



MOX-Report No. 05/2018

**Numerical approximation of parametrized problems in
cardiac electrophysiology by a local reduced basis
method**

Pagani, S.; Manzoni, A.; Quarteroni, A.

MOX, Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

mox-dmat@polimi.it

<http://mox.polimi.it>

Numerical approximation of parametrized problems in cardiac electrophysiology by a local reduced basis method

Stefano Pagani^a, Andrea Manzoni^a, Alfio Quarteroni^a

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

Abstract

The efficient solution of coupled PDEs/ODEs problems arising in cardiac electrophysiology is of key importance whenever interested to study the electrical behavior of the tissue for several instances of relevant physical and/or geometrical parameters. This poses significant challenges to reduced order modeling (ROM) techniques – such as the reduced basis method – traditionally employed when dealing with the repeated solution of parameter dependent differential equations. Indeed, the nonlinear nature of the problem, the presence of moving fronts in the solution, and the high sensitivity of this latter to parameter variations, make the application of standard ROM techniques very problematic. In this paper we propose a local ROM built through a k -means clustering in the state space of the snapshots for both the solution and the nonlinear term. Several comparisons among alternative local ROMs on a benchmark test case show the effectivity of the proposed approach. Finally, the application to a parametrized problem set on an idealized left-ventricle geometry shows the capability of the proposed ROM to face complex problems.

Keywords: cardiac electrophysiology; parametrized monodomain model; local reduced order model; reduced basis method; proper orthogonal decomposition; empirical interpolation method.

1. Introduction

The propagation of the electrical signal through the heart cells is the main responsible of their contraction mechanism, finally resulting in atrial and ventricular contractions. At the macroscopic level, the propagation of electrical potentials is described by means of partial differential equations (PDEs) suitably coupled with ordinary differential equations (ODEs); the latter describe the ionic currents in the cells, depending on a set of gating variables [1]. For instance, coupling the so-called monodomain model for the transmembrane potential $u = u(\mathbf{x}, t)$ with a phenomenological model for the ionic currents – involving a single gating variable $w = w(\mathbf{x}, t)$ – in a domain Ω representing, e.g., a portion of the myocardium (or the whole left ventricle) results in the following time-dependent nonlinear differential system

$$\begin{aligned} \frac{\partial u}{\partial t} - \operatorname{div}(\mathbf{D}\nabla u) + I_{ion}(u, w) &= I_{app}(\mathbf{x}; t), & \mathbf{x} \in \Omega, \quad t \in (0, T) \\ \frac{\partial w}{\partial t} + g(u, w) &= 0, & \mathbf{x} \in \Omega, \quad t \in (0, T) \\ \frac{\partial u}{\partial \mathbf{n}} &= 0, & \mathbf{x} \in \partial\Omega, \quad t \in (0, T) \\ u(\mathbf{x}, 0) = u_0, \quad w(\mathbf{x}, 0) &= w_0, & \mathbf{x} \in \Omega \end{aligned} \tag{1.1}$$

Email addresses: stefano.pagani@polimi.it (Stefano Pagani), andrea1.manzoni@polimi.it (Andrea Manzoni), alfio.quarteroni@polimi.it (Alfio Quarteroni)

where u represents the transmembrane potential for cardiac cells, w the recovery variable, and t a rescaled time; \mathbf{n} denotes the outward unit normal vector to the boundary $\partial\Omega$, whereas I_{app} is an applied current representing the initial activation of the tissue. $(1.1)_1$ is a nonlinear diffusion-reaction equation; the reaction term I_{ion} and the function g depend on both u and w , thus making the PDE and the ODE two-ways coupled. The most common choices for the two functions I_{ion} and g in order to efficiently reproduce the action-potential are, e.g., the FitzHugh-Nagumo [2, 3], the Aliev-Panfilov [4, 5] or the Mitchell and Schaeffer model [6]. The diffusivity tensor \mathbf{D} usually depends on the fibers-sheet structure of the tissue, affecting conduction velocities and directions.

When a full-order model (FOM) such as, e.g., the FE method, is used, the accurate solution of such a coupled system is computationally demanding [7, 8, 9, 10], because (i) strong constraints on the spatial mesh size have to be taken into account due to the propagation of very steep fronts, and (ii) very small time steps are required to capture the fast dynamics characterizing the propagation of the electrical signal. The same is true for the electrical potential of the atria [11]. The solution of the linear algebraic systems arising at each time step hinges upon the use of semi-implicit methods or operator splitting-based ones for the whole coupled PDEs/ODEs problem, requiring suitable preconditioning techniques [12, 13, 14, 15, 16].

1.1. The need of local reduced order models

Relying on full-order techniques is thus out of reach whenever a problem like (1.1) has to be solved many times, by varying parameter-dependent features affecting operators and/or data. These many-query problems can occur, for instance, when characterizing the evolution of the electrical potential for different tissue conductivity tensors \mathbf{D} , activation patterns I_{app} , physical coefficients in the expressions of I_{ion} and g or domains Ω possibly accounting for inter-subject variability. Long-term relevant goals include, among others, the solution of uncertainty quantification (UQ) problems [17, 18, 19, 20, 21] parameter estimation and inverse problems [22, 23, 24, 25].

Reduced basis (RB) methods and, more generally, reduced order models (ROMs) have been deeply investigated in the last decade and applied to a broad range of parameter dependent PDEs [26]. A possible approach to tackle nonlinear, time-dependent PDEs relies on:

1. proper orthogonal decomposition (POD) to generate a (unique, global) lower dimensional subspace in which the solution of the ROM problem is sought; the reduced basis is then provided by the first right singular vectors of a matrix \mathbf{S}_u collecting snapshots of the FOM obtained for different parameter values, at different time instants;
2. Galerkin or Petrov-Galerkin projection to generate the reduced-order arrays, whence the name of, e.g., *POD-Galerkin* ROM;
3. hyper-reduction techniques (such as the empirical interpolation (EIM) method, or its discrete counterpart (DEIM) [27, 28, 29, 30]) to speed up the evaluation of nonlinear and nonaffine¹ arrays, avoiding to access the FOM arrays and ensuring the overall ROM efficiency.

Such a *global ROM* strategy, however, might yield inefficient ROM approximations because of the unaffordable large sets of global basis functions required to approximate both the solution and the nonlinear terms. This happens, e.g., when the solutions manifold (that is, the set of all solutions of the FOM for varying parameters) is characterized by large parameter variations, different

¹A vector $\mathbf{f}(\boldsymbol{\mu})$ and a matrix $\mathbf{A}(\boldsymbol{\mu})$ depend affinely on $\boldsymbol{\mu}$ if they are expressed as $\mathbf{f}(\boldsymbol{\mu}) = \sum_{q=f}^{Q_f} \Theta_j^f(\boldsymbol{\mu}) \mathbf{f}_j$, $\mathbf{A}(\boldsymbol{\mu}) = \sum_{j=1}^{Q_a} \Theta_j^a(\boldsymbol{\mu}) \mathbf{A}_j$, for given $\boldsymbol{\mu}$ -dependent functions $\{\Theta_j^f\}_{j=1}^{Q_f}$, $\{\Theta_j^a\}_{j=1}^{Q_a}$, and $\boldsymbol{\mu}$ -independent vectors $\{\mathbf{f}_q\}_{q=1}^{Q_f}$ or matrices $\{\mathbf{A}_j\}_{j=1}^{Q_a}$; Q_f and Q_a are given integers, indicating the *parametric complexity* of \mathbf{f} and \mathbf{A} .

physical regimes, or moving features such as fronts or discontinuities. This is indeed the case of the monodomain problem (1.1), for which the transmembrane potential u is characterized by a traveling wave (depolarization) which progressively covers the domain representing the portion of cardiac tissue at hand. This mechanism, controlled by the ionic current, is then followed by a repolarization phase where the potential returns to its resting value.

In such a situation, the solution can be better approximated in a lower-dimensional subspace generated by local basis vectors, rather than in a unique subspace spanned by global basis vectors. To enhance computational efficiency in these contexts, several strategies have been proposed, relying on *local* reduced basis spaces (or *local ROMs*) possibly coupled with *adaptive* procedures for their construction or enrichment. Being able to switch between different local bases by evaluating quantities which only depend on the dimension of the local ROMs makes the whole procedure computationally attractive.

1.2. The content of this paper and comparison with other existing approaches

Despite the RB method has been applied to several classes of problems in the last decade, a reliable, efficient and accurate ROM for parametrized problems in cardiac electrophysiology like (1.1) is still lacking. In this paper we show how to construct local ROMs for such a problem by relying on different strategies to partition snapshot sets, assessing both the numerical accuracy and the computational efficiency of the proposed techniques. When depending on a set of parameters, problem (1.1) represents a new test-bed for ROM techniques, for several reasons. First of all, only few works have addressed this kind of problems thus far; moreover, only the case of few physical parameters has been considered, whereas our framework can also deal with geometrical parameters, related to real applications. Furthermore, often *ad-hoc* procedures have been devised to tackle problems in cardiac electrophysiology, without developing a *general purpose* framework able to account for several parameters. Last, but not least, the methodological framework for the problem at hand can be easily transferred to different time-dependent non-linear parametrized PDEs, depending on the application at hand.

A first application of POD to the bidomain equations for cardiac electrophysiology has been addressed in [24], limitedly to few parameters. In that paper POD is applied to compress snapshots over the time interval with the aim of reproducing *online* the dynamics of the system for new parameters slightly different than those sampled *offline*. The following issues represent further shortcomings in [24]: (i) no hyper-reduction technique has been employed, thus making the computational speed up carried by the ROM almost negligible; (ii) the ROM is consistent, as it is able to reproduce the solutions that have been used for its generation, however ROM accuracy is not guaranteed for other parameter values, and (iii) error convergence with respect to the ROM dimension has not been numerically assessed. A preliminary version of local ROM has been introduced in [31], however performing an *ad-hoc* partition of snapshots in the parameter space, without relying on a general clustering procedure. An alternative strategy, introduced in [32, 33], is based on the Lax-Pairs approach: here the basis functions are moved in time according to the traveling front. This approach entails additional *online* costs and, so far, is limited to two-dimensional problems.

