approximation of the nonlinear function. The key idea of DEIM is to

replace the nonlinear arrays in (4.3) with a collateral reduced-basis expansion, computed through a (hopefully, inexpensive) interpolation procedure. In the case of DEIM, the construction of the interpolation points, commonly referred to as *magic points*, is based on a greedy algorithm, while the (prior) construction of the reduced basis is obtained by performing POD (or randomized-SVD) on a set of proper snapshots; in the case of EIM, both tasks are performed at the same time, exploiting a greedy algorithm. The DEIM algorithm, as originally proposed in [37], is outlined in Appendix C.

For the case at hand, the high-dimensional residual \mathbf{R} is projected onto a reduced subspace of dimension $m < N_h$ spanned by a basis $\Phi_{\mathcal{R}} \in \mathbb{R}^{N_h \times m}$

$$\mathbf{R}(\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu}) \approx \Phi_{\mathcal{R}} \mathbf{c}(\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu}),$$

where $\mathbf{c} \in \mathbb{R}^m$ is the vector of the unknown amplitudes. The matrix $\Phi_{\mathcal{R}}$ can be pre-computed offline by performing POD on a set of high-fidelity residuals collected when solving (4.3) for n'_s training input parameters (different from the one used for the RB basis construction),

$$\mathbf{S}_R = \left[\mathbf{R}(\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}_{\ell'}), t^n; \boldsymbol{\mu}_{\ell'}), k \geq 0 \right]_{n=1:N_t}^{\ell'=1:n'_s}. \quad (4.4)$$

Remark 2. In addition to the set of snapshots (4.4), we can consider the FOM residuals collected when solving (3.4) during the RB-basis construction, i.e.,

$$\mathbf{S}_R = \mathbf{S}_R^{FOM} \cup \mathbf{S}_R^{ROM} = \left[\mathbf{R}(\mathbf{u}_h^{n,(k)}(\boldsymbol{\mu}_{\ell'}), t^n; \boldsymbol{\mu}_{\ell'}), k \geq 0 \right]_{n=1:N_t}^{\ell'=1:n_s} \cup \left[\mathbf{R}(\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}_{\ell'}), t^n; \boldsymbol{\mu}_{\ell'}), k \geq 0 \right]_{n=1:N_t}^{\ell'=1:n'_s}.$$

Taking into account both FOM and ROM residuals entail no extra computational cost and usually improves the overall accuracy, assuming $n'_s > n_s$. On the contrary, collecting FOM residuals only gave inaccurate results for all the performed test cases, as shown, for the stationary case, in [15]. In fact, DEIM aims at approximating the nonlinear operators evaluated at the ROM solution, rather than at the FOM solution. Therefore, the construction of the reduced space and system approximation must be performed sequentially.

For every new instance of the parameter, the $\boldsymbol{\mu}$ -dependent coefficient vector \mathbf{c} is efficiently evaluated online by collocating the approximation at the m components selected by a greedy procedure, that is,

$$\Phi_{\mathcal{R}|_{\mathcal{I}}} \mathbf{c}(\mathbf{V}\mathbf{u}_N^{n,(k)}, t^n; \boldsymbol{\mu}) = \mathbf{R}(\mathbf{V}\mathbf{u}_N^{n,(k)}, t^n; \boldsymbol{\mu})|_{\mathcal{I}},$$

where $\Phi_{\mathcal{R}|_{\mathcal{I}}}$ and $\mathbf{R}(\cdot)|_{\mathcal{I}}$ are the restrictions of $\Phi_{\mathcal{R}}$ and $\mathbf{R}(\cdot)$ to the subset of indices \mathcal{I} , respectively. We thus define the hyper-reduced residual vector approximating $\mathbf{V}^T \mathbf{R}(\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu})$ as

$$\mathbf{R}_{N,m}(\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu}) := \mathbf{V}^T \Phi_{\mathcal{R}}(\Phi_{\mathcal{R}}^{-1})|_{\mathcal{I}} \mathbf{R}(\mathbf{V}\mathbf{u}_N^{n,(k)}, t^n; \boldsymbol{\mu})|_{\mathcal{I}}.$$

During the online phase we need to assemble the $\boldsymbol{\mu}$ -dependent quantities $\mathbf{R}(\mathbf{V}\mathbf{u}_N^{n,(k)}, t^n; \boldsymbol{\mu})|_{\mathcal{I}}$ only, which are vectors of (possibly small) dimension m . All other quantities are constant (in fact, $\Phi_{\mathcal{R}}$ does not depend on $t > 0$, nor on $\boldsymbol{\mu} \in \mathcal{P}$) and can be pre-computed and stored offline. Finally, the Jacobian approximation to $\mathbf{V}^T \mathbf{J}(\mathbf{V}\mathbf{u}_N^{n,(k)}, t^n; \boldsymbol{\mu}) \mathbf{V}$ can be computed as the derivative of $\mathbf{R}_{N,m}(\mathbf{V}\mathbf{u}_N^{n,(k)}, t^n; \boldsymbol{\mu})$ with respect to the reduced displacement, that is,

$$\begin{aligned} \mathbf{J}_{N,m}(\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu}) &:= \frac{\partial \mathbf{R}_{N,m}(\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu})}{\partial \mathbf{u}_N} \\ &= \mathbf{V}^T \Phi_{\mathcal{R}}(\Phi_{\mathcal{R}}^{-1})|_{\mathcal{I}} \frac{\partial \mathbf{R}(\mathbf{V}\mathbf{u}_N^{n,(k)}, t^n; \boldsymbol{\mu})}{\partial \mathbf{u}_N} |_{\mathcal{I}} = \mathbf{V}^T \Phi_{\mathcal{R}}(\Phi_{\mathcal{R}}^{-1})|_{\mathcal{I}} \mathbf{J}(\mathbf{V}\mathbf{u}_N^{n,(k)}, t^n; \boldsymbol{\mu})|_{\mathcal{I}} \mathbf{V} \end{aligned}$$

or by relying on the MDEIM algorithm, as done in [15]. However, since we employ automatic differentiation to get (an approximation of) the Jacobian matrices, we adopt the former approximation. As before, the only quantity that must be computed online is the restriction of the Jacobian matrix to the rows corresponding to the magic points, i.e., $\mathbf{J}(\mathbf{V}\mathbf{u}_N^{n,(k)}, t^n; \boldsymbol{\mu})|_{\mathcal{I}} \in \mathbb{R}^{m \times N_h}$. Note that employing DEIM can be regarded as the use of an exact Newton method on the reduced problem $\mathbf{R}_{N,m}(\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu}) = \mathbf{0}$, such that the k -th Newton iteration for its solution reads

$$\begin{cases} \mathbf{J}_{N,m}(\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu})\delta\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}) = -\mathbf{R}_{N,m}(\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}), t^n; \boldsymbol{\mu}), \\ \mathbf{u}_N^{n,(k+1)}(\boldsymbol{\mu}) = \mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}) + \delta\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}). \end{cases} \quad (4.5)$$

DEIM thus avoids any full-order evaluation, highly decreasing the computational effort, provided its dimension is not too large [9]. Unfortunately, as we will see in the following section, when dealing with problems arising in nonlinear elastodynamics characterized by highly nonlinear constitutive laws, the DEIM dimension cannot be kept small, if aiming at ensuring a sufficient accuracy level.

In Algorithm 1 and 2 we report the offline stage and the online stage of the POD-Galerkin-DEIM ROM, respectively.

Algorithm 1 POD-Galerkin-DEIM for nonlinear time-dependent problems. Offline stage.

INPUT: $\boldsymbol{\mu}_\ell$, for $\ell = 1, \dots, n_s$, and $\boldsymbol{\mu}_{\ell'}$, for $\ell' = 1, \dots, n'_s$

OUTPUT: $\mathbf{V} \in \mathbb{R}^{N_h \times N}$, $\Phi_{\mathcal{R}} \in \mathbb{R}^{N_h \times m}$, $\mathcal{I} = \{i_1, \dots, i_m\}$

- 1: **for** $\ell = 1, \dots, n_s$ **do**
 - 2: **for** $n = 1, \dots, N_t$ **do**
 - 3: **for** $k \geq 0$ **until convergence do**
 - 4: Assemble and solve problem (3.4)
 - 5: Collect $\mathbf{S}_u \leftarrow \mathbf{S}_u \cup [\mathbf{u}_h^{n,(k)}(\boldsymbol{\mu}_\ell)]$ columns-wise
 - 6: Collect $\mathbf{S}_R \leftarrow \mathbf{S}_R \cup [\mathbf{R}(\mathbf{u}_h^{n,(k)}(\boldsymbol{\mu}_\ell), t^n; \boldsymbol{\mu}_{\ell'})]$ columns-wise (see Remark 2)
 - 7: Construct $\mathbf{V} = \text{POD}(\mathbf{S}_u, \varepsilon_{\text{POD}})$ (see Algorithm 3)
 - 8: **for** $\ell' = 1, \dots, n'_s$ **do**
 - 9: **for** $n = 1, \dots, N_t$ **do**
 - 10: **for** $k \geq 0$ **until convergence do**
 - 11: Assemble and solve reduced problem (4.3)
 - 12: Collect $\mathbf{S}_R \leftarrow \mathbf{S}_R \cup [\mathbf{R}(\mathbf{V}\mathbf{u}_N^{n,(k)}(\boldsymbol{\mu}_{\ell'}), t^n; \boldsymbol{\mu}_{\ell'})]$ columns-wise
 - 13: Construct $(\Phi_{\mathcal{R}}, \mathcal{I}) = \text{DEIM}(\mathbf{S}_R, \varepsilon_{\text{DEIM}})$ (see Algorithm 5)
-

Algorithm 2 POD-Galerkin-DEIM for nonlinear time-dependent problems. Online stage.

INPUT: $\boldsymbol{\mu} \in \mathcal{P}$

OUTPUT: $\mathbf{V}\mathbf{u}_N^n(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$, for $n = 1, \dots, N_t$

- 1: **for** $n = 0, \dots, N_t - 1$ **do**
 - 2: **for** $k \geq 0$ **until convergence do**
 - 3: Assemble and solve hyper-reduced problem (4.5)
 - 4: Recover $\mathbf{V}\mathbf{u}_N^n(\boldsymbol{\mu})$, for $n = 1, \dots, N_t$
-

Remark 3. The m points selected by the DEIM algorithm correspond to a subset of nodes of the computational mesh, which, together with the neighboring nodes (i.e., those sharing the same cell),

form the so-called reduced mesh (see Figure 1). Since the entries of any FE-vector are associated with the dofs of the problem, $\mathbf{R}_{N,m}$ and $\mathbf{J}_{N,m}$ can be computed by integrating the corresponding FOM residual and Jacobian only on the quadrature points belonging to the reduced mesh, respectively.

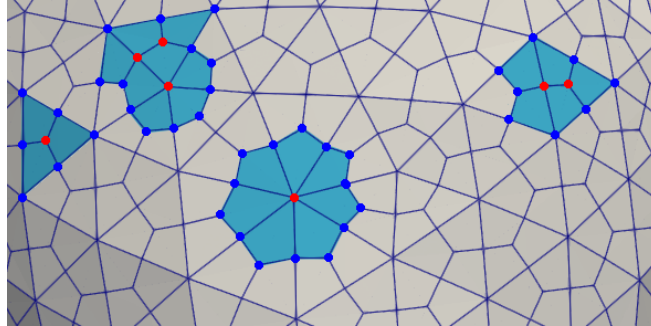


Figure 1. Sketch of a reduced mesh for an hexahedral computational grid in a two-dimensional case. The red dots represent the magic points selected by the DEIM algorithm.

Remark 4. We recall that both the FOM and the ROM arrays, such as the solution and the residual vectors, are column vectors whose elements are the values of the associated quantities evaluated on the dofs of the physical mesh. Therefore, when assembling the snapshots matrices in order to compute the RB and the DEIM basis, the corresponding arrays are stacked column-wise.

5. Numerical results

In this section we investigate the performances of POD-Galerkin-DEIM on the solution to the parameterized nonlinear time-dependent mechanical problem (2.3), focusing on cardiac applications, namely:

- (i) two test cases on an idealized left ventricle geometry, simulating cardiac relaxation and contraction, respectively; the steady-state versions of these test cases have been introduced in [47] as benchmarks for the validation of cardiac mechanics software;
- (ii) an idealized full cycle of a patient-specific left ventricle, where both pressure and active stress are imposed.

In all these scenarios, the traction vector $\bar{\mathbf{T}}$ is given by

$$\bar{\mathbf{T}}(\mathbf{X}, t, \mathbf{N}; \boldsymbol{\mu}) = -\mathbf{g}(t; \boldsymbol{\mu}) \mathbf{J} \mathbf{F}^{-T} \mathbf{N},$$

where $\mathbf{g}(t; \boldsymbol{\mu})$ represents blood pressure exerting on the endocardium and will be further specified, according to the application at hand. As a measure of the accuracy of the ROM with respect to the FOM, for a given parameter instance, we consider time-averaged L^2 -errors of the displacement vector, that are,

$$\epsilon_{abs}(\boldsymbol{\mu}) = \frac{1}{N_t} \sum_{n=1}^{N_t} \|\mathbf{u}_h(\cdot, t_n; \boldsymbol{\mu}) - \mathbf{V} \mathbf{u}_N(\cdot, t_n; \boldsymbol{\mu})\|_2, \quad \epsilon_{rel}(\boldsymbol{\mu}) = \frac{1}{N_t} \sum_{n=1}^{N_t} \frac{\|\mathbf{u}_h(\cdot, t_n; \boldsymbol{\mu}) - \mathbf{V} \mathbf{u}_N(\cdot, t_n; \boldsymbol{\mu})\|_2}{\|\mathbf{u}_h(\cdot, t_n; \boldsymbol{\mu})\|_2}.$$

The CPU time ratio, i.e., the ratio between FOM and ROM computational times, is used to measure the ROM efficiency, since it represents the speed-up of the ROM with respect to the FOM. The code is implemented in Python in our software package `pyfex`, which contains a Python binding with the in-house Finite Element library `lifex` (<https://lifex.gitlab.io/lifex>), a high-performance C++ library developed within the iHEART project* and based on the `deal.II` (<https://www.dealii.org>) Finite Element core [48]. All the computations have been performed on a PC desktop computer with 3.70GHz Intel Core i5-9600K CPU and 16GB RAM.