Regarding error estimation, to the authors' best knowledge no a posteriori error bounds have been obtained for this class of problems. A priori error estimates for a POD-Galerkin ROM for the monodomain system have been instead derived in [34], whereas a stability analysis of a similar ROM strategy for the bidomain model has been recently addressed in [35]. In both cases, however, reduction has only been performed with respect to time, and no parameter dependences have been considered, thus circumventing the main difficulty inherent with the efficient handling of parameter-dependent operators and matrices.

The structure of the paper is as follows. In Sect. 2 we provide a formulation of the parametrized problems we focus on, as well as of its full-order approximation. A POD-Galerkin-DEIM model is detailed in Sect. 3, showing in particular how to enhance computational efficiency in the evaluation of nonlinear ionic terms. Section 4 is devoted to the construction of local ROMs by presenting different clustering procedures. Finally, in Sect. 5 we compare the numerical performances of the different local ROMs on two test cases in both two and three spatial dimensions, in terms of efficiency, accuracy and memory storage. A simple one-dimensional test case dealing with a parametrized FitzHugh-Nagumo model is proposed in the Appendix for the sake of providing a further comparison, in terms of error and convergence analysis, among the proposed techniques.

2. A parametrized monodomain model for cardiac electrophysiology

The bidomain equations [36, 16] characterize the behavior of the intra- and extra-cellular potentials $u_i = u_i(\mathbf{x}, t)$ and $u_e = u_e(\mathbf{x}, t)$, respectively, resulting in a time-dependent parabolic PDE coupled with an elliptic PDE. A suitable simplification of the bidomain model, which nevertheless enables to provide accurate solutions in the case there is no injection of current in the extracellular region, yields the so-called monodomain model (1.1) for the (dimensionless²) transmembrane potential $u = u_i - u_e$. The reaction term accounts for the ionic currents and depends on a vector of gating variables, which represent the percentage of open channels per unit area of the membrane. When a simple phenomenological ionic model is considered (e.g., FitzHugh-Nagumo or Aliev-Panfilov), the ionic current takes the form of a cubic nonlinear function of u and a single (dimensionless) gating variable plays the role of a recovery function, allowing to model refractoriness of cells disregarding sub-cellular processes. In this paper, we focus on the Aliev-Panfilov model, given by

$$\begin{aligned} I_{ion}(u, w) &= Ku(u - a)(u - 1) + wu, \\ g(u, w) &= \left(\varepsilon_0 + \frac{c_1 w}{c_2 + u} \right) (-w - Ku(u - b - 1)) \end{aligned} \quad (2.1)$$

where the parameters K , a , b , ε_0 , c_1 , c_2 are related to the cell. In particular, a represents an oscillation threshold. The additional weighting factor $\varepsilon_0 + \frac{c_1 w}{c_2 + u}$, not present in the original FitzHugh-Nagumo model, was introduced in [4] to tune the restitution curve to experimental observations by adjusting the parameters c_1 and c_2 . If other models of cellular bioelectrical activity were considered, based on either a single ODE (such as FitzHugh-Nagumo, Roger-McCulloch or Mitchell-Schaeffer models) or a system of ODEs (such as the Fenton-Karma model), the construction of ROM would not change; see, e.g., [37, 16, 1] for a detailed review.

2.1. Parametrized formulation

We first state the weak formulation of problem (1.1)–(2.1), which stands at the basis of the full-order approximation of the problem, obtained with the Galerkin-finite element (FE) method. Hereon we denote by $\boldsymbol{\mu} \in \mathcal{P} \subset \mathbb{R}^p$ a parameter vector listing the p input parameters of interest characterizing physical and/or geometrical properties; \mathcal{P} is a subset of \mathbb{R}^p , denoting the parameter space. For $t > 0$, the weak formulation of problem (1.1)–(2.1) reads: given $I_{app}(t; \boldsymbol{\mu}) \in L^2(\Omega(\boldsymbol{\mu}))$,

²Dimensional times and potential are given by $\tilde{t}[ms] = 12.9t$ and $\tilde{u}[mV] = 100u - 80$, see [4]. The transmembrane potential indeed ranges from the resting state of -80 mV to the excited state of approximately $+20$ mV.

find $u(t; \boldsymbol{\mu}) \in X = H^1(\Omega(\boldsymbol{\mu}))$ and $w(t; \boldsymbol{\mu}) \in L^2(\Omega(\boldsymbol{\mu}))$ such that

$$\begin{aligned} \int_{\Omega(\boldsymbol{\mu})} \left(\frac{\partial u}{\partial t} + I_{ion}(u, w; \boldsymbol{\mu}) \right) \psi d\mathbf{x} + \int_{\Omega(\boldsymbol{\mu})} \mathbf{D}(\boldsymbol{\mu}) \nabla u \cdot \nabla \psi d\mathbf{x} &= \int_{\Omega(\boldsymbol{\mu})} I_{app}(t; \boldsymbol{\mu}) \psi d\mathbf{x} \quad \forall \psi \in H^1(\Omega(\boldsymbol{\mu})), \\ \int_{\Omega(\boldsymbol{\mu})} \frac{\partial w}{\partial t} \eta d\omega &= \int_{\Omega(\boldsymbol{\mu})} g(u, w; \boldsymbol{\mu}) \eta d\omega \quad \forall \eta \in L^2(\Omega(\boldsymbol{\mu})), \\ u(0; \boldsymbol{\mu}) &= u_0, \quad w(0; \boldsymbol{\mu}) = w_0, \end{aligned}$$

where $\Omega(\boldsymbol{\mu})$ is a Lipschitz domain of \mathbb{R}^p , $p = 2, 3$, possibly depending on a set of geometrical parameters. Here $\mathbf{D}(\boldsymbol{\mu}) = \mathbf{D}(\mathbf{x}; \boldsymbol{\mu})$ represents the conductivity tensor, which can vary within the myocardium due to fiber orientation and unhealthy conditions (e.g. the possible presence of ischemic regions). Well-posedness results of this coupled problem based on semi-discretization in time can be derived from the more general results established in [38] for the bidomain model coupled with the FitzHugh-Nagumo model, and in [39, 40, 41] where a Faedo-Galerkin technique is instead considered. In this work we focus on the case where a set of physical properties affect the diffusivity matrix \mathbf{D} (through the conduction velocities or the fibers' structure), the gating variable dynamics (through g) and the ionic current I_{ion} .

2.2. Full-order model: Galerkin finite element approximation

We apply the Galerkin-FE method on a finite-dimensional space $X_h \subset X(\Omega)$ of (usually very large) dimension $\dim(X_h) = N_h$; h denotes a parameter related to the mesh size of the computational grid. By denoting with $\{\varphi_i\}_{i=1}^{N_h}$ a set of basis functions of the FE space X_h , we express the discrete approximation to $u(\mathbf{x}, t; \boldsymbol{\mu})$ and $w(\mathbf{x}, t; \boldsymbol{\mu})$ by

$$u_h(\mathbf{x}, t; \boldsymbol{\mu}) = \sum_{i=1}^{N_h} u_i(t; \boldsymbol{\mu}) \varphi_i(\mathbf{x}), \quad w_h(\mathbf{x}, t; \boldsymbol{\mu}) = \sum_{i=1}^{N_h} w_i(t; \boldsymbol{\mu}) \varphi_i(\mathbf{x}),$$

where the vectors $\mathbf{u}_h = [u_1, \dots, u_{N_h}]^T$ and $\mathbf{w}_h = [w_1, \dots, w_{N_h}]^T$ are obtained by solving the following discrete system: given $\boldsymbol{\mu} \in \mathcal{P}$, find $\mathbf{u}_h = \mathbf{u}_h(t; \boldsymbol{\mu})$ and $\mathbf{w}_h = \mathbf{w}_h(t; \boldsymbol{\mu})$ such that

$$\begin{cases} \mathbf{M}(\boldsymbol{\mu}) \frac{\partial \mathbf{u}_h}{\partial t} + \mathbf{A}(\boldsymbol{\mu}) \mathbf{u}_h + \mathbf{I}_{ion}(\mathbf{u}_h, \mathbf{w}_h; \boldsymbol{\mu}) = \mathbf{I}_{app}(t; \boldsymbol{\mu}), & t \in (0, T) \\ \frac{\partial \mathbf{w}_h}{\partial t} = g(\mathbf{u}_h, \mathbf{w}_h), & t \in (0, T) \\ \mathbf{u}_h(0; \boldsymbol{\mu}) = \mathbf{u}_0(\boldsymbol{\mu}), \quad \mathbf{w}_h(0; \boldsymbol{\mu}) = \mathbf{w}_0(\boldsymbol{\mu}). \end{cases}$$

Here we denote the $\boldsymbol{\mu}$ -dependent mass matrix, the stiffness matrix and the activation term by

$$(\mathbf{M}(\boldsymbol{\mu}))_{ij} = \int_{\Omega(\boldsymbol{\mu})} \varphi_i \varphi_j d\mathbf{x}, \quad (\mathbf{A}(\boldsymbol{\mu}))_{ij} = \int_{\Omega(\boldsymbol{\mu})} \mathbf{D}(\boldsymbol{\mu}) \nabla \varphi_i \cdot \nabla \varphi_j d\mathbf{x}, \quad (\mathbf{I}_{app}(t; \boldsymbol{\mu}))_j = \int_{\Omega(\boldsymbol{\mu})} I_{app}(t; \boldsymbol{\mu}) \varphi_j d\mathbf{x},$$

and the $\boldsymbol{\mu}$ -dependent vectors accounting for the ionic terms by

$$(\mathbf{I}_{ion}(\mathbf{u}_h, \mathbf{w}_h; \boldsymbol{\mu}))_j = \int_{\Omega(\boldsymbol{\mu})} I_{ion}(\mathbf{u}_h, \mathbf{w}_h; \boldsymbol{\mu}) \varphi_j d\mathbf{x}, \quad (g(\mathbf{u}_h, \mathbf{w}_h; \boldsymbol{\mu}))_j = \int_{\Omega(\boldsymbol{\mu})} g(\mathbf{u}_h, \mathbf{w}_h; \boldsymbol{\mu}) \varphi_j d\mathbf{x}.$$

Regarding the treatment of nonlinear terms and time discretization, we use a semi-implicit, first order, one-step scheme [15]. Given a partition $(t^{(\ell)}, t^{(\ell+1)})$, $\ell = 0, \dots, N_t - 1$ of $(0, T)$ into N_t subintervals of length Δt , at each time-step $t^{(\ell+1)}$ the nonlinear vector \mathbf{I}_{ion} is evaluated around the solution already computed at time $t^{(\ell)}$. This decouples the PDE from the ODE leading to a linear system to be solved at each time step. Moreover, a ionic current interpolation strategy is used

to evaluate the ionic current term, so that only the nodal values are used to build a (piecewise linear) interpolant of the ionic current. In conclusion, the full-order model (FOM) reads as: given $\boldsymbol{\mu} \in \mathcal{P}$, find $\mathbf{u}_h^{(\ell+1)} = \mathbf{u}_h^{(\ell+1)}(\boldsymbol{\mu})$ and $\mathbf{w}_h^{(\ell+1)} = \mathbf{w}_h^{(\ell+1)}(\boldsymbol{\mu})$ such that $\mathbf{u}_h^{(0)} = \mathbf{u}_0(\boldsymbol{\mu})$, $\mathbf{w}_h^{(0)} = \mathbf{w}_0(\boldsymbol{\mu})$ and, for $\ell = 0, \dots, N_t - 1$,

$$\begin{cases} \frac{\mathbf{w}_h^{(\ell+1)} - \mathbf{w}_h^{(\ell)}}{\Delta t} - \mathbf{g}(\mathbf{u}_h^{(\ell)}, \mathbf{w}_h^{(\ell+1)}; \boldsymbol{\mu}) = \mathbf{0}, \\ \mathbf{M}(\boldsymbol{\mu}) \frac{\mathbf{u}_h^{(\ell+1)} - \mathbf{u}_h^{(\ell)}}{\Delta t} + \mathbf{A}(\boldsymbol{\mu}) \mathbf{u}_h^{(\ell+1)} + \mathbf{I}_{ion}(\mathbf{u}_h^{(\ell)}, \mathbf{w}_h^{(\ell+1)}; \boldsymbol{\mu}) - \mathbf{I}_{app}^{(\ell+1)}(\boldsymbol{\mu}) = \mathbf{0}. \end{cases} \quad (2.2)$$

The major computational costs are entailed by the assembling the terms \mathbf{I}_{ion} and \mathbf{g} at each time step and by the solution of the linear system (2.2)₂. On its turn, the time step Δt is required to be sufficiently small to ensure the convergence of the method [42]. We outline that our main interest is to solve problem (2.2) for several (order of hundreds, or even thousands) different values of the parameter vector $\boldsymbol{\mu} \in \mathcal{P}$, thus making the use of a FOM hardly infeasible, if not impossible.

3. Reduced-order model: POD-DEIM-Galerkin method

The goal of a (linear) ROM for parametrized PDEs is to approximate the solution of a FOM by means of a linear combination of few, problem-specific, global basis functions (reduced basis) obtained from a set of FOM snapshots. Here, the *integro-differential* nature of the FOM calls into play a different treatment for the potential variable \mathbf{u}_h and the ionic variable \mathbf{w}_h . The approximation of \mathbf{u}_h is sought under the form

$$\mathbf{u}_h^{(\ell)}(\boldsymbol{\mu}) \approx \mathbf{V} \mathbf{u}_n^{(\ell)}(\boldsymbol{\mu}), \quad (3.1)$$

where the columns of $\mathbf{V} \in \mathbb{R}^{N_h \times n}$ (hopefully, $n \ll N_h$) yield (algebraically) the basis functions of X_n^u [26]; how to construct \mathbf{V} will be addressed in Sect. 4. The approximation of \mathbf{w}_h will be instead obtained by evaluating the ODE only in a suitable subset of mesh nodes (the so-called *reduced mesh*) involved in the efficient evaluation of \mathbf{I}_{ion} , as shown below.

3.1. POD-Galerkin method

Let us focus on the approximation to the PDE for the transmembrane potential using a FOM approximation of the recovery variable; in particular, let us assume that this latter has already been updated to its current value at time $t^{(\ell+1)}$ by solving (2.2)₁. Inserting (3.1) into (2.2)₂ and performing a Galerkin projection, we obtain the following Galerkin-RB problem for the monodomain equation: given $\boldsymbol{\mu} \in \mathcal{P}$, find $\mathbf{u}_n^{(\ell+1)}$, $\ell = 0, \dots, N_t - 1$, such that

$$\begin{cases} \mathbf{M}_n(\boldsymbol{\mu}) \frac{\mathbf{u}_n^{(\ell+1)} - \mathbf{u}_n^{(\ell)}}{\Delta t} + \mathbf{A}_n(\boldsymbol{\mu}) \mathbf{u}_n^{(\ell+1)} + \mathbf{V}^T \mathbf{I}_{ion}(\mathbf{V} \mathbf{u}_n^{(\ell)}, \mathbf{w}_h^{(\ell+1)}; \boldsymbol{\mu}) - \mathbf{V}^T \mathbf{I}_{app}^{(\ell+1)}(\boldsymbol{\mu}) = \mathbf{0}, \\ \mathbf{u}_n^{(0)} = \mathbf{V}^T \mathbf{u}_0(\boldsymbol{\mu}) \end{cases} \quad (3.2)$$

where the RB mass and stiffness matrices are obtained as follows:

$$\mathbf{A}_n(\boldsymbol{\mu}) = \mathbf{V} \mathbf{A}(\boldsymbol{\mu}) \mathbf{V}^T, \quad \mathbf{M}_n(\boldsymbol{\mu}) = \mathbf{V} \mathbf{M}(\boldsymbol{\mu}) \mathbf{V}^T. \quad (3.3)$$

The linear operators in the ROM are therefore obtained by a simple pre- and post-multiplication of the corresponding FOM arrays at the algebraic level [26]. If both $\mathbf{A}(\boldsymbol{\mu})$ and $\mathbf{M}(\boldsymbol{\mu})$ depend affinely on the parameter $\boldsymbol{\mu}$, assembling matrices (3.3) for each new parameter vector $\boldsymbol{\mu}$ is very inexpensive as we can make use of pre-computed arrays $\mathbf{V}^T \mathbf{A}_q \mathbf{V}$, $q = 1, \dots, Q_a$, $\mathbf{V}^T \mathbf{M}_q \mathbf{V}$, $q = 1, \dots, Q_m$. In those cases where the $\boldsymbol{\mu}$ -dependence is nonaffine, we rely on discrete EIM, or matrix DEIM (MDEIM), to get an approximate affine expansion of a given vector or matrix, respectively.

3.2. Enhancing efficiency by hyper-reduction

Here we use DEIM to avoid the evaluation of the full-order array $\mathbf{I}_{ion} \in \mathbb{R}^{N_h}$, which would compromise the overall ROM efficiency. The problem is to find $m \ll N_h$ functions $\theta_q: \mathcal{P} \mapsto \mathbb{R}$ and $\boldsymbol{\mu}$ -independent vectors $\mathbf{z}_q \in \mathbb{R}^{N_h}$, $1 \leq q \leq m$, such that

$$\mathbf{I}_{ion}(\mathbf{V}\mathbf{u}_n^{(\ell)}, \mathbf{w}_h^{(\ell+1)}; \boldsymbol{\mu}) \approx \tilde{\mathbf{I}}_{ion}(\boldsymbol{\mu}) = \sum_{q=1}^m \theta_q(t^{(\ell)}; \boldsymbol{\mu}) \mathbf{z}_q.$$

During the offline stage of this procedure we apply DEIM as in [28] to a set of snapshots

$$\mathbf{S}_I = \{\mathbf{I}_{ion}(\mathbf{V}\mathbf{u}_n^{(\ell)}, \mathbf{w}_h^{(\ell+1)}; \boldsymbol{\mu}^k), k = 1, \dots, N_s, \ell = 0, \dots, N_t - 1\} \quad (3.4)$$

in order to obtain the basis $\boldsymbol{\Phi} = [\mathbf{z}_1 \mid \dots \mid \mathbf{z}_m] \in \mathbb{R}^{N_h \times m}$ and a set of m interpolation indices $\mathcal{I} \subset \{1, \dots, N_h\}$, with $|\mathcal{I}| = m$. The former is computed by applying over the columns of \mathbf{S}_I the same local POD technique used for the state space presented in Sect. 4, whereas the latter is iteratively selected by employing the so-called *magic points* algorithm [27, 43].