5.1. Benchmark problems with a prolate spheroid geometry

We first investigate the performances of a POD-Galerkin-DEIM reduced order models to address benchmark problems in cardiac mechanics. In both cases, the reference geometry $\Omega_0 \subset \mathbb{R}^3$ is that of a truncated ellipsoid and the material law adopted is the nearly-incompressible Guccione relation (2.2), although different parameter values are taken into account, according to [47]. For what concerns the boundary condition of problem (2.3), we apply a linear external pressure

$$\mathbf{g}(t; \boldsymbol{\mu}) = \tilde{p} t/T$$

at the endocardium (i.e., Γ_0^N), with $\tilde{p} > 0$, simulating the presence of blood inside the cardiac chamber, and consider homogeneous Neumann and Dirichlet conditions at the epicardium (i.e., Γ_0^R with $\alpha = \beta = 0$) and on the base (i.e., Γ_0^D), respectively, the latter representing the artificial boundary resulting from truncation of the heart below the valves in a short axis plane.

The FOM is built on a hexahedral mesh with 4804 elements and 6455 vertices, depicted in Figure 2, resulting in a high-fidelity dimension $N_h = 19365$, since \mathbb{Q}_1 -FE (that is, bilinear FE on a hexahedral mesh) are used. We point out that having chosen a rather coarse computational mesh results in moderate speed-ups of the ROM with respect to the FOM in the considered test cases. However, taking into account finer meshes, thus larger high-fidelity dimensions N_h , higher speed-ups can be achieved by the POD-Galerkin-DEIM ROMs, since the reduced basis dimension N remains small (see Section 5.1.2).

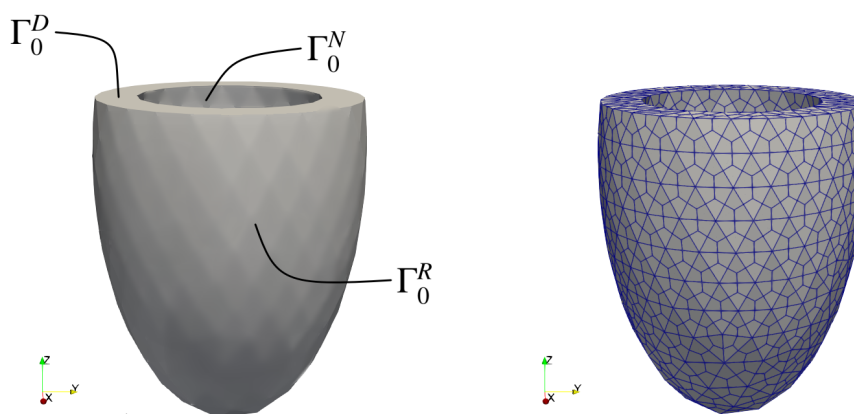


Figure 2. Test case 1. Idealized truncated ellipsoid geometry and computational grid.

*iHEART - An Integrated Heart Model for the simulation of the cardiac function, European Research Council (ERC) grant agreement No 740132, P.I. Prof. A. Quarteroni.

5.1.1. Passive inflation of a ventricle

In this first case, we do not account for anisotropy and only consider the passive contribution of the Piola tensor, so that the obtained deformation pattern simulates passive ventricular diastole. As parameters, we consider

- the material stiffness in fiber and cross-fiber directions $b_f, b_n \in [0.8, 1.2]$,
- the multiplicative factor $C \in [8 \cdot 10^3, 12 \cdot 10^3]$ Pa.

All the other parameters are fixed to their reference values: $b_s = b_{fs} = b_{fn} = b_{sn} = 1$, $K = 50 \cdot 10^3$ Pa and $\bar{p} = 10 \cdot 10^3$. For the time setting, we choose $t \in (0, 0.25)$ s and a uniform time step $\Delta t = 5 \cdot 10^{-3}$ s, resulting in a total number of 50 time instances. In Figure 3 we report the high-fidelity displacement at different time instants. In order to compute the FOM solution, almost 340 s are required in average.

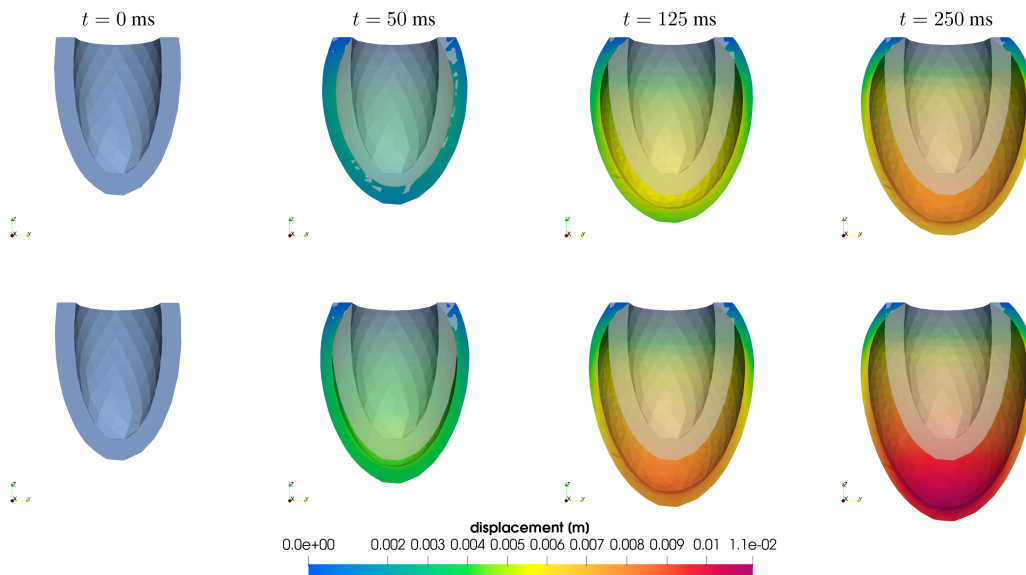


Figure 3. Test case 1, passive inflation of a ventricle. FOM solution computed at different times for $\mu = [0.881, 1.171, 11650]$ (top) and $\mu = [1.007, 0.853, 8610]$ (bottom).

For $n_s = 50$ points sampled in the parameter space \mathcal{P} through latin hypercube sampling (LHS), we collect the solution snapshots in order to compute the RB basis $\mathbf{V} \in \mathbb{R}^{N_s \times N}$. The singular values arising from the SVD of the snapshots matrix \mathbf{S}_u are reported in Figure 4, where a rapid decay of the reported quantity is observed. We expect that a small number of basis functions is sufficient for the ROM to guarantee a good approximation of the high-fidelity solution manifold. To assess the performance of the method with respect to the dimension of the reduced subspace, we consider

$$\varepsilon_{POD} \in \{10^{-3}, 5 \cdot 10^{-4}, 10^{-4}, 5 \cdot 10^{-5}, 10^{-5}, 5 \cdot 10^{-6}, 10^{-6}\},$$

as POD tolerances, corresponding to a RB dimension of $N = 5, 7, 12, 16, 26, 32$ and 55 , respectively.

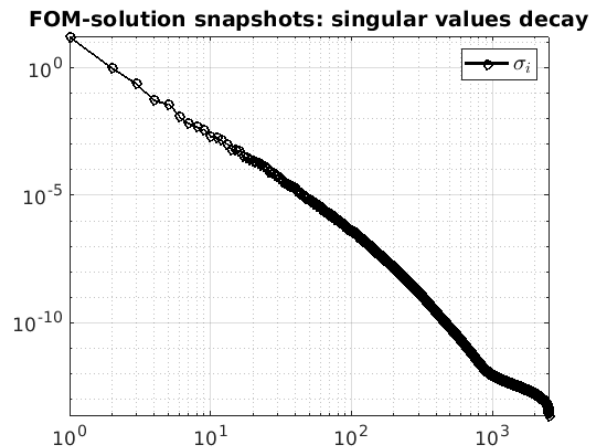


Figure 4. Test case 1, passive inflation of a ventricle. Decay of the singular values of the FOM solution snapshots matrix.

First, we investigate the performance of the ROM without hyper-reduction. Figure 5 shows the error and the CPU time ratio over ε_{POD} , where the average is computed over a testing set of 20 parameters, different from those used for the generation of FOM snapshots. The plot confirms that few basis functions are required to accurately approximate the high-fidelity displacements.

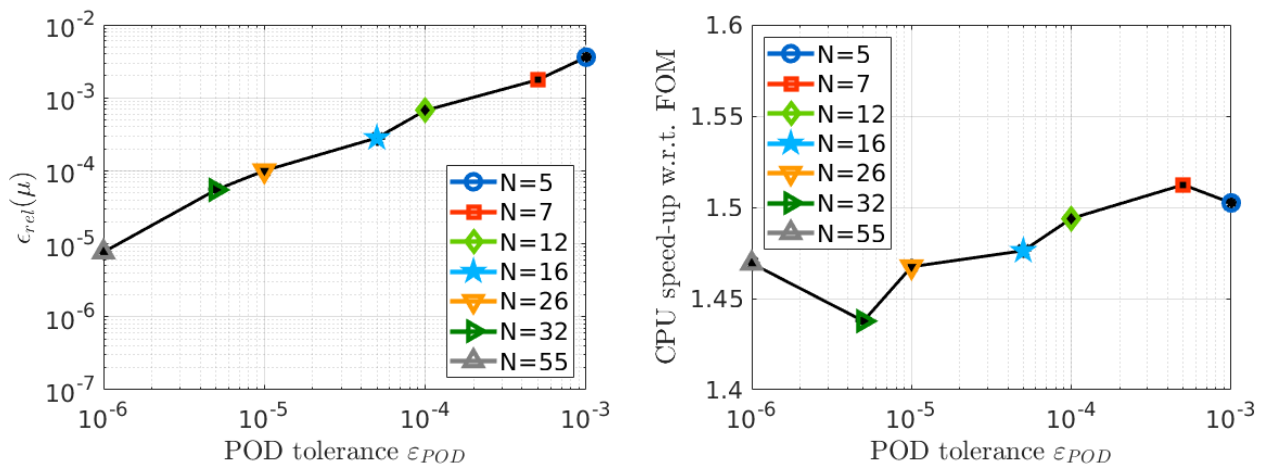


Figure 5. Test case 1, passive inflation of a ventricle. Average over 20 testing parameters of relative error ε_{rel} (left) and average speed-up (right) of the POD-Galerkin ROM without hyper-reduction.

In particular, we observe that the relative error ε_{rel} associated with the ROM approximation decreases of almost three orders of magnitude when going from $N = 5$ to $N = 55$. However, since at each Newton step we need to assemble high-dimensional arrays before projecting them onto the reduced space, the CPU time required by the ROM for all RB dimensions, despite decreasing as N becomes smaller, is almost comparable to the one of the FOM.

Then, we investigate the impact of hyper-reduction on the ROM solution error and efficiency. In

order to build the POD-Galerkin-DEIM hyper-ROM, we consider $N = 16$ basis functions for state reduction, obtained for $\varepsilon_{POD} = 10^{-5}$. For the construction of the residual basis $\Phi_R \in \mathbb{R}^{N_h \times m}$, we rely on a snapshots set computed for $n'_s = 50$ points in \mathcal{P} and apply POD on \mathbf{S}_R using

$$\varepsilon_{DEIM} \in \{5 \cdot 10^{-4}, 10^{-4}, 5 \cdot 10^{-5}, 10^{-5}, 5 \cdot 10^{-6}, 10^{-6}\},$$

corresponding to $m = 84, 127, 148, 207, 239$ and 368 , and compare these options.

As shown in Table 1 and Figure 6, no loss of accuracy is experienced when decreasing the number of DEIM interpolation points m . However, higher tolerances ε_{DEIM} on the DEIM error would not ensure the convergence of the reduced Newton system for all testing parameters, such that no higher speed-up can be achieved by further reducing the size of the residual basis. Furthermore, we point out that $m > N$, meaning that the residual vector shows higher variability with respect to the displacement. The POD-Galerkin-DEIM hyper-ROM with $m = 84$ is able to achieve a speed-up of $\times 9.3$ compared to the FOM, still achieving an approximation error equals to the projection error, that is $\varepsilon_{rel} \approx 10^{-3}$.

Table 1. Test case 1, passive inflation of a ventricle. Computational data related to POD-Galerkin-DEIM with $N = 16$ and different values of m .

	$m = 84$	$m = 127$	$m = 368$
Computational speed-up	$\times 9.3$	$\times 8.0$	$\times 4.3$
Avg. CPU time	36 s	42 s	77 s
Time-avg. $L^2(\Omega_0)$ -absolute error	$5.3 \cdot 10^{-5}$	$2.9 \cdot 10^{-5}$	$2.9 \cdot 10^{-5}$
Time-avg. $L^2(\Omega_0)$ -relative error	$5.3 \cdot 10^{-4}$	$2.9 \cdot 10^{-4}$	$2.8 \cdot 10^{-4}$

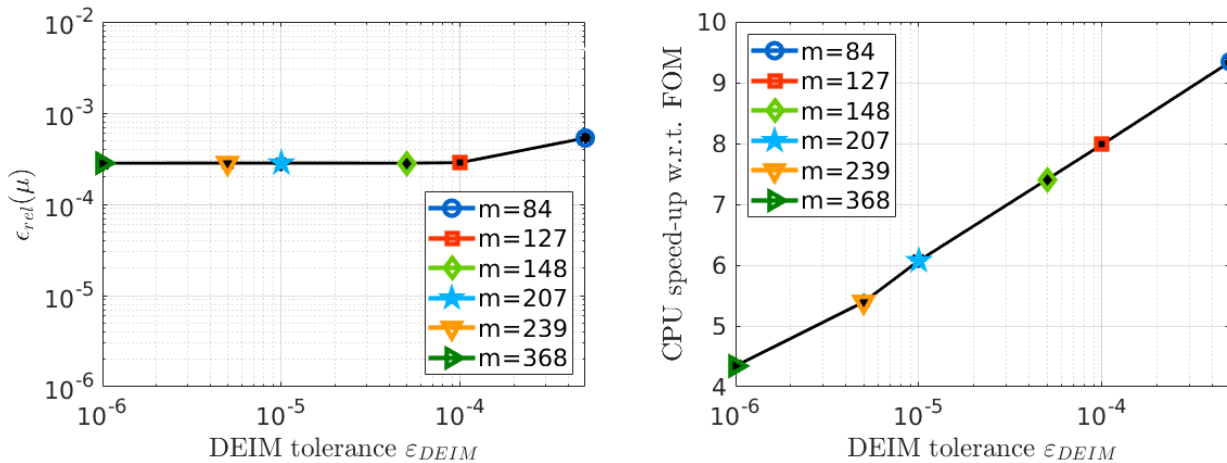


Figure 6. Test case 1, passive inflation of a ventricle. Average over 20 testing parameters of relative error ε_{rel} (left) and average speed-up (right) of the POD-Galerkin-DEIM hyper-ROM with $N = 16$ and different values of m .

5.1.2. Passive inflation and active contraction of a ventricle

The second benchmark takes into account both a varying fiber distribution and contractile forces, dealing with the inflation and the active contraction of an idealized left ventricle with transversely

isotropic material properties. Although it is still an idealized test case, the displacement field reproduces the typical twisting motion of the systole in the left ventricle, caused by the distribution of the muscular fibers. To compute the fiber orientation in cardiac geometries, suitable rule-based methods have been developed [49–52], which usually depend on a set of parameter angles. In this particular case, we consider the method proposed in [53], where α^{epi} and α^{endo} represent possible values of the rotation angle of the fibers on the epicardium and endocardium, respectively.

Moreover, we surrogate the presence of the active generation forces driving the contraction mechanics exploiting an active stress approach, and considering anisotropic active tension applied in the fiber direction only, see Section 2.1. In particular, the parameterized active tension $T_a = T_a(t; \boldsymbol{\mu})$ in the fiber direction is modeled as a linear function of the form

$$T_a(t; \boldsymbol{\mu}) = \widetilde{T}_a t / T.$$

To assess the performance of a POD-Galerkin-DEIM hyper-ROM to reduce the myocardium contraction, we consider as unknown parameters:

- the maximum value of the active tension $\widetilde{T}_a \in [49.5 \cdot 10^3, 70.5 \cdot 10^3]$ Pa,
- the fiber angles $\alpha^{epi} \in [-105.5, -74.5]^\circ$ and $\alpha^{endo} \in [74.5, 105.5]^\circ$,

both related to the active components of the strain energy functions. In particular, higher values of the active tension correspond to a larger contraction of the cardiac muscle, whilst the fiber angles influence the twisting motion. All the other parameters are fixed to their reference values, namely $b_f = 8$, $b_s = b_n = b_{sn} = 2$, $b_{fs} = b_{fn} = 4$, $C = 2 \cdot 10^3$ Pa, $K = 50 \cdot 10^3$ Pa and $\bar{p} = 15 \cdot 10^3$. Regarding time discretization, we choose $t \in (0, 0.25)$ s and a uniform time step $\Delta t = 5 \cdot 10^{-3}$ s. Given a training set of $n_s = 50$ points obtained by sampling the parameter space \mathcal{P} , we perform a convergence analysis of the ROM without hyper-reduction by constructing the reduced basis $\mathbf{V} \in \mathbb{R}^{N_h \times N}$ for different values of N and computing the associated approximation errors.

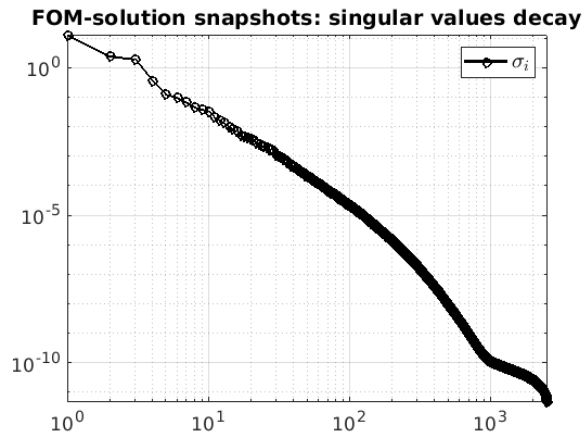


Figure 7. Test case 1, passive inflation and active contraction of a ventricle. Decay of the singular values of the FOM solution snapshots matrix.

From Figure 7, we observe a slower decay of the singular values of \mathbf{S}_u with respect to the previous test case of passive inflation (note the different values on the y axis). In fact, using POD tolerances

$$\varepsilon_{POD} \in \{10^{-3}, 5 \cdot 10^{-4}, 10^{-4}, 5 \cdot 10^{-5}, 10^{-5}, 5 \cdot 10^{-6}, 10^{-6}\},$$

we obtain the RB dimensions $N = 16, 22, 39, 50, 87, 109$ and 178 , respectively, much larger than the ones obtained for similar tolerances on the previous test case. This behavior is somehow expected, as the underlying system dynamics, simulating ventricular contraction and associated torsion, is more involved than the idealized diastole, in which tissue anisotropy is neglected.

As shown by the results obtained so far, the construction of a ROM is highly problem dependent, since the parameters considered, e.g., in the constitutive relation, strongly influence the form of the solution manifold, thus the RB dimension necessary to obtain comparable accuracy between the different ROMs. The error and the CPU speed-ups averaged over a testing set of 20 parameters are both shown in Figure 8, as functions of the POD tolerance ε_{POD} . As already discussed, although we observe lower online CPU times when smaller RB dimensions N are employed, the speed-up achieved by the ROM without hyper-reduction is negligible. For what concerns the approximation error, we observe a reduction of almost two orders of magnitude when going from $N = 16$ to $N = 178$.

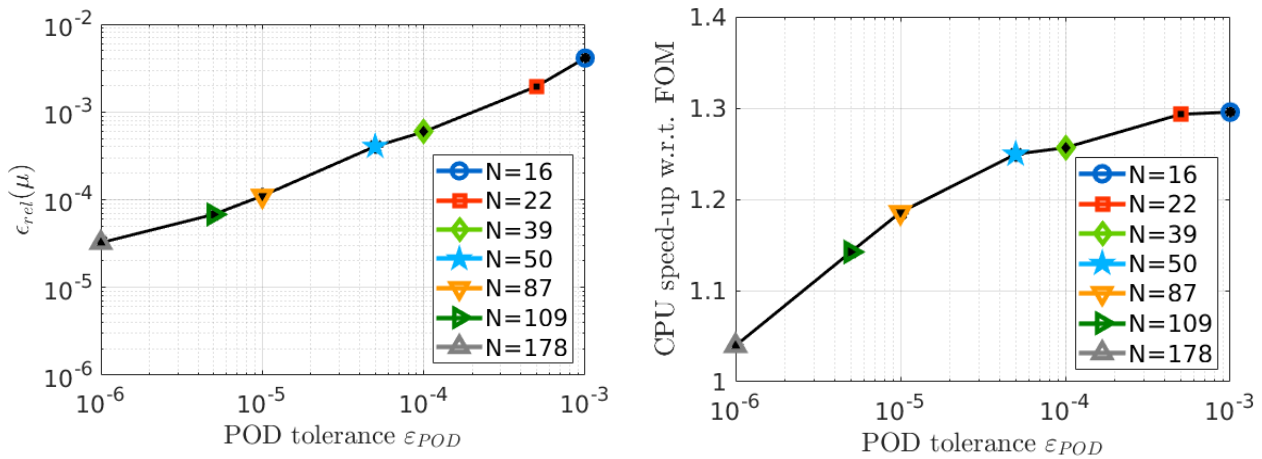


Figure 8. Test case 1, passive inflation and active contraction of a ventricle. Average over 20 testing parameters of relative error ε_{rel} (left) and average speed-up (right) of ROM without hyper-reduction.

Given $\mathbf{V} \in \mathbb{R}^{N_h \times N}$ with $N = 16$, we construct the POD-Galerkin-DEIM approximation by considering $n'_s = 200$ parameter samples. Figure 9 shows the decay of the singular values of \mathbf{S}_R , that is, the snapshots matrix of the residual vectors $\mathbf{R}(\mathbf{V}\mathbf{u}_N^{n_s(k)}(\boldsymbol{\mu}_{\ell'}), t^n; \boldsymbol{\mu}_{\ell'})$. We observe that the reported curve decreases very slowly, so that we expect a large number of basis functions to be required to correctly approximate the nonlinear operators. In fact, by computing $\Phi_{\mathcal{R}} \in \mathbb{R}^{N_h \times m}$ using

$$\varepsilon_{DEIM} \in \{5 \cdot 10^{-4}, 10^{-4}, 5 \cdot 10^{-5}, 10^{-5}, 5 \cdot 10^{-6}, 10^{-6}\}$$

as DEIM tolerances, we obtain $m = 303, 456, 543, 776, 902$ and 1233 , respectively. Higher values of ε_{DEIM} (related to hopefully smaller dimensions m) were not sufficient to guarantee the convergence of the reduced Newton problem for all the parameter combinations considered. The average relative error over a set of 20 parameters and the computational speed-up are both reported in Figure 10. In particular, we observe that the relative error is between $4 \cdot 10^{-3}$ and $8 \cdot 10^{-3}$, indeed quite close to the ones reported in Figure 8. In this respect, provided a sufficient DEIM dimension is considered, hyper-reduction does not impact significantly on the ROM accuracy. DEIM performances are not influenced

by the size of the sample considered for the residual arrays. Indeed, collecting a smaller number n'_s of samples would only have small effects on the approximation error, as shown in Figure 11.

Similar results are obtained when using a finer mesh with 13025 vertices, corresponding to $N_h = 39075$ dofs for the FOM. In this case, a reduced basis of dimension $N = 16$ is computed using $\varepsilon_{POD} = 10^{-3}$, while the POD tolerance ε_{DEIM} for the approximation of the nonlinear arrays has to be chosen no larger than 10^{-4} to ensure Newton convergence, obtaining $m = 385$ DEIM terms and a corresponding approximation error $\varepsilon_{rel} \approx 5 \cdot 10^{-2}$. Also in this case, the highest speed-up achieved by the ROM is greater than that obtained for the coarser mesh (i.e., $\times 8$ against $\times 6$), suggesting that the POD-Galerkin-DEIM convenience grows as the dimension of the underlying FOM increases.

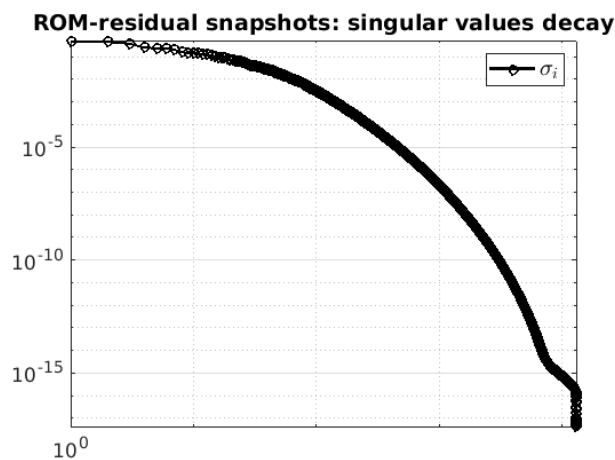


Figure 9. Test case 1, passive inflation and active contraction of a ventricle. Decay of the singular values of the ROM residual snapshots matrix.

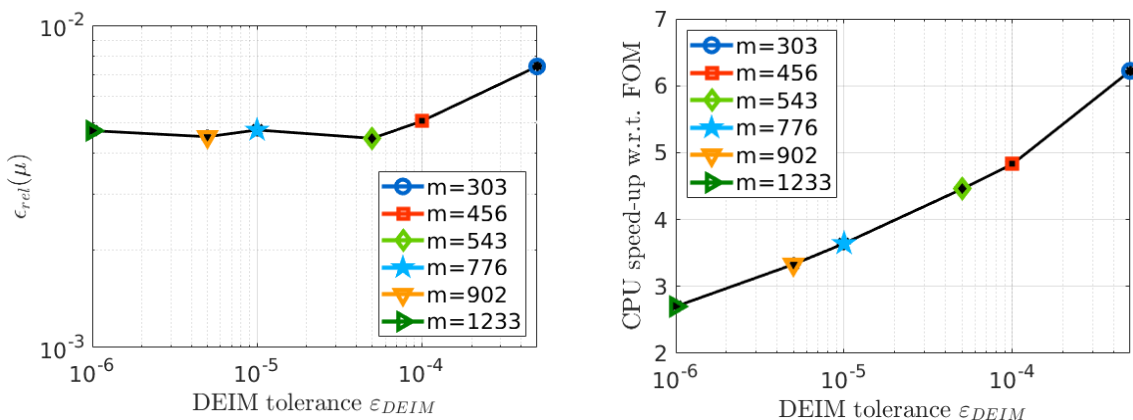


Figure 10. Test case 1, passive inflation and active contraction of a ventricle. Average over 20 testing parameters of relative error ε_{rel} (left) and average speed-up (right) of POD-Galerkin-DEIM with $N = 16$.