During the online phase, given a new $\boldsymbol{\mu} \in \mathcal{P}$, we can compute $\tilde{\mathbf{I}}_{ion}(\boldsymbol{\mu})$ as

$$\tilde{\mathbf{I}}_{ion}(\boldsymbol{\mu}) = \boldsymbol{\Phi}\boldsymbol{\theta}(t^{(\ell)}; \boldsymbol{\mu}) \quad \text{with} \quad \boldsymbol{\Phi}_{\mathcal{I}}\boldsymbol{\theta}(t^{(\ell)}; \boldsymbol{\mu}) = \mathbf{I}_{ion, \mathcal{I}}(\mathbf{V}\mathbf{u}_n^{(\ell)}, \mathbf{w}_h^{(\ell+1)}; \boldsymbol{\mu}), \quad (3.5)$$

where $\boldsymbol{\Phi}_{\mathcal{I}}$ and $\mathbf{I}_{ion, \mathcal{I}}(\boldsymbol{\mu})$ denote the matrix formed by the \mathcal{I} rows of $\boldsymbol{\Phi}$ and the vector $\mathbf{I}_{ion}(\boldsymbol{\mu})$ evaluated at the \mathcal{I} entries, respectively. The vector $\boldsymbol{\theta}(t^{(\ell)}; \boldsymbol{\mu}) = [\theta_1(t^{(\ell)}; \boldsymbol{\mu}), \dots, \theta_m(t^{(\ell)}; \boldsymbol{\mu})] \in \mathbb{R}^m$ is evaluated by solving the linear system in (3.5), encoding m interpolation constraints at the mesh points selected in \mathcal{I} . Denoting by $\mathbf{P} = [\mathbf{e}_{\mathcal{I}_1} \mid \dots \mid \mathbf{e}_{\mathcal{I}_m}] \in \mathbb{R}^{N_h \times m}$, where $\mathbf{e}_{\mathcal{I}_i} = [0, \dots, 0, 1, 0, \dots, 0]^T \in \mathbb{R}^{N_h}$ is the \mathcal{I}_i -th column of the identity matrix $\mathbf{I} \in \mathbb{R}^{N_h \times N_h}$, we can express

$$\boldsymbol{\Phi}_{\mathcal{I}} = \mathbf{P}^T \boldsymbol{\Phi}, \quad \mathbf{I}_{ion, \mathcal{I}}(\boldsymbol{\mu}) = \mathbf{I}_{ion}(\mathbf{P}^T \mathbf{V}\mathbf{u}_n^{(\ell)}, \mathbf{P}^T \mathbf{w}_h^{(\ell+1)}; \boldsymbol{\mu}).$$

The crucial step in the online evaluation of $\boldsymbol{\theta}(t^{(\ell)}; \boldsymbol{\mu})$ (that is, for any new parameter instance $\boldsymbol{\mu}$) is the computation of $\mathbf{I}_{ion, \mathcal{I}}(\boldsymbol{\mu})$; however, this quantity can be evaluated efficiently by employing the same assembly routine used for the full-order problem on the *reduced mesh* associated to the selected interpolation indices; see, e.g., [30] for further details. In conclusion, the ionic term in the potential equation can be approximated by

$$\mathbf{V}^T \mathbf{I}_{ion}(\mathbf{V}\mathbf{u}_n^{(\ell)}, \mathbf{w}_h^{(\ell+1)}; \boldsymbol{\mu}) \approx \underbrace{\mathbf{V}^T \boldsymbol{\Phi} (\mathbf{P}^T \boldsymbol{\Phi})^{-1}}_{n \times m} \underbrace{\mathbf{I}_{ion}(\mathbf{P}^T \mathbf{V}\mathbf{u}_n^{(\ell)}, \mathbf{P}^T \mathbf{w}_h^{(\ell+1)}; \boldsymbol{\mu})}_{m \times 1}.$$

This means that the point-wise approximation of the ODE could be advanced in time only on the m dofs $\mathcal{I}_1, \dots, \mathcal{I}_m$ forming the *reduced mesh*, thus gaining an additional speedup in the resolution of the problem. Hence, the *hyper-reduced order model* reads as follows:

$$\begin{cases} \mathbf{P}^T \frac{\mathbf{w}_h^{(\ell+1)} - \mathbf{w}_h^{(\ell)}}{\Delta t} - \mathbf{g}(\mathbf{P}^T \mathbf{V}\mathbf{u}_n^{(\ell)}, \mathbf{P}^T \mathbf{w}_h^{(\ell+1)}; \boldsymbol{\mu}) = \mathbf{0}, & \ell = 0, \dots, N_t - 1, \\ \mathbf{M}_n(\boldsymbol{\mu}) \frac{\mathbf{u}_n^{(\ell+1)} - \mathbf{u}_n^{(\ell)}}{\Delta t} + \mathbf{A}_n(\boldsymbol{\mu}) \mathbf{u}_n^{(\ell+1)} \\ \quad + \mathbf{V}^T \boldsymbol{\Phi} (\mathbf{P}^T \boldsymbol{\Phi})^{-1} \mathbf{I}_{ion}(\mathbf{P}^T \mathbf{V}\mathbf{u}_n^{(\ell)}, \mathbf{P}^T \mathbf{w}_h^{(\ell+1)}; \boldsymbol{\mu}) - \mathbf{V}^T \mathbf{I}_{app}^{(\ell+1)}(\boldsymbol{\mu}) = \mathbf{0}, & \ell = 0, \dots, N_t - 1, \\ \mathbf{u}_n^{(0)} = \mathbf{V}^T \mathbf{u}_0(\boldsymbol{\mu}). \end{cases} \quad (3.6)$$

Remark 1. *The same DEIM procedure has been extended in [30] to deal with parameter-dependent matrices; see, e.g., [44] for further details about the application of this procedure to nonlinear problems in cardiac mechanics. This approach will be exploited in test case 2, where a nonaffine $\boldsymbol{\mu}$ -dependence affects any matrix or vector quantity because of the geometrical parametrization.*

Summarizing, during the offline stage the following operations, must be performed:

1. from a set of snapshots $\mathbf{S}_u = [\mathbf{u}_h(t^{(1)}, \boldsymbol{\mu}_k), \mathbf{u}_h(t^{(1)}, \boldsymbol{\mu}_k), \dots, \mathbf{u}_h(t^{(N_t)}, \boldsymbol{\mu}_k)]$, $k = 1, \dots, N_s$, a basis \mathbf{V} for the potential variable is built through POD (see Sect. 4);
2. the linear ROM operators are obtained according to (3.3);
3. a first (still not efficient) ROM is built by evaluating the nonlinear term \mathbf{I}_{ion} exactly;
4. snapshots $\{\mathbf{I}_{ion}(\mathbf{V}\mathbf{u}_n^{(\ell)}, \mathbf{w}_h^{(\ell+1)}; \boldsymbol{\mu}^k), k = 1, \dots, N_s, \ell = 0, \dots, N_t - 1\}$ are computed, then a basis Φ is built through POD;
5. the $\boldsymbol{\mu}$ -independent matrix $\mathbf{V}^T \Phi (\mathbf{P}^T \Phi)^{-1}$ is assembled and stored.

In a subsequent *online* phase, the ROM can then be exploited to efficiently approximate the problem solution for each new parameter instance. At each time-step, a ROM query only requires to assemble the nonlinear terms on the *reduced mesh* and to solve the nonlinear system (3.2). We summarize this POD-DEIM-Galerkin approach in Algorithm 1.

Algorithm 1 Standard POD-DEIM procedure

```

1: procedure [ ROM ARRAYS] = OFFLINE(FOM ARRAYS,  $P_{train}, \epsilon_{tol}^u, \epsilon_{tol}^I$ )
2:   Full-order matrices:
3:    $\{\Theta_j^m(\boldsymbol{\mu}), \mathbf{M}_j\} \leftarrow$  affine decomposition of  $\mathbf{M}(\boldsymbol{\mu})$ 
4:    $\{\Theta_j^a(\boldsymbol{\mu}), \mathbf{A}_j\} \leftarrow$  affine decomposition of  $\mathbf{A}(\boldsymbol{\mu})$ 
5:    $\{\Theta_j^I(\boldsymbol{\mu}), \mathbf{I}_{app,j}\} \leftarrow$  affine decomposition of  $\mathbf{I}_{app}$ 
6:   Solution snapshots:
7:   for  $\boldsymbol{\mu} \in P_{train}$  do
8:     for  $\ell = 1, \dots, N_t$  do
9:        $\mathbf{S}_u = [\mathbf{S}_u, \mathbf{u}_h^{(\ell)}(\boldsymbol{\mu})]$ ;
10:    end for
11:  end for
12:   $\mathbf{V} \leftarrow$  POD( $\mathbf{S}_u, \epsilon_{tol}^u$ );
13:  Reduced-order matrices:
14:   $\{\mathbf{M}_{n,j}, \mathbf{A}_{n,j}, (\mathbf{I}_{app,n})_j\} \leftarrow$  projection of the full order arrays onto  $\mathbf{V}$ 
15:  Nonlinear term snapshots:
16:  for  $\boldsymbol{\mu} \in P_{train}$  do
17:    for  $\ell = 1, \dots, N_t$  do
18:       $\mathbf{S}_I = [\mathbf{S}_I, \mathbf{I}_{ion}(\mathbf{V}\mathbf{u}_n^{(\ell)}, \mathbf{w}_h^{(\ell+1)}; \boldsymbol{\mu})]$ ;
19:    end for
20:  end for
21:  DEIM reduced-order arrays:
22:   $\Phi \leftarrow$  POD( $\mathbf{S}_I, \epsilon_{tol}^I$ );  $\mathbf{P} \leftarrow$  DEIM-indices( $\Phi$ );
23: end procedure
24:
25: procedure [  $\mathbf{u}_n$ ] = ONLINE QUERY(ROM ARRAYS,  $\boldsymbol{\mu}, \mathbf{u}_{n,0}, \mathbf{P}^T \mathbf{w}_0$ )
26:   Reduced-order matrices:
27:    $\mathbf{M}_n(\boldsymbol{\mu}) = \sum_j \Theta_j^m(\boldsymbol{\mu}) \mathbf{M}_{n,j}$ ;  $\mathbf{A}_n(\boldsymbol{\mu}) = \sum_j \Theta_j^a(\boldsymbol{\mu}) \mathbf{A}_{n,j}$ ;  $\mathbf{I}_{app,n} = \sum_j \Theta_j^I(\boldsymbol{\mu}) (\mathbf{I}_{app,n})_j$ ;
28:   for  $\ell = 1, \dots, N_t$  do
29:      $\mathbf{P}^T \mathbf{w}_n^{(\ell+1)} \leftarrow$  solve ODE system in (3.6);
30:     assemble over the reduced mesh  $\mathbf{I}_{ion}(\mathbf{P}^T \mathbf{V}\mathbf{u}_n^{(\ell)}, \mathbf{P}^T \mathbf{w}_h^{(\ell+1)}; \boldsymbol{\mu})$ ;
31:     DEIM approximation  $\leftarrow \mathbf{V}^T \Phi (\mathbf{P}^T \Phi)^{-1} \mathbf{I}_{ion}(\mathbf{P}^T \mathbf{V}\mathbf{u}_n^{(\ell)}, \mathbf{P}^T \mathbf{w}_h^{(\ell+1)}; \boldsymbol{\mu})$ ;
32:      $\mathbf{u}_n^{(\ell+1)} \leftarrow$  solve PDE linear system in (3.6);
33:   end for
34: end procedure