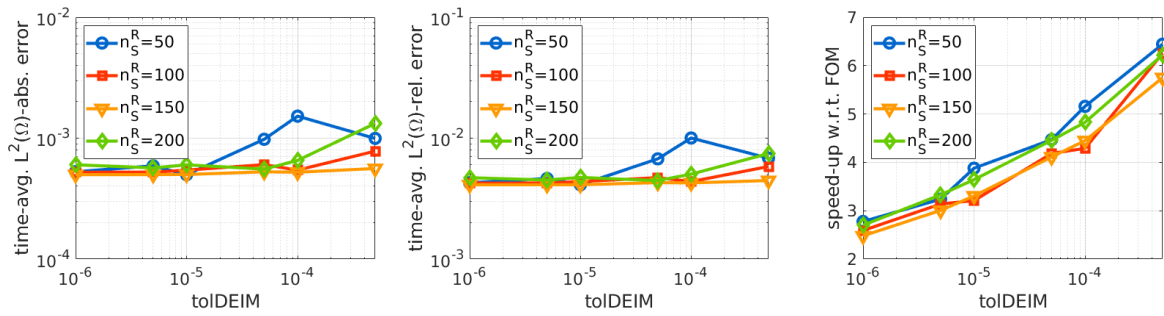


Figure 11. Test case 1, passive inflation and active contraction of a ventricle. Average on 20 testing parameters of the time-averaged $L^2(\Omega_0)$ -absolute error (left), the time-averaged $L^2(\Omega_0)$ -relative error (center) and the average speed-up (right) over ε_{DEIM} with respect to the FOM, when $n_s^R = 50$ (blue circle), 100 (red square), 150 (green diamond), 200 (orange downward facing triangle) training samples are used.

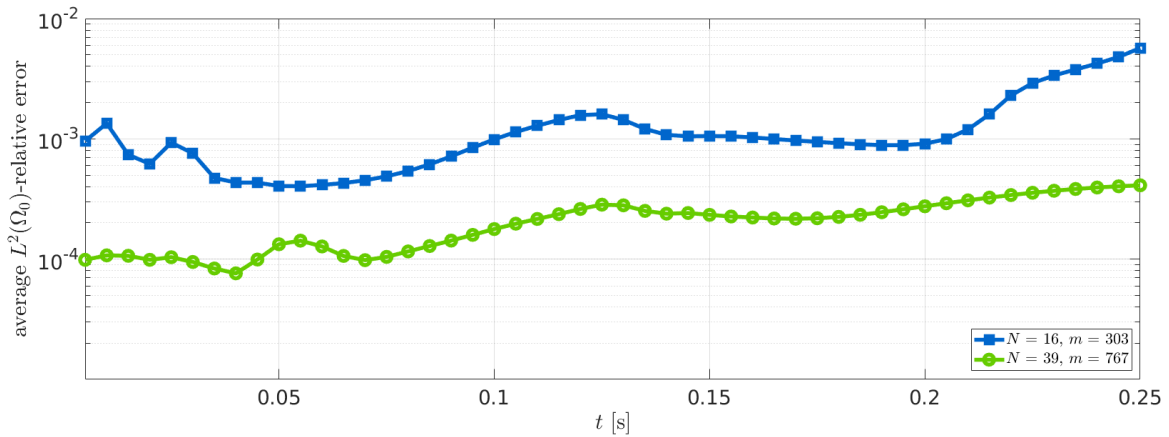


Figure 12. Test case 1, passive inflation and active contraction of a ventricle. Evolution in time of the average $L^2(\Omega_0)$ -relative error for $(N, m) = (16, 303)$ (blue squares) and $(N, m) = (39, 767)$ (green circles).

Table 2. Test case 1, passive inflation and active contraction of a ventricle. Computational data related to POD-Galerkin-DEIM with $(N, m) = (16, 303)$ and $(N, m) = (39, 767)$.

ε_{POD}	10^{-3} ($N = 16$)	10^{-4} ($N = 39$)
ε_{DEIM}	$5 \cdot 10^{-4}$ ($m = 303$)	10^{-5} ($m = 767$)
Computational speed-up	$\times 6.2$	$\times 2.9$
Avg. CPU time	58 s	124 s
Time-avg. $L^2(\Omega_0)$ -absolute error	$1 \cdot 10^{-3}$	$2 \cdot 10^{-4}$
Time-avg. $L^2(\Omega_0)$ -relative error	$7 \cdot 10^{-3}$	$1 \cdot 10^{-3}$

Indeed, when a larger reduced basis is considered, such as in the case of $N = 39$ basis functions obtained for $\varepsilon_{POD} = 10^{-4}$ in the case of a FOM with dimension $N_h = 19365$, a remarkable improvement

in accuracy is achieved, as shown in Figure 12 and Table 2. However, a higher number of DEIM magic points is required for the solution of the reduced nonlinear system, ultimately doubling the online CPU time required to solve the hyper-ROM for each parameter instance.

As a matter of fact, the residual basis dimension m is highly influenced by the size N of the reduced subspace for the solution, so that a larger basis \mathbf{V} requires a greater number of interpolation points to correctly approximate the reduced nonlinear operators, thus reducing the overall speed-up of the ROM. Therefore, choosing $N = 16$ represents a good trade off between accuracy and efficiency. The FOM and the POD-Galerkin-DEIM ROM solutions computed using $m = 303$ at different time instants are shown in Figures 13 and 14 for two parameter values, together with their point-wise difference. We observe a good agreement between the high-fidelity and the reduced dynamics, with the greatest pointwise approximation error located near the apex, that is where the greatest displacement is experienced. About 6 minutes are required in average to solve the FOM for each instance of the parameter, while the solution of the hyper-ROM requires less than a minute, thus achieving a speed-up equal to $\times 6.2$.

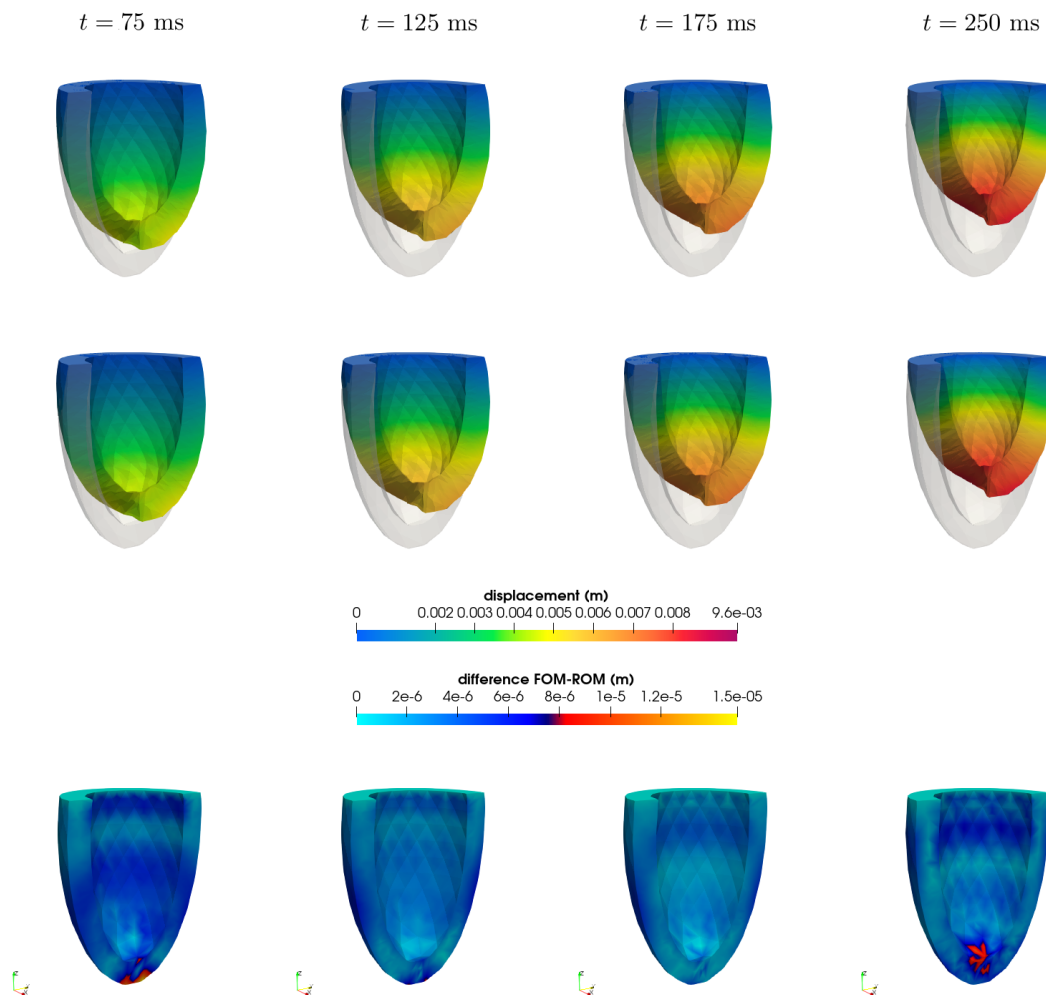


Figure 13. Test case 1, passive inflation and active contraction of a ventricle. FOM (top) and POD-Galerkin-DEIM (middle) displacements computed at different times for $\boldsymbol{\mu} = [61942.5, -77.5225, 87.9075]$ and difference between them (bottom).

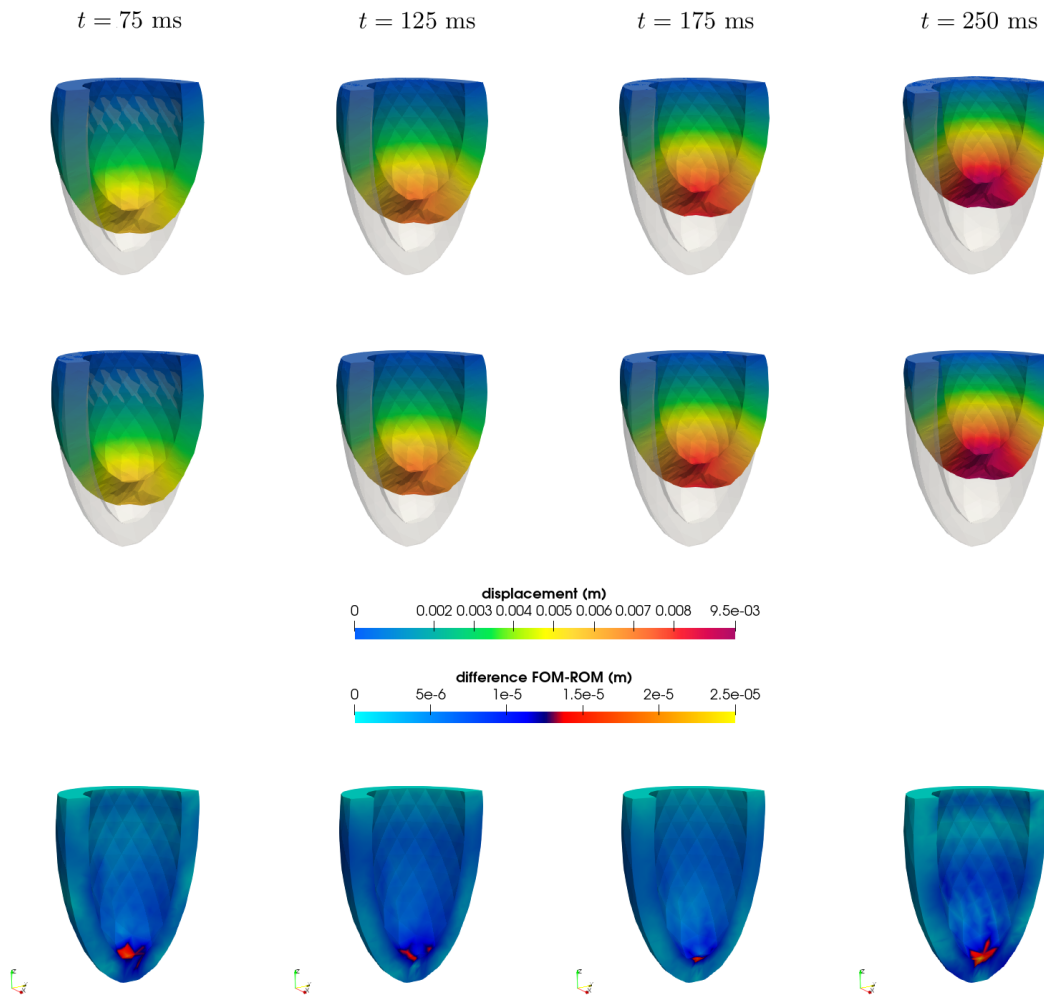


Figure 14. Test case 1, passive inflation and active contraction of a ventricle. FOM (top) and POD-Galerkin-DEIM (middle) displacements computed at different times for $\mu = [59737.5, -102.3225, 91.1625]$ and difference between them (bottom).

At last, we consider a higher dimensional parameter space, taking $P = 12$ input parameters,

$$\mu = [b_f, b_s, b_n, b_{fs}, b_{fn}, b_{sn}, K, C, \tilde{T}_a, \tilde{p}, \alpha^{epi}, \alpha^{endo}] \in \mathcal{P} \subset \mathbb{R}^{12},$$

as follows:

- the material stiffness in different directions $b_f \in [6.6, 9.4]$, $b_s, b_n, b_{sn} \in [1.65, 2.35]$, $b_{fs}, b_{fn} \in [3.3, 4.7]$,
- the bulk modulus $K \in [4 \cdot 10^4, 6 \cdot 10^4]$ Pa,
- the multiplicative constant $C \in [1.5 \cdot 10^3, 2.5 \cdot 10^3]$ Pa,
- the maximum active tension $\tilde{T}_a \in [49.5 \cdot 10^3, 70.5 \cdot 10^3]$ Pa,
- the fiber angles $\alpha^{epi} \in [-105.5, -74.5]^\circ$ and $\alpha^{endo} \in [74.5, 105.5]^\circ$,
- the steepness of the pressure ramp $\tilde{p} \in [14 \cdot 10^3, 16 \cdot 10^3]$ Pa.

We consider $n_s = 50$ and $n'_s = 75$ training parameters sampled in the parameter space \mathcal{P} through LHS for the construction of the FOM solution and the ROM residual snapshots matrices, respectively. Further details are summarized in Table 3.

Table 3. Test case 1, passive inflation and active contraction of a ventricle, case $P = \text{size}(\boldsymbol{\mu}) = 12$. Computational data related to POD-Galerkin-DEIM with $N = 19$ and $m = 574$.