```

4. Local Reduced basis method

Parametrized problems in cardiac electrophysiology might easily yield solutions showing a remarkable variability over the parameter space. For instance, the fibers direction and their maximum angle have a large impact on both the direction and the speed of the wave front; the same conclusion holds when some of the parameters affecting the cell model are varied. This causes the solution manifold $\mathcal{M}_{h,\Delta t} = \{\mathbf{u}_h(t^{(\ell)}, \boldsymbol{\mu}) : \ell = 0, \dots, N_t - 1, \boldsymbol{\mu} \in \mathcal{P}\}$ to be highly nonlinear; as a matter of fact, its (linear) approximation by means of a single linear subspace yields accurate approximation only at the price of considering very large dimensions (up to some hundreds) n and m of the POD expansion and of the DEIM approximation, respectively, thus preventing the ROM from ensuring a considerable speed-up compared to the FOM.

Multiple local subspaces can be generated when performing the RB approximation of the PDE solution, and the DEIM approximation can be used for the nonlinear term. Approximating the manifold by a series of subspaces of smaller dimension results in a more efficient approach than building a single subspace of larger dimension. With this aim, we employ a four-step procedure as proposed in [45]:

1. we collect $N_s = N_t \times N_{train}$ snapshots $\{\mathbf{u}_h(t^{(1)}, \boldsymbol{\mu}_k), \mathbf{u}_h(t^{(2)}, \boldsymbol{\mu}_k), \dots, \mathbf{u}_h(t^{(N_t)}, \boldsymbol{\mu}_k)\}_{k=1, \dots, N_{train}}$, into a matrix \mathbf{S}_u by solving the FOM over time for suitably chosen parameter values;
2. for a given N_c we group snapshots into clusters \mathbf{S}_u^k , $k = 1, \dots, N_c$; each column of \mathbf{S}_u is thus assigned to a cluster accordingly to a given criterion;
3. we construct a local reduced basis for each cluster through POD;
4. we construct a ROM for each cluster by projecting the original FOM onto each reduced subspace \mathbf{V}_k , $k = 1, \dots, N_c$, as in the classical POD-Galerkin method.

As soon as the local ROMs for the state solution have been built, snapshots of the nonlinear term are computed to form the matrix \mathbf{S}_I defined in (3.4). Then, also the columns of \mathbf{S}_I are partitioned into N_c clusters \mathbf{S}_I^k , $k = 1, \dots, N_c$ and the DEIM procedure of Section 3.2 is applied to each cluster, yielding a set of N_c bases $\boldsymbol{\Phi}^k$, $k = 1, \dots, N_c$, to be used to approximate the nonlinear term efficiently. The same number N_c of clusters is chosen for both approximations, although in principle a different number of clusters could be selected to partition both column sets \mathbf{S}_u and \mathbf{S}_I .

The online query to the ROM is then performed by exploiting the local RB matrices and vectors, as well as the local DEIM approximation associated to the reduced-subspace selected at step 2; note that switching from a local ROM to a new one must be done inexpensively during the online stage. In this paper we compare several strategies to perform snapshot clustering, employing in any case a Galerkin projection to construct a ROM for each cluster. Neighboring snapshots can be either added or not to each cluster to obtain overlapping clusters; in our examples we do not consider any overlap among clusters, although in principle this can also be done [45, 46].

The approach described above was firstly proposed in [45] to address the construction of local ROMs in the *state space* – although without constructing a ROM to be systematically queried over the parameter space – and further extended in [47, 48, 46]. It was also applied in [49] for the sake of approximating nonlinear quantities by DEIM. Other options we are going to explore to build multiple approximations rely on a partition of the *time interval* or the *parameter space* as well. In order to compare these options, we will consider as main criteria the *efficiency* in terms of speed-up regarding CPU time with respect to the FOM solution, *accuracy* (by measuring the error between the ROM and FOM solutions) and memory storage.

4.1. Time-based clustering

Since the system dynamics consists of a traveling front which characterizes the depolarization mechanism of the electrical potential, a natural clustering of \mathbf{S}_u can be obtained by considering clusters related to different temporal windows. For the sake of simplicity we consider the same length for each time window, although an adaptive procedure could also be set. See, e.g., [50, 51] for the original development of this technique, and [52, 44] for further applications to more involved problems. In [52] a more advanced version of time-based subdivision is suggested, consisting of the combination of global and local modes selected for (non overlapping) subintervals of time in a goal-oriented way, in order to better capture small temporal scales.

To perform a time-based clustering, for a large $K > 1$ we introduce a coarse partition of the time interval $(0, T)$ into N_c windows $(\tau^{(k)}, \tau^{(k+1)})$ of length $\Delta\tau = K\Delta t$, $k = 0, \dots, N_c - 1$ (see Fig. 1). Following this approach, the index ℓ of the time-step $t^{(\ell)}$ becomes the cluster indicator, that is, each solution $\mathbf{u}_h^{(\ell)}(\boldsymbol{\mu})$ is assigned to a specific cluster k if $t^{(\ell)} \in (\tau^{(k-1)}, \tau^{(k)}]$.

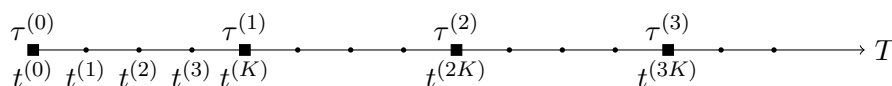


Figure 1: A partition of the time interval in windows of length $\Delta\tau = K\Delta t$, $K = 4$.

Then, the matrices $\mathbf{V}_k \in \mathbb{R}^{N_h \times n_k}$, $k = 1, \dots, N_c$, collecting the local basis functions computed by the POD technique, enable to approximate the full-order solution as

$$\mathbf{u}_h^{(\ell)}(\boldsymbol{\mu}) \approx \mathbf{V}_k \mathbf{u}_{n_k}^{(\ell)}(\boldsymbol{\mu}), \quad k : t^{(\ell)} \in (\tau^{(k-1)}, \tau^{(k)}].$$

Similarly, the matrices $\mathbf{U}_k \in \mathbb{R}^{N_h \times m_k}$, $k = 1, \dots, N_c$ collecting the local basis functions of the DEIM approximation of the nonlinear term \mathbf{I}_{ion} , allow to express the non-linear term as

$$\mathbf{I}_{ion}(\mathbf{u}_n^{(\ell)}, \mathbf{w}_h^{(\ell)}; \boldsymbol{\mu}) \approx \underbrace{\mathbf{V}_k^T \mathbf{U}_k (\mathbf{P}_k^T \mathbf{U}_k)^{-1}}_{n_k \times m_k} \underbrace{\mathbf{I}_{ion}(\mathbf{P}_k^T \mathbf{V}_k \mathbf{u}_n^{(\ell)}, \mathbf{P}_k^T \mathbf{w}_h^{(\ell)}; \boldsymbol{\mu})}_{m_k \times 1}, \quad k : t^{(\ell)} \in (\tau^{(k-1)}, \tau^{(k)}].$$

This approach is effective if the propagation velocity of the signal is constant with respect to the parameters. If not (e.g. when the conductivity field is parametrized), similar solutions could be assigned to different clusters, thus affecting the overall efficiency of the local ROM.

4.2. Parameter-based clustering

An alternative approach is obtained by considering the vector of parameters as cluster indicator, that is, by assigning each solution $\mathbf{u}_h^{(\ell)}(\boldsymbol{\mu})$ of \mathbf{S}_u to a specific cluster k if $\boldsymbol{\mu} \in \mathcal{P}_k$, where $\mathcal{P} = \cup_k \mathcal{P}_k$ is a suitable partition of the parameter space. Except for this new criterion used for subdividing the snapshots matrix \mathbf{S}_u , the offline procedure is similar to the time-based clustering of Sect. 4.1: for each cluster the basis functions of the solution and of the nonlinear term are computed through the POD technique and consequently stored (see Algorithm 2).

The problem is then shifted to finding the optimal subdivision $\{\mathcal{P}_k\}_{k=1}^{N_c}$ to obtain low dimensional local ROMs. A sequential partitioning has been firstly proposed in [53], by dividing the parameter domain using grid-adaptive refinement until each partition satisfies a given accuracy and size of the local reduced space. The hp refinement of RB methods has been considered in [54, 55, 56] by subdividing the original parameter domain into smaller regions (h -refinement) and then constructing individual RB approximation spaces spanned by snapshots restricted to parameter values within each of these parameter subdomains (p -refinement). An alternative strategy exploiting an anisotropic notion of measure in the parameter space has also been proposed [57].

A parameter-based clustering could easily lead to a high number of clusters, hence to high *offline* computational costs. Moreover, this partitioning also neglects the possibility that different parameters could generate similar solutions for different time-steps. An approach aiming at partitioning the parameter domain in cardiac electrophysiology has been first proposed in [31]: it is based on the empirical observation of the approximation error between the FOM and a POD-DEIM-Galerkin ROM. This construction of the local ROMs might generate a large number of small subregions, and other drawbacks similar to the sequential partitioning proposed in [53].