\mathcal{E}_{POD}	$1e-3$ ($N = 19$)
\mathcal{E}_{DEIM}	$1e-4$ ($m = 574$)
Computational speed-up	$\times 4.1$
Avg. CPU time	87 s
Time-avg. $L^2(\Omega_0)$ -absolute error	$3 \cdot 10^{-3}$
Time-avg. $L^2(\Omega_0)$ -relative error	$2 \cdot 10^{-3}$

Despite increasing the number of unknown parameters from 3 to 12, the hyper-ROM dimension N is almost unaffected by the choice of a higher dimensional parameter space, as the mechanical behavior during systole is mostly influenced by the active stress and the fiber orientation. To better see this, we perform a sensitivity analysis in order to quantify the effects of parameter variation on some quantities of interest, that are outputs $\mathbf{y}(\boldsymbol{\mu})$ associated with the PDE solution $\mathbf{u}(\mathbf{X}, t; \boldsymbol{\mu})$.

A variance-based measure of sensitivity can be obtained, e.g., through the Sobol' indices. These latter can be computed by evaluating the underlying model with multiple parameter values and analyzing the statistical properties of the associated input-output samples. In particular, we define

$$S_i = \frac{\text{Var}_{\boldsymbol{\mu}^{(i)}}(\mathbb{E}_{\boldsymbol{\mu}_{\sim i}}[\mathbf{y}(\boldsymbol{\mu}) \mid \boldsymbol{\mu}^{(i)}])}{\text{Var}(\mathbf{y}(\boldsymbol{\mu}))} \quad (5.1)$$

$$S_{T_i} = 1 - \frac{\text{Var}_{\boldsymbol{\mu}_{\sim i}}(\mathbb{E}_{\boldsymbol{\mu}^{(i)}}[\mathbf{y}(\boldsymbol{\mu}) \mid \boldsymbol{\mu}_{\sim i}])}{\text{Var}(\mathbf{y}(\boldsymbol{\mu}))} \quad (5.2)$$

for $i = 1, \dots, P$, where the *main effect* indices (5.1) represent the main contribution of each input factor alone to the variance of the output, whereas the *total effect* indices (5.2) include also the interactions with the other parameters. Above, $\mathbb{E}_{\boldsymbol{\mu}_{\sim i}}[\cdot]$ and $\text{Var}_{\boldsymbol{\mu}_{\sim i}}(\cdot)$ denote the conditional expectation and variance, respectively, taken over $\boldsymbol{\mu}_{\sim i}$ (all parameter components but $\boldsymbol{\mu}^{(i)}$); similarly, $\mathbb{E}_{\boldsymbol{\mu}^{(i)}}[\cdot]$ and $\text{Var}_{\boldsymbol{\mu}^{(i)}}(\cdot)$ denote the conditional expectation and variance, respectively, taken over $\boldsymbol{\mu}^{(i)}$.

In our case, we consider as quantities of interest the displacement of the apex \mathbf{A}_0 in the direction perpendicular to the basal plane and the displacement of a point \mathbf{P}_0 located between the apex and the base on the epicardium in the direction perpendicular to the basal plane, both evaluated every 10 time steps (see Figure 15).

We consider $N_{base} = 50$ base samples for the Saltelli method [54], corresponding to $N_{base}(P + 2) = 700$ input-output evaluations, and rely on the POD-Galerkin-DEIM ROM with $N = 19$ and $m = 574$ (see Table 3) to speed-up the computations. The average over time of the sensitivity indices related to the apex \mathbf{A}_0 and point \mathbf{P}_0 displacements are reported in Figure 16. From these results, we can conclude that the most influential parameters are the maximum value \tilde{T}_a of the active tension and the parameter α^{epi} related to the fiber orientation at the epicardium.

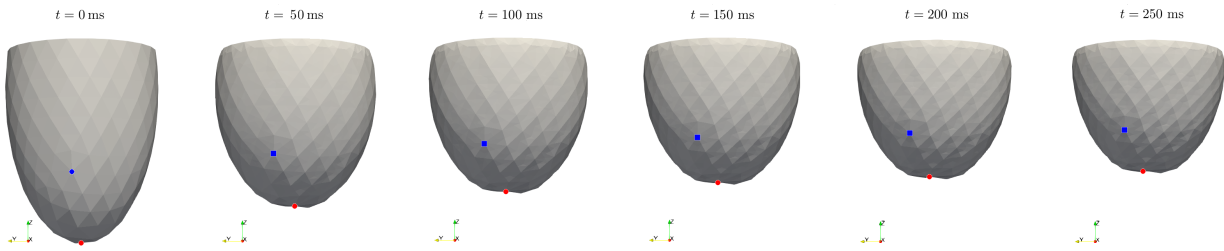


Figure 15. Points \mathbf{A}_0 (red circle) and \mathbf{P}_0 (blue square) at different times, for a given $\boldsymbol{\mu}$ value.

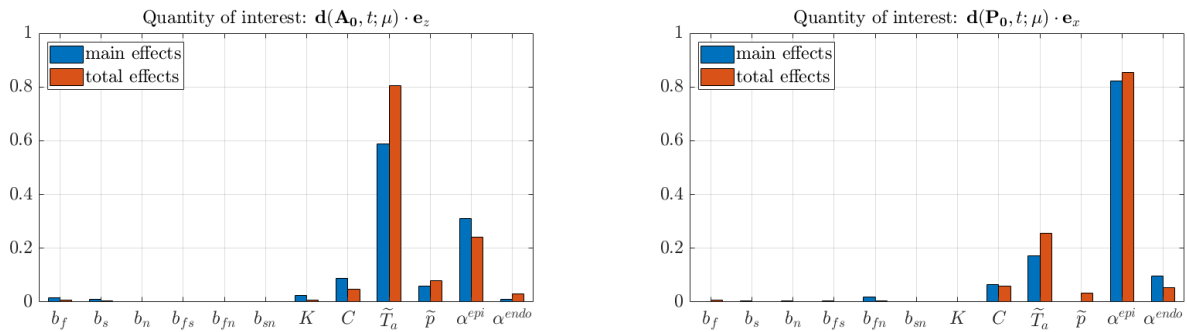


Figure 16. Passive inflation and active contraction of a left ventricle. Average sensitivity indices for \mathbf{A}_0 (left) and \mathbf{P}_0 (right), computed using POD-Galerkin-DEIM ROM with $N = 19$ and $m = 574$.

5.2. Idealized full cycle of the left ventricle with a patient specific geometry

Finally, we want to analyze the performances of the POD-Galerkin-DEIM method when the whole cardiac cycle is taken into account. For the sake of simplicity, we employ suitable analytical time-dependent functions to represent the influence of the active stress model and blood circulation. In particular, the active tension $T_a(t; \boldsymbol{\mu})$ and the pressure $\mathbf{g}(t; \boldsymbol{\mu})$ are computed as follows:

- 1) for a fixed set of physiological parameters, we solve the cardiac electromechanics (EM) problem coupled with a lumped-parameter model for hemodynamics, as done in [55].
- 2) by performing a cubic spline interpolation of the space-averaged active tension $T_a^{EM}(t)$ and of the pressure $\mathbf{g}^{EM}(t)$ coming from the EM simulation, we obtain the corresponding analytical surrogate functions $T_a^M(t)$ and $\mathbf{g}^M(t)$, reported in Figure 17.
- 3) to take into account a parameter-dependence, we define

$$T_a(t; \boldsymbol{\mu}) = \frac{\tilde{T}_a}{\max_{t \in [0, T]} T_a^M(t)} T_a^M(t), \quad \mathbf{g}(t; \boldsymbol{\mu}) = \frac{\tilde{p}}{\max_{t \in [0, T]} \mathbf{g}^M(t)} \mathbf{g}^M(t),$$

in order to model different maximum active tensions and loading conditions.

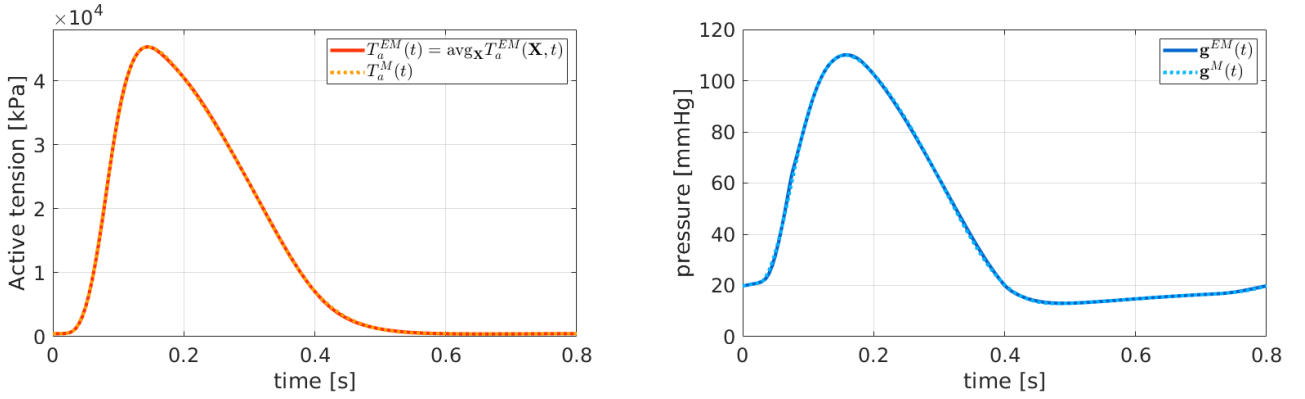


Figure 17. Test case 2, idealized full cycle of the left ventricle with a patient specific geometry. Space-averaged active tension (left) and blood pressure (right) computed during a EM simulation and the corresponding fitted curves.

The reference geometry $\Omega_0 \subset \mathbb{R}^3$ is a patient-specific left ventricle, pre-processed from the Zygote Solid 3D heart model [56] reconstructed from an high resolution computed tomography scan, whose associated hexahedral mesh with 6167 vertices is reported in Figure 18. The FOM is built using \mathbb{Q}_1 -FE, such that the high-fidelity dimension is $N_h = 18501$. Differently from all other test cases considered so far, we assume the following boundary conditions, according to [57]:

- Robin boundary conditions at the epicardium Γ_0^{epi} (or Γ_0^R),

$$\mathbf{P}(\mathbf{u})\mathbf{N} = -(\mathbf{N} \otimes \mathbf{N})(K_{\perp}\mathbf{u} + C_{\perp}\partial_t\mathbf{u}) - (\mathbf{I} - \mathbf{N} \otimes \mathbf{N})(K_{\parallel}\mathbf{u} + C_{\parallel}\partial_t\mathbf{u}),$$

with $K_{\perp} = 2 \cdot 10^5$, $K_{\parallel} = C_{\perp} = 2 \cdot 10^4$ and $C_{\parallel} = 2 \cdot 10^3$;

- Neumann boundary conditions at the endocardium Γ_0^{endo} (or Γ_0^N),

$$\mathbf{P}(\mathbf{u})\mathbf{N} = -\mathbf{g}(t; \boldsymbol{\mu})\mathbf{J}\mathbf{F}^{-T}\mathbf{N};$$

- energy-consistent boundary conditions [58] at the base Γ_0^{base} ,

$$\mathbf{P}(\mathbf{u})\mathbf{N} = -\frac{\|\mathbf{J}\mathbf{F}^{-T}\mathbf{N}\|}{\int_{\Gamma_0^{base}}\|\mathbf{J}\mathbf{F}^{-T}\mathbf{N}\|d\Gamma} \int_{\Gamma_0^{endo}} \mathbf{P}(\mathbf{u})\mathbf{N}d\Gamma.$$

This choice of boundary conditions do not affect the construction of the ROM, since no assumptions about the form of the FOM are made in the reduction strategy. As for the previous test cases, we consider the nearly-incompressible Guccione law and adopt the active stress approach. Regarding the fiber distribution, we employ the rule-based method proposed in [49], depending on parameter angles $\boldsymbol{\alpha}^{epi}$, $\boldsymbol{\alpha}^{endo}$, $\boldsymbol{\beta}^{epi}$ and $\boldsymbol{\beta}^{endo}$. For time discretization, we consider a uniform time step $\Delta t = 5 \cdot 10^{-3}$ s and set $T = 0.8$ s, corresponding to the duration of a single heartbeat, resulting in a total number of 160 time iterations. We choose, as unknown parameters, those most affecting cardiac deformation during systole and diastole, that are,

- the multiplicative factor of the constitutive law $C \in [0.44 \cdot 10^3, 1.32 \cdot 10^3]$ Pa;
- the active tension parameter $\widetilde{T}_a \in [49.5 \cdot 10^3, 70.5 \cdot 10^3]$ Pa;

- the fiber angle at the epicardium $\alpha^{epi} \in [-75, -45]^\circ$.

The active tension and the fiber angle influence the contraction and the twisting motion typical of the ventricular systole, whereas C is related to the stiffness of the tissue. All the other parameters are fixed to their reference values, namely $b_f = 8$, $b_s = 6$, $b_n = b_{fn} = b_{sn} = 3$, $b_{fn} = 12$, $K = 50 \cdot 10^3$ Pa, $\bar{p} = 15 \cdot 10^3$ Pa, $\alpha^{endo} = 60^\circ$, $\beta^{epi} = 20^\circ$ and $\beta^{endo} = -20^\circ$. Computing the high-fidelity solution for each instance of the parameter required almost 15 minutes.

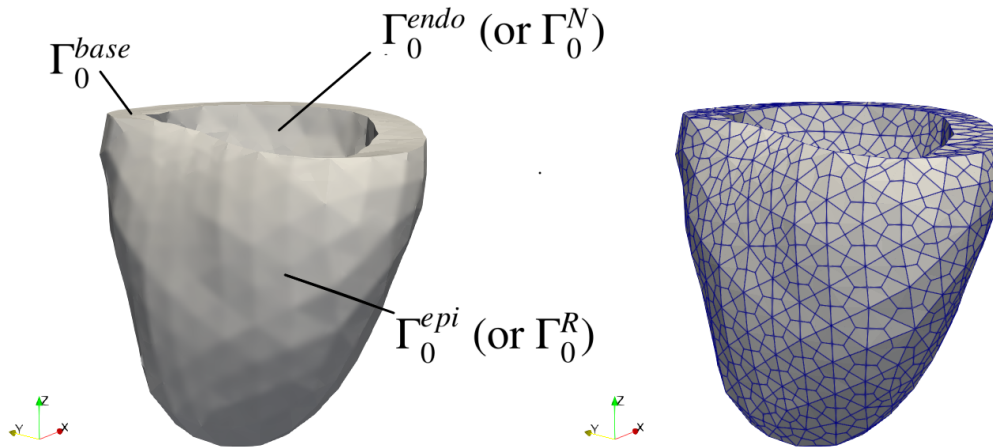


Figure 18. Test case 2, idealized full cycle of the left ventricle with a patient specific geometry. Unloaded geometry and computational grid.

Remark 5. *Due to the fact that we are neglecting the coupling between mechanics and circulation, non-physiological pressure-volume loops are obtained, especially in the first time steps of the heartbeat, corresponding to ventricular systole, which lacks of the isometric contraction phase. In addition to this, the end-diastolic configuration of the left ventricle should be recovered before starting the mechanical simulation. However, we point out that our aim is to test and analyze the reduction methodology on parameterized time-dependent nonlinear PDEs. Moreover, the simulated cardiac cycle can be regarded as a sufficiently accurate reproduction of both systole and diastole deformations for the purpose at hand.*

For the construction of the reduced basis \mathbf{V} , we collect $n_s = 20$ FOM solutions and perform POD with $\varepsilon_{POD} = 10^{-3}$, obtaining a reduced subspace of dimension $N = 28$. To build the residual basis, we perform $n'_s = 50$ ROM simulations and collect the residual snapshots. Since we are using $N_t = 160$ and $n'_s = 50$, and at least two Newton iterations are performed at each time step, we end up with a residual snapshots matrix \mathbf{S}_R of more than 16000 columns. For this reason, we rely on randomized-SVD to speed-up the computation of Φ_R , by choosing a priori the number of DEIM basis, rather than the POD tolerance ε_{DEIM} (see Algorithm 4). Table 4 summarizes the average errors computed over a testing set of 20 random parameters and the CPU times obtained using $m = 850$, 1000 and 1200, while volume and pressure-volume loop for $m = 850$ are reported in Figure 19. Here, POD-Galerkin-DEIM outputs are compared to those of the FOM for two different values of the parameters vector. We point out that no convergence of the reduced Newton system has been obtained for all testing parameters when using smaller residual basis.

Table 4. Test case 2, idealized full cycle of the left ventricle with a patient specific geometry. Computational data related to POD-Galerkin-DEIM with $N = 28$ and different values of m .

	$m = 850$	$m = 1000$	$m = 1200$
Computational speed-up	$\times 3.0$	$\times 2.8$	$\times 2.7$
Avg. CPU time	285 s	303 s	311 s
Time-avg. $L^2(\Omega_0)$ -absolute error	$1 \cdot 10^{-3}$	$1 \cdot 10^{-3}$	$1 \cdot 10^{-3}$
Time-avg. $L^2(\Omega_0)$ -relative error	$8 \cdot 10^{-3}$	$8 \cdot 10^{-3}$	$8 \cdot 10^{-3}$

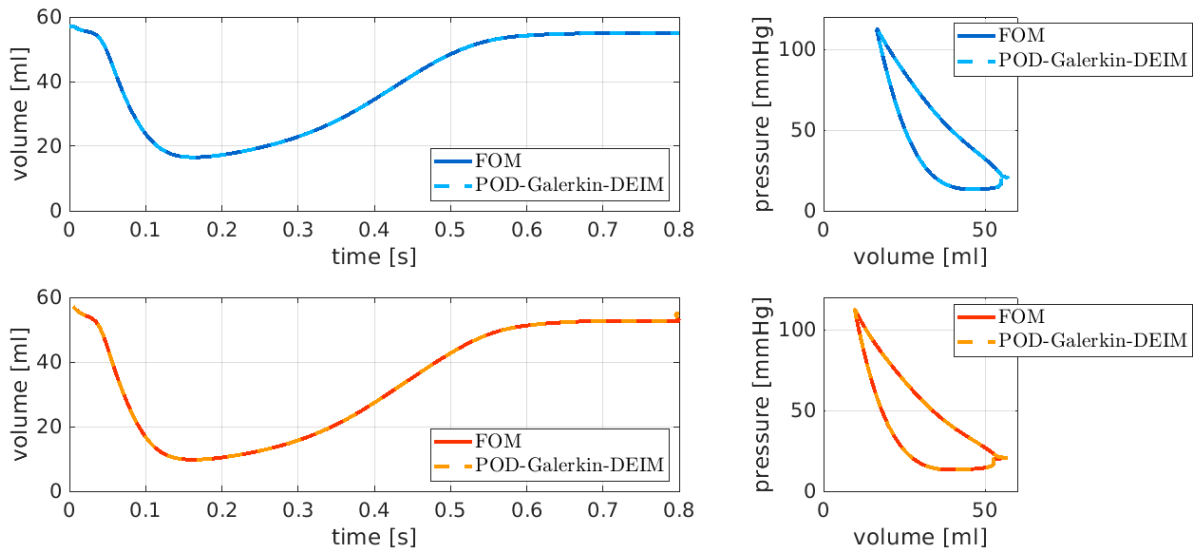


Figure 19. Test case 2, idealized full cycle of the left ventricle with a patient specific geometry. Ventricular cavity volume (left) and pressure-volume loop (right) for parameter values $\mu_1 = [1.078, 50.025, -66.75]$ (top) and $\mu_2 = [0.682, 69.975, -54.75]$ (bottom).

In Figure 20 we finally report the POD-Galerkin-DEIM solutions at four time instants, for two different values of the parameter vector $\mu = [C, \tilde{T}_a, \alpha^{epi}]$, respectively. Moreover, the corresponding pointwise difference between the high-fidelity solution and its reduced-order approximation is also reported, showing that the error does not increase over time.

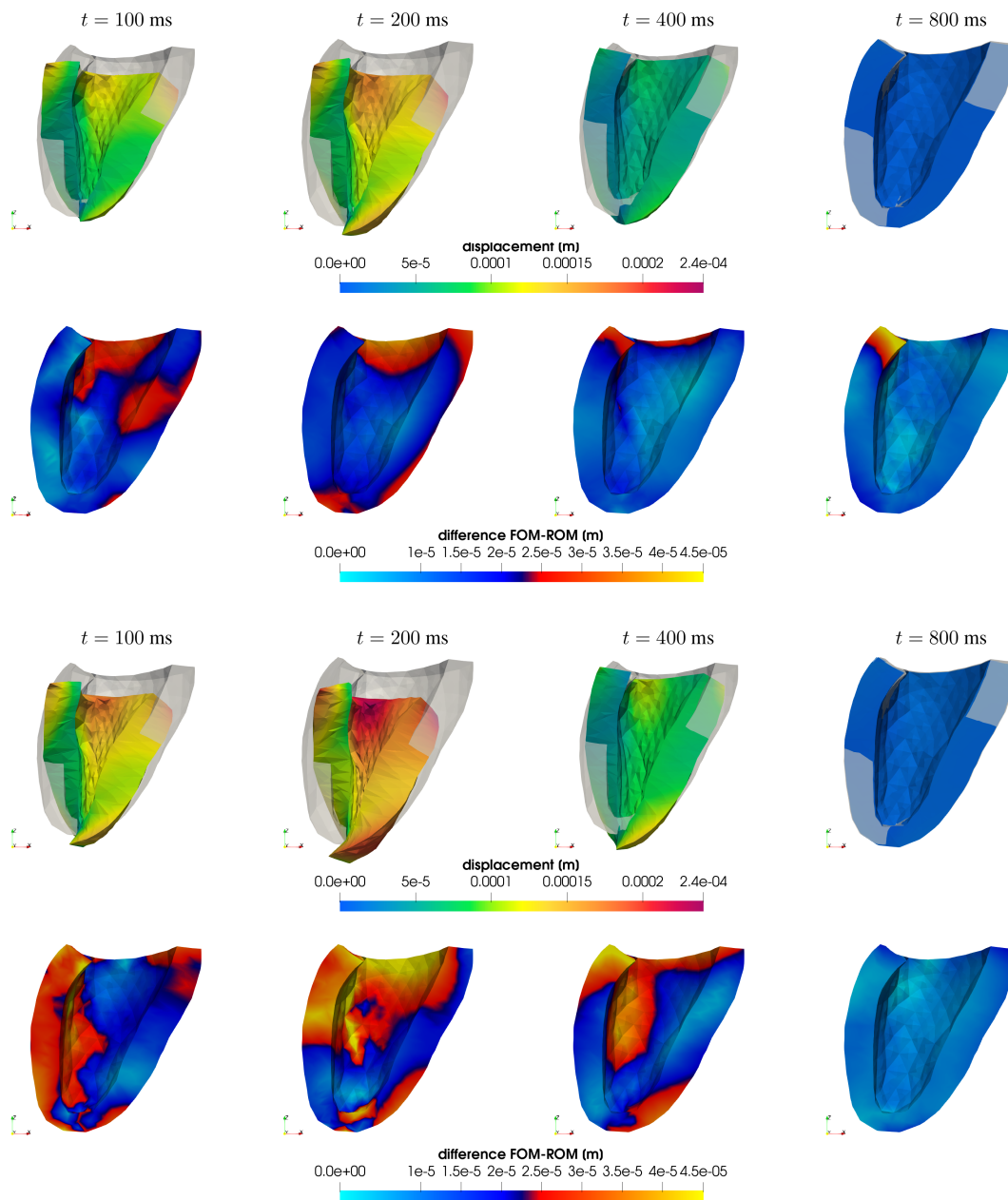


Figure 20. Test case 2, idealized full cycle of the left ventricle with a patient specific geometry. Lines 1–2: POD-Galerkin-DEIM solution (top) and error (bottom) computed at different times for $\mu = [1.078, 50.025, -66.75]$; lines 3–4: POD-Galerkin-DEIM solution (top) and error (bottom) computed at different times for $\mu = [0.682, 69.975, -54.75]$.

6. Conclusions

In this paper we have introduced and investigated a projection-based reduced order modeling framework to deal with problems arising in cardiac mechanics, which present great challenges from a numerical standpoint due to the complex material behavior. The POD-Galerkin ROM, equipped

with DEIM hyper-reduction, has enabled to accurately approximate the high-fidelity solution of parameterized nonlinear, time-dependent problems in elastodynamics. Examples have been shown related to test cases reproducing different phases of the cardiac cycle on both idealized and realistic left ventricle geometries. The hyper-ROMs proposed rely on (i) POD for the construction of a low-dimensional trial subspace to represent the problem solution, (ii) a Galerkin projection to generate, in a physically consistent way, the reduced order model, and (iii) suitable hyper-reduction techniques such as the discrete empirical interpolation method to enhance the assembling of nonlinear terms (with respect to the solution, to input parameters, or to both). In the considered test cases, POD provides low-dimensional subspace, still preserving a sufficient fidelity: despite their highly nonlinear nature, elastodynamics problems can be effectively reduced by exploiting projection-based strategies, with POD-Galerkin ROMs showing very good accuracy even in presence of a handful of basis functions. In this respect, mechanical problems do not pose the same issue as transport, wave, or convection-dominated phenomena, such as those related with cardiac electrophysiology, where coherent structures propagate over time and the RB method may yield inefficient ROMs, necessitating a high number of basis functions.

Performing hyper-reduction by means of the DEIM technique allows to achieve good results in terms of computational speed-ups of the ROMs with respect to the FOM without affecting the overall approximation error. In particular, since only a handful of basis functions is required for solution-space reduction, we expect even higher gains when finer computational meshes are used. Furthermore, we point out that relying on randomized-SVD allows to deal with massive datasets of nonlinear arrays which would be otherwise difficult to handle with deterministic techniques, such as POD.

A serious issue is instead represented by the assembling of reduced operators in this framework, and the projection of the approximated operators through DEIM onto the RB space. In fact, even if the nonlinear quantities are assembled onto a reduced mesh, a large residual basis may be needed to ensure the convergence of Newton method for complex applications, overall compromising the ROM efficiency, as most of the online CPU time is required for assembling the reduced residual vector. In addition to the fact that hyper-reduction aims at approximating vanishing terms with a global basis, the highly nonlinear nature of the constitutive laws makes the residual terms orders of magnitude more expensive to be approximated than the parameter-to-solution map, regarding both the number of basis functions involved in their expansions, and the computational time required at the offline and the online stages.

This observation suggests the idea of relying on surrogate models to perform operator approximation, overcoming the need to assemble the nonlinear terms onto the computational mesh. Pursuing this strategy, will allow us to rely on physics-based (thus, consistent) ROMs built through a POD-Galerkin RB method, however avoiding the computational burden entailed by classical hyper-reduction strategies. The analysis and the application of deep learning-based methods to perform hyper-reduction will make the object of forthcoming publications.

Author contributions

- Conceptualization: all authors.
- Formal analysis: Ludovica Cicci.
- Funding acquisition: Alfio Quarteroni.

- Investigation: Ludovica Cicci.
- Methodology: Ludovica Cicci, Stefano Pagani, Andrea Manzoni.
- Software: Ludovica Cicci, Stefano Pagani.
- Supervision: Stefania Fresca, Stefano Pagani, Andrea Manzoni, Alfio Quarteroni.
- Validation: Ludovica Cicci.
- Visualization: Ludovica Cicci.
- Writing - original draft: Ludovica Cicci, Andrea Manzoni.
- Writing - review & editing: Stefania Fresca, Stefano Pagani, Andrea Manzoni, Alfio Quarteroni.