4.3. State-based clustering

In this third approach the snapshot matrix \mathbf{S}_u is partitioned into (non overlapping) subregions constructed with respect to the state vector $\mathbf{u}_h^{(\ell)}$ by minimizing its distance from preselected centroids $\{\mathbf{c}^k\}$. In the online stage, at each time step $\ell = 0, \dots, N_t - 1$ the subregion where the current state lies is first identified. Then, the local reduced arrays associated to the selected cluster are used for the approximation of $\mathbf{u}_h^{(\ell+1)}$ in the space spanned by the local basis functions. In the following we will focus on two possible techniques for partitioning \mathbf{S}_u : the *k-means* algorithm and a new partitioning techniques based on a binary tree recently developed in [46].

4.3.1. Case a: k-means clustering

The k-means algorithm [58, 59] is a well-known unsupervised statistical learning technique for finding clusters and cluster centers in an unlabeled dataset. The clusters are constructed in such a way that the snapshots within each group are similar to each other, while snapshots in different groups are different from each other with respect to a chosen metric.

We employ *k-means* to partition the snapshots matrix \mathbf{S}_u into N_c submatrices $\{\mathbf{S}_u^1, \dots, \mathbf{S}_u^{N_c}\}$ in order to minimize the distance between each vector in the cluster and the cluster sample mean. In other words, the objective is to find:

$$\{\mathbf{S}_u^1, \dots, \mathbf{S}_u^k\} = \arg \min_{\mathbf{S}_u} \sum_{k=1}^{N_c} \sum_{\mathbf{u}_h \in \mathbf{S}_u^k} \|\mathbf{u}_h - \mathbf{c}_h^k\|_{\mathbf{X}}^2, \quad \mathbf{c}_h^k = \frac{1}{|\mathbf{S}_u^k|} \sum_{\mathbf{u}_h \in \mathbf{S}_u^k} \mathbf{u}_h, \quad k = 1, \dots, N_c,$$

during the *offline* stage. Here, $\{\mathbf{c}_h^k\}_{k=1}^{N_c}$ are the so-called centroids (i.e. the cluster centers) selected by the k-means algorithm with respect to the selected norm $\|\cdot\|_{\mathbf{X}}$. Hereon, we denote by $\|\cdot\|_{\mathbf{X}}$ the algebraic counterpart of the X -norm defined at the continuous level; $\mathbf{X} \in \mathbb{R}^{N_h \times N_h}$ is a symmetric positive definite matrix defining the chosen metric. For the case at hand, $X = H^1(\Omega(\boldsymbol{\mu}))$, so that \mathbf{X} results from the sum of the stiffness matrix and the mass matrix, that is,

$$(\mathbf{X})_{ij} = \int_{\Omega(\boldsymbol{\mu})} (\nabla \varphi_i \cdot \nabla \varphi_j + \varphi_i \varphi_j) d\mathbf{x}, \quad i, j = 1, \dots, N_h;$$

we omit the $\boldsymbol{\mu}$ -dependence in the definition of \mathbf{X} for the sake of notation.

In the *online* stage a local ROM is selected at each time step $\ell = 0, \dots, N_t - 1$ with respect to the current solution of the system $\mathbf{V}\mathbf{u}_n^{(\ell)}(\boldsymbol{\mu})$ by minimizing the distance between $\mathbf{V}\mathbf{u}_n^{(\ell)}(\boldsymbol{\mu})$ and the centroids:

$$\bar{k} = \arg \min_k \|\mathbf{V}\mathbf{u}_n^{(\ell)}(\boldsymbol{\mu}) - \mathbf{c}_h^k\|_{\mathbf{X}}^2. \quad (4.1)$$

The main advantage of the k-means approach is the automatic detection of the similarities between the snapshots, which is not guaranteed in the time- and parameter-based localization strategies. The additional cost of solving the minimization problem (4.1) decreases when the reduced arrays are considered: indeed the objective in (4.1) can be rewritten as:

$$\|\mathbf{V}\mathbf{u}_n^{(\ell)}(\boldsymbol{\mu}) - \mathbf{c}_h^k\|_{\mathbf{X}}^2 = (\mathbf{V}\mathbf{u}_n^{(\ell)}(\boldsymbol{\mu}) - \mathbf{c}_h^k)^T \mathbf{X} (\mathbf{V}\mathbf{u}_n^{(\ell)}(\boldsymbol{\mu}) - \mathbf{c}_h^k), \quad k = 1, \dots, N_c. \quad (4.2)$$

We then rewrite the right-hand side of (4.2) as

$$(\mathbf{V}\mathbf{u}_n^{(\ell)}(\boldsymbol{\mu}) - \mathbf{c}_h^k)^T \mathbf{X} (\mathbf{V}\mathbf{u}_n^{(\ell)}(\boldsymbol{\mu}) - \mathbf{c}_h^k) = (\mathbf{u}_n^{(\ell)}(\boldsymbol{\mu}))^T (\mathbf{V}^T \mathbf{X} \mathbf{V}^T) \mathbf{u}_n^{(\ell)}(\boldsymbol{\mu}) - 2(\mathbf{u}_n^{(\ell)}(\boldsymbol{\mu}))^T (\mathbf{V}^T \mathbf{X} \mathbf{c}_h^k) + \|\mathbf{c}_h^k\|_{\mathbf{X}}^2,$$

in order to separate the constant term from the ones depending on the RB solution. As a consequence, during the offline phase the norms $\|\mathbf{c}_h^k\|_{\mathbf{X}}^2$ can be precomputed for each $k = 1, \dots, N_c$ and the reduced matrices $\mathbf{V}^T \mathbf{X} \mathbf{V}^T$ and $\mathbf{V}^T \mathbf{X} \mathbf{c}_h^k$ can be preassembled. Then, the online evaluation of the k norms in (4.2) can be performed efficiently only relying on low-dimensional arrays.

4.3.2. Case b: projection error based local ROM (PEBL-ROM)

An alternative approach, firstly developed in [46], builds up a hierarchical partitioning of the state space based on a binary tree structure and the projection error

$$E_{\Pi}(\mathbf{u}, \mathbf{c}) = \|\mathbf{u} - \Pi_{\mathbf{c}} \mathbf{u}\|_{\mathbf{X}},$$

where $\Pi_{\mathbf{c}} : \mathbb{R}^{N_h} \rightarrow \mathbb{R}^{N_h}$ is the orthogonal projection operator of the state \mathbf{u} along the direction spanned by \mathbf{c} . This different metric is adopted in order to better capture scale-invariances, which are completely ignored when the standard Euclidean distance is considered.

During the *offline* phase the snapshots set \mathbf{S}_u is sequentially divided into $\{\mathbf{S}_u^k\}_{k=1}^{N_c}$ with a bisection procedure yielding a binary tree \mathcal{T} . Each node contains an anchor point \mathbf{c} (associated to the orthogonal projector operator) and the relative subset of the training snapshots resulting from the scanning of \mathcal{T} . This recursive bisection of \mathbf{S}_u is obtained by comparing the projection errors $E_{\Pi}(\mathbf{u}, \mathbf{c})$ of each vector $\mathbf{u} \in \mathbf{S}_u$ over the corresponding spaces spanned by the anchor points contained in the leaves (see Figure 2 for a schematic representation).

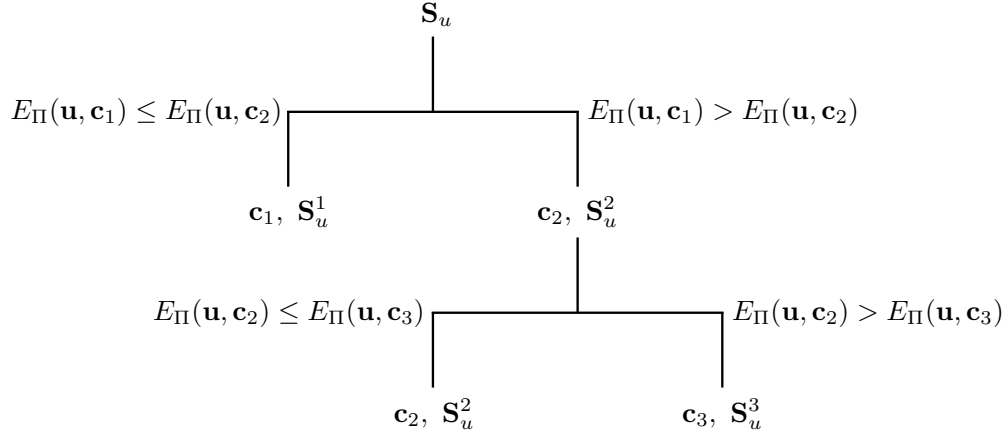


Figure 2: Example of construction of the PEBL binary tree \mathcal{T} during the *offline* phase: the snapshot matrix \mathbf{S}_u is sequentially divided into $\{\mathbf{S}_u^k\}_{k=1}^3$ with a bisection procedure.

Starting from the anchor point \mathbf{c}_1 , selected as the solution in \mathbf{S}_u with maximum norm, each step of the recursive construction of the tree \mathcal{T} includes the following operations:

1. select the node to be branched as the one which maximizes the leaf projection error:

$$\bar{k} = \arg \max_{k=1, \dots, |\mathcal{T}|} \left(\max_{\mathbf{u} \in \mathbf{S}_u^k} E_{\Pi}(\mathbf{u}, \mathbf{c}_k) \right);$$

2. search for the next anchor point \mathbf{c}_{new} as

$$\mathbf{c}_{new} = \max_{\mathbf{u} \in \mathbf{S}_u^{\bar{k}}} E_{\Pi}(\mathbf{u}, \mathbf{c}_{\bar{k}});$$

3. divide $\mathbf{S}_u^{\bar{k}}$ into $\mathbf{S}_u^{new} = \{\mathbf{u} \in \mathbf{S}_u^{\bar{k}} | E_{\Pi}(\mathbf{u}, \mathbf{c}_{\bar{k}}) > E_{\Pi}(\mathbf{u}, \mathbf{c}_{new})\}$ and $\mathbf{S}_u^{\bar{k}} = \mathbf{S}_u^{\bar{k}} \setminus \mathbf{S}_u^{new}$.