Acknowledgments

The authors have been supported by the ERC Advanced Grant iHEART, “An integrated heart model for the simulation of the cardiac function”, 2017-2022, P. I. A. Quarteroni (ERC2016AdG, project ID: 740132).

Conflict of interest

The authors declare no conflict of interest.

References

1. A. Quarteroni, A. Manzoni, C. Vergara, The cardiovascular system: mathematical modelling, numerical algorithms and clinical applications, *Acta Numer.*, **26** (2017), 365–590. <http://doi.org/10.1017/S0962492917000046>
2. A. Quarteroni, L. Dede', A. Manzoni, C. Vergara, *Mathematical modelling of the human cardiovascular system. data, numerical approximation, clinical applications*, Cambridge: Cambridge University Press, 2019. <https://doi.org/10.1017/9781108616096>
3. P. Hauseux, J. Hale, S. Cotin, S. P. A. Bordas, Quantifying the uncertainty in a hyperelastic soft tissue model with stochastic parameters, *Appl. Math. Model.*, **62** (2018), 86–102. <https://doi.org/10.1016/j.apm.2018.04.021>
4. J. Campos, J. Sundnes, R. Dos Santos, B. M. Rocha, Uncertainty quantification and sensitivity analysis of left ventricular function during the full cardiac cycle, *Phil. Trans. R. Soc. A*, **378** (2020), 20190381. <https://doi.org/10.1098/rsta.2019.0381>
5. L. Marx, M. A. F. Gsell, A. Rund, F. Caforio, A. J. Prassl, G. Toth-Gayor, et al., Personalization of electro-mechanical models of the pressure-overloaded left ventricle: fitting of windkessel-type afterload models, *Phil. Trans. R. Soc. A*, **378** (2020), 20190342. <https://doi.org/10.1098/rsta.2019.0342>
6. R. Rodríguez-Cantano, J. Sundnes, M. Rognes, Uncertainty in cardiac myofiber orientation and stiffnesses dominate the variability of left ventricle deformation response, *Int. J. Numer. Meth. Biomed. Eng.*, **35** (2020), e3178. <https://doi.org/10.1002/cnm.3178>
7. S. Pagani, A. Manzoni, Enabling forward uncertainty quantification and sensitivity analysis in cardiac electrophysiology by reduced order modeling and machine learning, *Int. J. Numer. Meth. Biomed. Eng.*, **37** (2021), e3450. <https://doi.org/10.1002/cnm.3450>

8. A. Quarteroni, A. Manzoni, F. Negri, *Reduced basis methods for partial differential equations: an Introduction*, Cham: Springer, 2016. <https://doi.org/10.1007/978-3-319-15431-2>
9. P. Benner, S. Gugercin, K. Willcox, A survey of projection-based model reduction methods for parametric dynamical systems, *SIAM Rev.*, **57** (2015), 483–531. <https://doi.org/10.1137/130932715>
10. M. Morales, M. van den Boomen, C. Nguyen, J. Kalpathy-Cramer, B. R. Rosen, C. M. Stultz, et al., Deepstrain: A deep learning workflow for the automated characterization of cardiac mechanics, *Front. Cardiovasc. Med.*, **8** (2021), 1041. <https://doi.org/10.3389/fcvm.2021.730316>
11. P. Di Achille, A. Harouni, S. Khamzin, O. Solovyova, J. J. Rice, V. Gurev, Gaussian process regressions for inverse problems and parameter searches in models of ventricular mechanics, *Front. Physiol.*, **9** (2018), 1002. <https://doi.org/10.3389/fphys.2018.01002>
12. U. Noè, A. Lazarus, H. Gao, V. Davies, B. Macdonald, K. Mangion, et al., Gaussian process emulation to accelerate parameter estimation in a mechanical model of the left ventricle: a critical step towards clinical end-user relevance, *J. R. Soc. Interface*, **16** (2019), 20190114. <https://doi.org/10.1098/rsif.2019.0114>
13. S. Longobardi, A. Lewalle, S. Coveney, I. Sjaastad, E. K. S. Espe, W. E. Louch, et al., Predicting left ventricular contractile function via gaussian process emulation in aortic-banded rats, *Phil. Trans. R. Soc. A*, **378** (2019), 20190334. <https://doi.org/10.1098/rsta.2019.0334>
14. S. Buoso, T. Joyce, S. Kozerke, Personalising left-ventricular biophysical models of the heart using parametric physics-informed neural networks, *Med. Image Anal.*, **71** (2021), 102066. <https://doi.org/10.1016/j.media.2021.102066>
15. D. Bonomi, A. Manzoni, A. Quarteroni, A matrix DEIM technique for model reduction of nonlinear parametrized problems in cardiac mechanics, *Comput. Meth. Appl. Mech. Eng.*, **324** (2017), 300–326. <https://doi.org/10.1016/j.cma.2017.06.011>
16. A. Manzoni, D. Bonomi, A. Quarteroni, Reduced order modeling for cardiac electrophysiology and mechanics: New methodologies, challenges and perspectives, In: *Mathematical and numerical modeling of the cardiovascular system and applications*, Cham: Springer, 2018, 115–166. https://doi.org/10.1007/978-3-319-96649-6_6
17. D. Chapelle, A. Gariah, P. Moireau, J. Sainte-Marie, A Galerkin strategy with proper orthogonal decomposition for parameter-dependent problems – analysis, assessments and applications to parameter estimation, *ESAIM: M2AN*, **47** (2013), 1821–1843. <https://doi.org/10.1051/m2an/2013090>
18. M. Pfaller, M. Cruz Varona, J. Lang, C. Bertoglio, W. A. Wall, Using parametric model order reduction for inverse analysis of large nonlinear cardiac simulations, *Int. J. Numer. Meth. Biomed. Eng.*, **36** (2020), e3320. <https://doi.org/10.1002/cnm.3320>
19. J. M. Guccione, A. D. McCulloch, Finite element modeling of ventricular mechanics, In: *Theory of heart*, New York, NY: Springer, 1991, 121–144. https://doi.org/10.1007/978-1-4612-3118-9_6
20. J. Guccione, A. McCulloch, Mechanics of active contraction in cardiac muscle: Part I—Constitutive relations for fiber stress that describe deactivation, *J. Biomech. Eng.*, **115** (1993), 72–81. <http://doi.org/10.1115/1.2895473>

21. K. Costa, J. Holmes, A. McCulloch, Modelling cardiac mechanical properties in three dimensions, *Phil. Trans. R. Soc. A*, **359** (2001), 1233–1250. <https://doi.org/10.1098/rsta.2001.0828>
22. G. Holzapfel, R. Ogden, Constitutive modelling of passive myocardium: a structurally based framework for material characterization, *Phil. Trans. R. Soc. A*, **367** (2009), 3445–3475. <https://doi.org/10.1098/rsta.2009.0091>
23. D. Nordsletten, S. Niederer, M. Nash, P. J. Hunter, N. P. Smith, Coupling multi-physics models to cardiac mechanics, *Prog. Biophys. Mol. Biol.*, **104** (2011), 77–88. <https://doi.org/10.1016/j.pbiomolbio.2009.11.001>
24. H. M. Wang, H. Gao, X. Y. Luo, C. Berry, B. E. Griffith, R. W. Ogden, et al., Structure-based finite strain modelling of the human left ventricle in diastole, *Int. J. Numer. Meth. Biomed. Eng.*, **29** (2013), 83–103. <https://doi.org/10.1002/cnm.2497>
25. L. Barbarotta, S. Rossi, L. Dede', A. Quarteroni, A transmurally heterogeneous orthotropic activation model for ventricular contraction and its numerical validation, *Int. J. Numer. Meth. Biomed. Eng.*, **34** (2018), e3137. <https://doi.org/10.1002/cnm.3137>
26. L. Dede', A. Gerbi, A. Quarteroni, Segregated algorithms for the numerical simulation of cardiac electromechanics in the left human ventricle, In: *The mathematics of mechanobiology*, Cham: Springer, 2020, 81–116. https://doi.org/10.1007/978-3-030-45197-4_3
27. D. Lin, F. Yin, A multiaxial constitutive law for mammalian left ventricular myocardium in steady-state barium contracture or tetanus, *J. Biomech. Eng.*, **120** (1998), 504–517. <https://doi.org/10.1115/1.2798021>
28. D. Ambrosi, S. Pezzuto, Active stress vs. active strain in mechanobiology: constitutive issues, *J. Elast.*, **107** (2012), 199–212. <https://doi.org/10.1007/s10659-011-9351-4>
29. G. A. Holzapfel, *Nonlinear solid mechanics. A continuum approach for engineering*, Chichester: Wiley, 2001.
30. C. G. Broyden, A class of methods for solving nonlinear simultaneous equations, *Math. Comp.*, **19** (1965), 577–593. <https://doi.org/10.1090/S0025-5718-1965-0198670-6>
31. A. Radermacher, S. Reese, POD-based model reduction with empirical interpolation applied to nonlinear elasticity, *Int. J. Numer. Meth. Eng.*, **107** (2016), 477–495. <https://doi.org/10.1002/nme.5177>
32. J. Edmonds, Matroids and the greedy algorithm, *Mathematical Programming*, **1** (1971), 127–136. <https://doi.org/10.1007/BF01584082>
33. C. Farhat, S. Grimberg, A. Manzoni, A. Quarteroni, Computational bottlenecks for proms: precomputation and hyperreduction, In: *Model order reduction*, Berlin: De Gruyter, 2020, 181–244. <https://doi.org/10.1515/9783110671490-005>
34. M. Barrault, Y. Maday, N. Nguyen, A. T. Patera, An ‘empirical interpolation’ method: application to efficient reduced-basis discretization of partial differential equations, *C. R. Math.*, **339** (2004), 667–672. <https://doi.org/10.1016/j.crma.2004.08.006>
35. M. Grepl, Y. Maday, N. Nguyen, A. T. Patera, Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations, *ESAIM: M2AN*, **41** (2007), 575–605. <https://doi.org/10.1051/m2an:2007031>

36. Y. Maday, N. Nguyen, A. Patera, S. H. Pau, A general multipurpose interpolation procedure: The magic points, *Commun. Pure Appl. Anal.*, **8** (2008), 383–404. <https://doi.org/10.3934/cpaa.2009.8.383>
37. S. Chaturantabut, D. Sorensen, Nonlinear model reduction via discrete empirical interpolation, *SIAM J. Sci. Comput.*, **32** (2010), 2737–2764. <https://doi.org/10.1137/090766498>
38. D. Wirtz, D. Sorensen, B. Haasdonk. A posteriori error estimation for DEIM reduced nonlinear dynamical systems, *SIAM J. Sci. Comput.*, **36** (2014), A311–A338. <https://doi.org/10.1137/120899042>
39. K. Carlberg, R. Tuminaro, P. Boggs, Preserving lagrangian structure in nonlinear model reduction with application to structural dynamics, *SIAM J. Sci. Comput.*, **37** (2015), B153–B184. <https://doi.org/10.1137/140959602>
40. F. Negri, A. Manzoni, D. Amsallem, Efficient model reduction of parametrized systems by matrix discrete empirical interpolation, *J. Comput. Phys.*, **303** (2015), 431–454. <https://doi.org/10.1016/j.jcp.2015.09.046>
41. P. Astrid, S. Weiland, K. Willcox, T. Backx, Missing point estimation in models described by proper orthogonal decomposition, *IEEE Trans. Automat. Contr.*, **53** (2008), 2237–2251. <https://doi.org/10.1109/TAC.2008.2006102>
42. K. Carlberg, C. Bou-Mosleh, C. Farhat, Efficient non-linear model reduction via a least-squares petrov–galerkin projection and compressive tensor approximations, *Int. J. Numer. Meth. Eng.*, **86** (2011), 155–181. <https://doi.org/10.1002/nme.3050>
43. M. Drohmann, B. Haasdonk, M. Ohlberger, Reduced basis approximation for nonlinear parametrized evolution equations based on empirical operator interpolation, *SIAM J. Sci. Comput.*, **34** (2012), A937–A969. <https://doi.org/10.1137/10081157X>
44. D. Amsallem, M. Zahr, C. Farhat, Nonlinear model order reduction based on local reduced-order bases, *Int. J. Numer. Meth. Eng.*, **92** (2012), 891–916. <https://doi.org/10.1002/nme.4371>
45. P. Tiso, D. J. Rixen, Discrete empirical interpolation method for finite element structural dynamics, In: *Topics in nonlinear dynamics, volume 1*, New York, NY: Springer, 2013, 203–212. https://doi.org/10.1007/978-1-4614-6570-6_18
46. F. Ghavamian, P. Tiso, A. Simone, POD-DEIM model order reduction for strain-softening viscoplasticity, *Comput. Meth. Appl. Mech. Eng.*, **317** (2017), 458–479. <https://doi.org/10.1016/j.cma.2016.11.025>
47. S. Land, V. Gurev, S. Arens, C. M. Augustin, L. Baron, R. Blake, et al., Verification of cardiac mechanics software: benchmark problems and solutions for testing active and passive material behaviour, *Proc. R. Soc. A*, **471** (2015), 20150641. <https://doi.org/10.1098/rspa.2015.0641>
48. D. Arndt, W. Bangerth, B. Blais, T. C. Clevenger, M. Fehling, A. V. Grayver, et al., The deal.II library, version 9.2, *J. Numer. Math.*, **28** (2020), 131–146. <https://doi.org/10.1515/jnma-2020-0043>
49. J. Bayer, R. Blake, G. Plank, N. A. Trayanova, A novel rule-based algorithm for assigning myocardial fiber orientation to computational heart models, *Ann. Biomed. Eng.*, **40** (2012), 2243–2254. <https://doi.org/10.1007/s10439-012-0593-5>

50. J. Wong, E. Kuhl, Generating fibre orientation maps in human heart models using poisson interpolation, *Comput. Meth. Biomech. Biomed. Eng.*, **17** (2014), 1217–1226. <https://doi.org/10.1080/10255842.2012.739167>
51. R. Doste, D. Soto-Iglesias, G. Bernardino, A. Alcaine, R. Sebastian, S. Giffard-Roisin, et al., A rule-based method to model myocardial fiber orientation in cardiac biventricular geometries with outflow tracts, *Int. J. Numer. Meth. Biomed. Eng.*, **35** (2019), e3185. <http://doi.org/10.1002/cnm.3185>
52. R. Piersanti, P. Africa, M. Fedele, C. Vergara, L. Dedè, A. F. Corno, et al., Modeling cardiac muscle fibers in ventricular and atrial electrophysiology simulations, *Comput. Meth. Appl. Mech. Eng.*, **373** (2021), 113468. <https://doi.org/10.1016/j.cma.2020.113468>
53. S. Rossi, T. Lassila, R. Ruiz-Baier, A. Sequeira, A. Quarteroni, Thermodynamically consistent orthotropic activation model capturing ventricular systolic wall thickening in cardiac electromechanics, *Eur. J. Mech-A/Solids*, **48** (2014), 129–142. <https://doi.org/10.1016/j.euromechsol.2013.10.009>
54. A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, et al., *Global sensitivity analysis: the primer*, John Wiley & Sons, 2008. <https://doi.org/10.1002/9780470725184>
55. F. Regazzoni, L. Dede', A. Quarteroni, Machine learning of multiscale active force generation models for the efficient simulation of cardiac electromechanics, *Comput. Meth. Appl. Mech. Eng.*, **370** (2020), 113268. <https://doi.org/10.1016/j.cma.2020.113268>
56. Zygote solid 3d heart generation ii development report, Zygote Media Group Inc., 2014.
57. F. Regazzoni, M. Salvador, P. C. Africa, M. Fedele, L. Dedè, A. Quarteroni, A cardiac electromechanical model coupled with a lumped-parameter model for closed-loop blood circulation, *J. Comput. Phys.*, **457** (2022), 111083. <https://doi.org/10.1016/j.jcp.2022.111083>
58. F. Regazzoni, L. Dede', A. Quarteroni, Biophysically detailed mathematical models of multiscale cardiac active mechanics, *PLoS Comput. Biol.*, **16** (2020), e1008294. <https://doi.org/10.1371/journal.pcbi.1008294>
59. N. Halko, P. Martinsson, J. Tropp, Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions, *SIAM Rev.*, **53** (2011), 217–288. <https://doi.org/10.1137/090771806>

Appendix

A. Proper orthogonal decomposition

Let $s_h : \mathcal{P} \rightarrow V_h$ be a parameter-dependent function and $\mathbf{s}_h(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ the corresponding vector of dofs, where $N_h = \dim(V_h)$. For a given set of training parameters $\mathcal{S} = \{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_{n_s}\} \subset \mathcal{P}$, with $n_s < N_h$, we define the snapshots matrix

$$\mathbf{S} = [\mathbf{s}_1 \mid \dots \mid \mathbf{s}_{n_s}] \in \mathbb{R}^{N_h \times n_s},$$

where $\mathbf{s}_i = \mathbf{s}_h(\boldsymbol{\mu}_i)$, $i = 1, \dots, n_s$. Proper Orthogonal Decomposition (POD) aims at approximating the manifold $\mathcal{M}_s = \{s_h(\boldsymbol{\mu}) \in V_h : \boldsymbol{\mu} \in \mathcal{P}\}$ identified by the image of s_h with a low-dimensional linear

subspace, retaining as much as possible of the information gathered in the snapshots. To achieve this goal, the singular value decomposition (SVD) of \mathbf{S} ,

$$\mathbf{S} = \mathbf{U}\mathbf{\Sigma}\mathbf{Z}^T,$$

is computed, where $\mathbf{U} = [\boldsymbol{\xi}_1 | \dots | \boldsymbol{\xi}_{N_h}] \in \mathbb{R}^{N_h \times N_h}$ and $\mathbf{Z} = [\boldsymbol{\zeta}_1 | \dots | \boldsymbol{\zeta}_{n_s}] \in \mathbb{R}^{n_s \times n_s}$ are orthogonal matrices collecting column-wise the left and the right singular vectors, respectively. The diagonal matrix $\mathbf{\Sigma} = \text{diag}(\sigma_1, \dots, \sigma_r) \in \mathbb{R}^{N_h \times n_s}$ contains all singular values of \mathbf{S} , sorted in descending order $\sigma_1 \geq \dots \geq \sigma_r > 0$, where $r \leq \min(N_h, n_s)$ is the rank of \mathbf{S} , so that we can write

$$\mathbf{S} = \sum_{j=1}^r \sigma_j \boldsymbol{\xi}_j \boldsymbol{\zeta}_j^T.$$

The N -dimensional POD basis $\mathbf{V} = [\boldsymbol{\xi}_1 | \dots | \boldsymbol{\xi}_N]$ is obtained by collecting the first N columns of \mathbf{U} corresponding to the N largest singular values (see Algorithm 3). At the basis of POD, the Schmidt-Eckart-Young theorem (originally introduced for integral operators) provides a criterion for the best low-rank approximation of a given matrix of rank r . In fact, the basis $\mathbf{V} \in \mathbb{R}^{N_h \times N}$ is such that

$$\mathbf{V}\mathbf{V}^T\mathbf{S} = \sum_{j=1}^N \sigma_j$$

has rank $N \leq r$ and satisfies the following optimality property:

$$\|\mathbf{S} - \mathbf{V}\mathbf{V}^T\mathbf{S}\|_F^2 = \min_{\substack{\mathbf{A} \in \mathbb{R}^{N_h \times N_h} \\ \text{rank}(\mathbf{A}) \leq N}} \|\mathbf{S} - \mathbf{A}\|_F^2 = \sum_{j=N+1}^r \sigma_j^2, \quad (\text{A.1})$$

where $\|\cdot\|_F$ is the Frobenius norm. Furthermore, thanks to (A.1), the singular values of the snapshots matrix provide an heuristic criteria for choosing the basis dimension N , which can be computed as the minimum integer satisfying the condition

$$\frac{\sum_{i=1}^N \sigma_i^2}{\sum_{i=1}^r \sigma_i^2} \geq 1 - \varepsilon_{POD}^2, \quad (\text{A.2})$$

where $\varepsilon_{POD} > 0$ is a given tolerance.

Algorithm 3 Proper orthogonal decomposition (POD)

INPUT: $\mathbf{S} \in \mathbb{R}^{N_h \times n_s}$ OUTPUT: $\mathbf{V} \in \mathbb{R}^{N_h \times N}$

- 1: Perform SVD of \mathbf{S} , i.e., $\mathbf{S} = \mathbf{U}\mathbf{\Sigma}\mathbf{Z}^T$
 - 2: Select basis dimension N as the minimum integer fulfilling condition (A.2)
 - 3: Construct \mathbf{V} collecting the first N columns of \mathbf{U}
-

B. Randomized singular valued decomposition

Randomized Singular Valued Decomposition (randomized-SVD) provides a non-deterministic alternative to POD. Randomization offers, in fact, a powerful tool for performing low-rank matrix

approximation, especially when dealing with massive data sets. The randomized approach usually beats its classical competitors in terms of computational speed-up, accuracy and robustness [59]. The key idea of randomized SVD is to split the task of computing an approximated singular value decomposition of a given matrix into a first random stage, and a second deterministic one, see Algorithm 4. The former exploits random sampling to construct a low-dimensional subspace that captures most of the action of the input matrix; the latter is meant to restrict the given matrix to this subspace and then manipulate the associated reduced matrix with classical deterministic algorithms, to obtain the desired low-rank approximations.

Algorithm 4 Randomized-SVD

 INPUT: $\mathbf{S} \in \mathbb{R}^{m \times n}$, target rank $k \in \mathbb{N}$

 OUTPUT: $\mathbf{U}\mathbf{\Sigma}\mathbf{Z}^T \approx \mathbf{S}$
stage 1

- 1: Generate a Gaussian matrix $\mathbf{\Theta} \in \mathbb{R}^{N_h \times k}$
- 2: Compute $\mathbf{Q} \in \mathbb{R}^{N_h \times k}$ whose columns form an orthonormal basis for the range of $\mathbf{S}\mathbf{\Theta}$ and such that

$$\|\mathbf{S} - \mathbf{Q}\mathbf{Q}^T\mathbf{S}\|_2 \leq \min_{\text{rank}(\mathbf{X}) \leq k} \|\mathbf{S} - \mathbf{X}\|_2,$$

e.g., using the QR factorization.

stage 2

- 1: Form $\tilde{\mathbf{S}} = \mathbf{Q}^T\mathbf{S} \in \mathbb{R}^{k \times n}$
 - 2: Compute SVD of $\tilde{\mathbf{S}} = \tilde{\mathbf{U}}\mathbf{\Sigma}\tilde{\mathbf{Z}}^T$
 - 3: Set $\mathbf{U} = \mathbf{Q}\tilde{\mathbf{U}}$
-

This randomized approach is particularly convenient when the snapshots matrix is high-dimensional, i.e., when N_h and n_s are large. In fact, finding the first k dominant singular-values for a dense input matrix of dimension $N_h \times n_s$, requires $O(N_h n_s \log(k))$ floating-point operations for a randomized algorithm, in contrast with $O(N_h n_s k)$ flops for a classical one. Finally, Algorithm 4 can be adapted to solve the following problem: given a target error tolerance $\varepsilon > 0$, find $k = k(\varepsilon)$ and $\mathbf{Q} \in \mathbb{R}^{N_h \times k}$ satisfying

$$\|\mathbf{S} - \mathbf{Q}\mathbf{Q}^T\mathbf{S}\|_2 \leq \varepsilon.$$

C. Discrete empirical interpolation method (DEIM)

The DEIM algorithm for the approximation of a generic nonlinear function $\mathbf{f} : \mathcal{P} \rightarrow \mathbb{R}^{N_h}$, $\mathbf{f} = \mathbf{f}(\tau)$ (where $\tau = t$ and/or $\boldsymbol{\mu}$), as originally proposed in [37], is outlined as follows[†]:

- 1) construct a set of snapshots obtained by sampling $\mathbf{f}(\tau)$ at random values $\tau_1, \dots, \tau_{n_s}$ and apply POD to extract a basis from these snapshots, i.e.,

$$\mathbb{R}^{N_h \times m} \ni \boldsymbol{\Phi}_{\mathcal{F}} = \text{POD}([\mathbf{f}(\tau_1), \dots, \mathbf{f}(\tau_{n_s})], \varepsilon_{DEIM}),$$

where $\varepsilon_{DEIM} > 0$ is a given tolerance such that a condition similar to (A.2) holds;

[†]For simplicity, we consider vector representations of functions, assuming that all quantities have been discretized on a FE mesh.

- 2) iteratively select $m \ll N_h$ indices $\mathcal{I} \subset \{1, \dots, N_h\}$, corresponding to a subset rows of $\Phi_{\mathcal{F}}$, using a greedy procedure, which minimizes the interpolation error over the snapshots set;
- 3) given $\tau \notin \{\tau_1, \dots, \tau_{N_s}\}$, impose the interpolation conditions at the selected entries \mathcal{I}

$$\Phi_{\mathcal{F}|_{\mathcal{I}}} \mathbf{c}(\tau) = \mathbf{f}(\tau)|_{\mathcal{I}}, \quad (\text{C.1})$$

Here $\Phi_{\mathcal{F}|_{\mathcal{I}}} \in \mathbb{R}^{m \times m}$ is the matrix formed by the \mathcal{I} rows of $\Phi_{\mathcal{F}}$; as a result, we obtain

$$\mathbf{f}(\tau) \approx \mathbf{f}_m(\tau) := \Phi_{\mathcal{F}} \Phi_{\mathcal{F}|_{\mathcal{I}}}^{-1} \mathbf{f}(\tau)|_{\mathcal{I}}.$$

Algorithm 5 reports the greedy procedure to determine the DEIM interpolation points. At each step $k \in \{2, \dots, m\}$, the k -th magic point is selected as the quadrature point maximizing the (absolute value of the) difference between the basis vector ϕ_k and its current approximation $\Phi_{\mathcal{F}}(\cdot, 1:k-1)\psi$ relying on the $k-1$ computed magic points. Magic points are hierarchical and non-repeated by construction.

Algorithm 5 Discrete empirical interpolation method (DEIM)

INPUT: $\Phi_{\mathcal{F}} = [\phi_1, \dots, \phi_m] \in \mathbb{R}^{N_h \times m}$

OUTPUT: $\mathcal{I} = \{i_1, \dots, i_m\}$

- 1: Find $i_1 = \arg \max_i \{|\phi_1|_i\}_{i=1}^{N_h}$
 - 2: Set $\mathcal{I} = \{i_1\}$
 - 3: **for** $k = 2, \dots, m$ **do**
 - 4: Solve $\Phi_{\mathcal{F}}(\cdot, 1:k-1)|_{\mathcal{I}} \psi = \phi_k|_{\mathcal{I}}$
 - 5: Compute $\mathbf{r} = \phi_k - \Phi_{\mathcal{F}}(\cdot, 1:k-1)\psi$
 - 6: Find $i_k = \arg \max_i \{|\mathbf{r}|_i\}_{i=1}^{N_h}$
 - 7: Set $\mathcal{I} \leftarrow \mathcal{I} \cup \{i_k\}$
 - 8: **end for**
-

Condition (C.1) can be generalized to the case where more sample indices than basis function are considered, leading to a gappy POD reconstruction. The solution to the least-squares problem

$$\mathbf{c}(\tau) = \arg \min_{\mathbf{x} \in \mathbb{R}^m} \|\mathbf{f}(\tau)|_{\mathcal{I}} - \Phi_{\mathcal{F}|_{\mathcal{I}}} \mathbf{x}\|_2^2$$

would yield $\mathbf{f}_m(\tau) = \Phi_{\mathcal{F}} \Phi_{\mathcal{F}|_{\mathcal{I}}}^{\dagger} \mathbf{f}(\tau)|_{\mathcal{I}}$, where the Moore-Penrose inverse of a full column rank matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$ is $\mathbf{A}^{\dagger} := (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T$.



AIMS Press

© 2023 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)