When the recursive construction of the tree is completed, the local basis functions are computed by applying POD on $\{\mathbf{S}_u^k\}_{k=1}^{N_c}$ contained in the leafs (see algorithm 2). During the *online* phase, at each time step the bisection tree is scanned with respect to the minimum projection error along the two candidate anchor points. When a leaf node is reached, the associated local arrays are used for approximation of the state at the following time-step.

5. Numerical results

In this section we investigate the numerical performances of the different strategies for the construction of the local ROMs described in the previous Section. We present a detailed comparison of the numerical results by looking at the basic features that in our opinion should be respected in the design and development of a ROM: efficiency, accuracy and low memory storage³.

5.1. Test case I: monodomain equation

We first consider a two-dimensional monodomain model describing the behavior of the cardiac potential in presence of an ischemic region over a simplified slab of the myocardial tissue $\Omega = (0, 1)^2$. We assume an isotropic conductivity tensor $\mathbf{D} = \sigma_v(\mathbf{x}; \boldsymbol{\mu})\mathbf{I}$, being $\sigma_v(\mathbf{x}; \boldsymbol{\mu})$ a non-homogeneous parametrized conductivity field:

$$\sigma_v(\mathbf{x}; \boldsymbol{\mu}) = \sigma_h \rho(\mathbf{x}; \boldsymbol{\mu}) + \sigma_i (1 - \rho(\mathbf{x}; \boldsymbol{\mu})) \quad \rho(\mathbf{x}; \boldsymbol{\mu}) = 1 - \exp\left(-\frac{(x_1 - \mu_1)^2 + (x_2 - \mu_2)^2}{2\mu_3^2}\right),$$

where $\mu_1 \in [0.25, 0.75]$ and $\mu_2 \in [0.25, 0.75]$ are two parameters representing the coordinates of the center of the ischemic area, and $\mu_3 \in [0.05, 0.4]$ its size. Here, we neglect the role of the recovery variable w : as a consequence, the model is only able to describe depolarization patterns, which are represented in Figure 3 for different choices of the parameters vector $\boldsymbol{\mu} = (\mu_1, \mu_2, \mu_3)^T$. The simplified model consists of (1.1)₁ in which $I_{ion}(u; \boldsymbol{\mu}) = \rho(\mathbf{x}; \boldsymbol{\mu})u(u - a)(u - 1)$ is the model for the ionic current (obtained by neglecting the term wu in (2.1)) and

$$I_{app}(\mathbf{x}, t) = C \exp\left(-\frac{x_1^2 + x_2^2}{0.02}\right) \mathbf{1}_{[0, \Delta t]}(t)$$

the initial applied stimulus⁴. Despite the simplifying assumptions that we have made, the solution of the problem shows a great variability due to the different propagation patterns caused by the location and dimensions of the ischemic region (see Figure 3). As a consequence, this problem represents a very challenging test case for developing efficient ROM techniques.

In this case, the matrix $\mathbf{A}(\boldsymbol{\mu})$ is non-affine because of the presence of the term $\rho(\mathbf{x}; \boldsymbol{\mu})$. The vector $\boldsymbol{\rho}(\boldsymbol{\mu})$, obtained by evaluating $\rho(\mathbf{x}; \boldsymbol{\mu})$ on the mesh nodes, can be approximated by the DEIM method (see Section 3.2) thus yielding:

$$\boldsymbol{\rho}_{DEIM}(\boldsymbol{\mu}) = \sum_{j=1}^{m_D} \beta_j(\boldsymbol{\mu}) \boldsymbol{\zeta}^j.$$

³The numerical simulations have been performed on a desktop computer equipped with an Intel Xeon CPU E5. The code has been developed in Matlab using as external library the RB package redbKIT [60].

⁴Here $\mathbf{1}$ denotes the indicator function, defined as

$$\mathbf{1}_{(a,b)}(t) = \begin{cases} 0 & \text{if } t \notin (a, b) \\ 1 & \text{if } t \in (a, b). \end{cases}$$

Algorithm 2 Offline procedure: local ROMs construction

```
1: procedure [ ROM ARRAYS] = OFFLINE(FOM ARRAYS,  $P_{train}, \epsilon_{tol}, N_c$ )
2:   if case 1: time-based clustering: then
3:     for  $k = 1, \dots, N_c$  do
4:       for  $\mu \in P_{train}$  do
5:         for  $t^{(\ell)} \in (\tau^{(k-1)}, \tau^{(k)})$  do
6:            $\mathbf{S}_u^k = [\mathbf{S}_u^k, \mathbf{u}_h^{(\ell)}(\mu)]; \quad \mathbf{S}_I^k = [\mathbf{S}_I^k, \mathbf{I}_{ion}(\mathbf{u}_h^{(\ell)}, \mathbf{w}_h^{(\ell)}; \mu)];$ 
7:         end for
8:       end for
9:     end for
10:  end if
11:  if case 2: parameter-based clustering: then
12:    for  $k = 1, \dots, N_c$  do
13:      for  $\mu \in \mathcal{P}_k$  do
14:        for  $\ell = 1, \dots, N_t$  do
15:           $\mathbf{S}_u^k = [\mathbf{S}_u^k, \mathbf{u}_h^{(\ell)}(\mu)]; \quad \mathbf{S}_I^k = [\mathbf{S}_I^k, \mathbf{I}_{ion}(\mathbf{u}_h^{(\ell)}, \mathbf{w}_h^{(\ell)}; \mu)];$ 
16:        end for
17:      end for
18:    end for
19:  end if
20:  if case 3: state-based clustering: then
21:    for  $\mu \in P_{train}$  do
22:      for  $\ell = 1, \dots, N_t$  do
23:         $\mathbf{S}_u = [\mathbf{S}_u, \mathbf{u}_h^{(\ell)}(\mu)];$ 
24:      end for
25:    end for
26:    if case 3-a: k-means clustering: then
27:       $\{\mathbf{c}_h^k\} \leftarrow \text{compute centroids}(\mathbf{S}_u)$ 
28:      for  $\mu \in P_{train}$  do
29:        for  $\ell = 1, \dots, N_t$  do
30:           $\bar{k} = \arg \min_k \|\mathbf{u}_h - \mathbf{c}_h^k\|^2$ 
31:           $\mathbf{S}_u^{\bar{k}} = [\mathbf{S}_u^{\bar{k}}, \mathbf{u}_h^{(\ell)}(\mu)]; \quad \mathbf{S}_I^{\bar{k}} = [\mathbf{S}_I^{\bar{k}}, \mathbf{I}_{ion}(\mathbf{u}_h^{(\ell)}, \mathbf{w}_h^{(\ell)}; \mu)];$ 
32:        end for
33:      end for
34:    end if
35:    if case 3-b: PEBL clustering: then
36:       $\mathbf{c}_1 = \arg \max_{\mathbf{u} \in \mathbf{S}_u} \|\mathbf{u}\|, \mathcal{T} = \{\mathbf{c}_1\}$ 
37:       $\mathbf{S}_u^1 = \mathbf{S}_u, \epsilon^1 = \max_{\mathbf{u} \in \mathbf{S}_u} \|\mathbf{u} - \Pi_{\mathbf{c}_1} \mathbf{u}\|$ 
38:      while  $|\mathcal{T}| > N_c$  do
39:        search maximum error leaf index  $\bar{k} = \arg \max_{k=1, \dots, |\mathcal{T}|} \epsilon^k$ 
40:        search next anchor point  $\mathbf{c}_{new} = \max_{\mathbf{u} \in \mathbf{S}_u^{\bar{k}}} \|\mathbf{u} - \Pi_{\mathbf{c}_{\bar{k}}} \mathbf{u}\|$ 
41:        divide  $\mathbf{S}_u^{\bar{k}}$  into  $\mathbf{S}_u^{new} = \{\mathbf{u} \in \mathbf{S}_u^{\bar{k}} \mid \|\mathbf{u} - \Pi_{\mathbf{c}_{\bar{k}}} \mathbf{u}\| > \|\mathbf{u} - \Pi_{\mathbf{c}_{new}} \mathbf{u}\|\}$  and  $\mathbf{S}_u^{\bar{k}} = \mathbf{S}_u^{\bar{k}} \setminus \mathbf{S}_u^{new}$ 
42:        insert  $(\mathbf{c}_{\bar{k}}, \mathbf{c}_{new})$  into  $\mathcal{T}$  as children of leaf anchor point  $\mathbf{c}_{\bar{k}}$ ;
43:        recompute leaf errors  $\epsilon^k = \max_{\mathbf{u} \in \mathbf{S}_u^k} \|\mathbf{u} - \Pi_{\mathbf{c}_k} \mathbf{u}\|$  for  $k = 1, \dots, |\mathcal{T}|$ 
44:      end while
45:    end if
46:  end if
47:  Compute local basis functions:
48:  for  $k = 1, \dots, N_c$  do
49:     $\mathbf{V}_k \leftarrow \text{POD}(\mathbf{S}_u^k, \epsilon_{tol}); \quad \mathbf{U}_k \leftarrow \text{POD}(\mathbf{S}_I^k, \epsilon_{tol}); \quad \mathbf{P}_k \leftarrow \text{DEIMindices}(\mathbf{U}_k);$ 
50:     $\{\mathbf{M}_{n_k}^j, \mathbf{A}_{n_k}^j, \mathbf{I}_{app, n_k}^j\} \leftarrow \text{projection of the full order matrices onto } \mathbf{V}_k$ 
51:  end for
52: end procedure
```

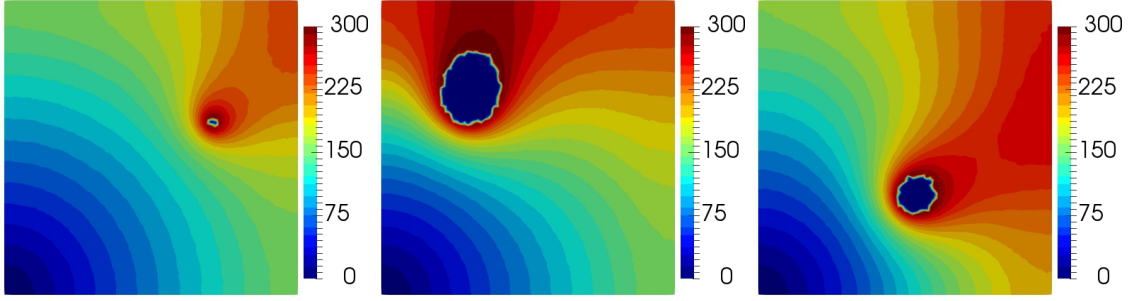


Figure 3: Depolarization time [ms] for three different values of the parameter vector $\boldsymbol{\mu}$. The velocity of the depolarization front slows down when the signal reaches the ischemic region.

Here, ζ^j , $j = 1, \dots, m_D$, are the basis functions used to express the non-affine term ρ , computed using the POD technique; in this case, $m_D = 60$ satisfies the following criterium (see Figure 4):

$$m = \arg \min_k \left(\frac{\sum_i^k \sigma_i^2}{\sum_i^{N_s} \sigma_i^2} \geq 0.99 \right),$$

related to the accuracy of the truncated approximation given by the POD [26]. Then, an affine approximation of the matrix $\mathbf{A}(\boldsymbol{\mu})$ is obtained in this case as $\mathbf{A}(\boldsymbol{\mu}) = \sum_{j=1}^{60} \beta_j(\boldsymbol{\mu}) \mathbf{A}^j$, where the \mathbf{A}^j , $j = 1, \dots, 60$, are full-order matrices assembled considering as diffusive term $(\sigma_h - \sigma_i)\zeta^j + \sigma_i$.

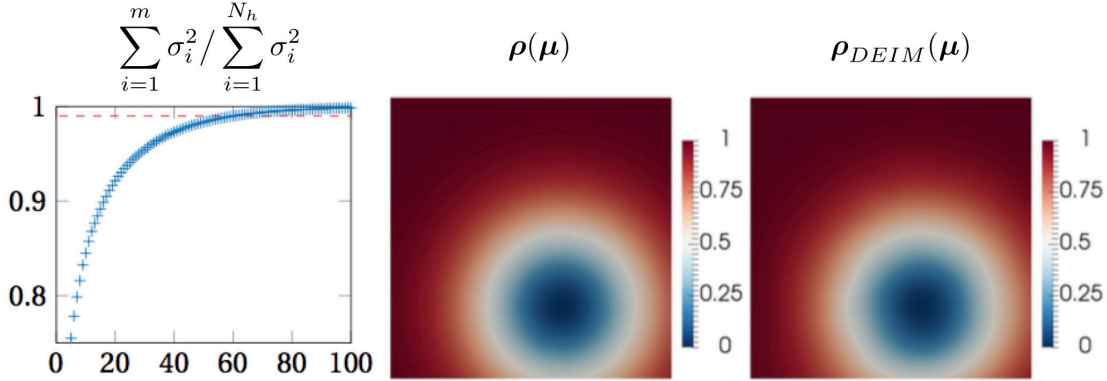


Figure 4: Left: ratio between the truncated sum of the retained singular values squares and their total sums. Center and right: $\rho(\boldsymbol{\mu})$ and its DEIM approximation $\rho_{DEIM}(\boldsymbol{\mu})$ for $\boldsymbol{\mu} = [0.39, 0.26, 0.22]$.

5.1.1. Global ROM

Here we show how the standard POD-DEIM-Galerkin approach, based on a unique global basis, is not feasible for the solution of the problem at hand, since the solutions manifold is considerably complex. In fact, the lack of conductivity substantially modifies the shape of the traveling front, which differs from a parameter to another (see the activation times reported in Figure 3). This difficulty affects both the dimensionality of the training set $\{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_{N_{train}}\}$ (and, as a consequence, the dimensionality of \mathbf{S}_u) and the resulting number m of POD-DEIM basis functions.

We first evaluate the singular values decay by varying the dimension $N_{train} = \{20, 40, 60, 80, 100\}$ of the training set (the corresponding snapshots matrix \mathbf{S}_u is formed by $N_s = N_t N_{train}$ full-order vectors, being $N_t = 300$ the number of time-steps adopted by the semi-implicit time-advancing

scheme). The decay of the singular values for the solution and the nonlinear term, reported in Figure 5, shows that the training set needs to be sufficiently rich ($N_{train} = 80$, at least) in order to correctly capture the variability of the solution over the parameter space. To test our procedure in the *online* phase we select additional $N_{test} = 50$ values of the parameter vector and we compute the mean relative error (over the N_{test} trials) between the ROM and the FOM solution by varying the number n of basis functions for the solution and m for the nonlinear term (see right Figure 6). In this case we observe that the varying m dramatically affects the efficiency of the ROM, while varying n improves remarkably its accuracy, without affecting the overall efficiency. The need of a high-dimensional database of pre-computed solutions of the problem poses non negligible issues: the construction of the database and of POD-DEIM basis functions are extremely demanding both in terms of CPU time and memory storage (even in this simple case we cannot use a common laptop to perform the offline phase). Moreover, the CPU time required for the *online* solution increases, as a larger dimensional POD-DEIM approximation is considered. In the case where the POD-DEIM ROM is built on a snapshots matrix of dimension $N_h \times 300N_{train}$, with $N_{train} = 100$, we require $n = 160$ basis functions for u and $m = 220$ basis functions for I_{ion} to reach a relative error lower than 10^{-2} over the test sample.

As a matter of fact, the resulting global POD-DEIM ROM only entails a speedup of $5.6x$ with respect to the FOM: this is mainly due to the high dimensionality of the reduced mesh used to assemble the nonlinear term during the DEIM-procedure (see Figure 7). In fact, the parametrized ischemic region could be centered in all the different points of the domain, as shown in Figure 3. This clearly motivates the need of local reduced spaces to approximate both the solution and the nonlinear term within the ROM.

5.1.2. Local ROMs

Here, we present a detailed comparison of the different strategies described in Section 4:

1. *Time-based clustering*: the subdivision of the snapshots matrix \mathbf{S}_u of dimensions $N_s = 300N_{train}$, with $N_{train} = 100$, is obtained by considering as cluster indicator the time variable $t^{(\ell)}$. In this case, the cluster which generates the local ROM with the smallest dimension is the one corresponding to the first time window $[0, t^{(1)}]$: here the solution does not show a great variability since the initial impulse I_{app} is not parametrized. As a matter of fact, we end up with $n_1 = 50$ basis functions for the state solution and $m_1 = 17$ terms for the DEIM approximation of the nonlinear term, when $N_c = 6$ windows are considered. By increasing N_c , less basis functions are required in the initial windows ($n_1 = 43$ and $m_1 = 11$ for $N_c = 16$). Unfortunately, the subsequent time windows contain solutions showing much higher variability, since the wave fronts are modified by the possibly different locations of the ischemia. As a consequence, the DEIM approximation of the nonlinear term is no longer assembled on a small reduced mesh: for instance, in the case $N_c = 6$ we have $m_3 = 118$, $m_4 = 175$ and $m_5 = 162$ (see Figure 8). By increasing N_c , error propagation becomes more relevant, resulting from the approximation error arising from the change of local ROMs during the online simulation. To minimize this error propagation, it is possible to use overlapping windows by enriching consequently the dimensionality of the local ROMs. Despite these drawbacks, we obtain a speedup of $10.8x$ for $N_c = 15$, mainly due to the considerable computational savings obtained in the initial time windows (see Figure 9 for a comparison of the performances when different numbers N_c of clusters are considered).
2. *Parameter-based clustering*: in this case we partition the snapshots matrix by considering the vector of parameters $\boldsymbol{\mu}$ as cluster indicator. The results are similar to the ones obtained

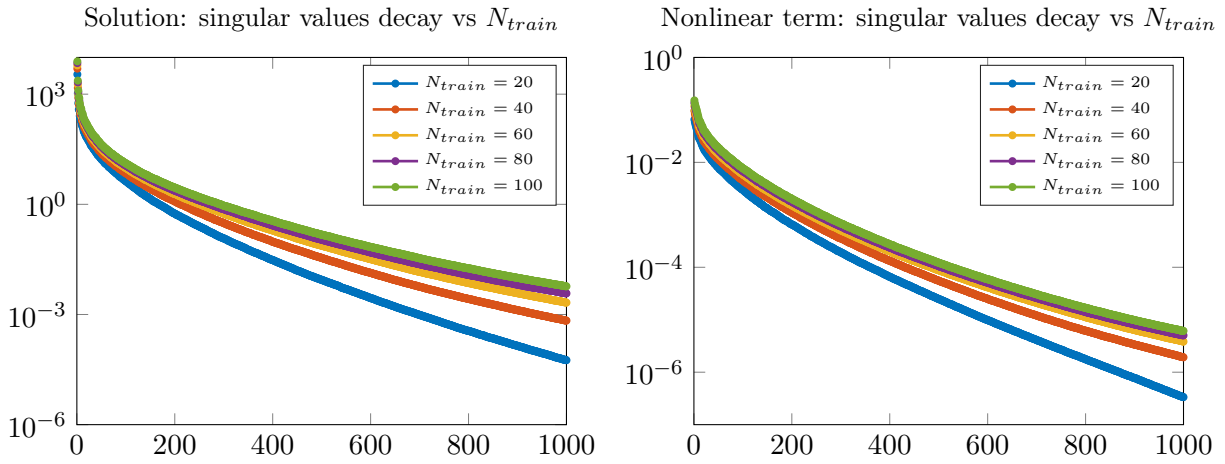


Figure 5: Decay of the singular values of the solution (left) and the nonlinear term (right). The decay changes considerably when additional snapshots are considered, confirming the great variability with respect to the parameters of this test case.

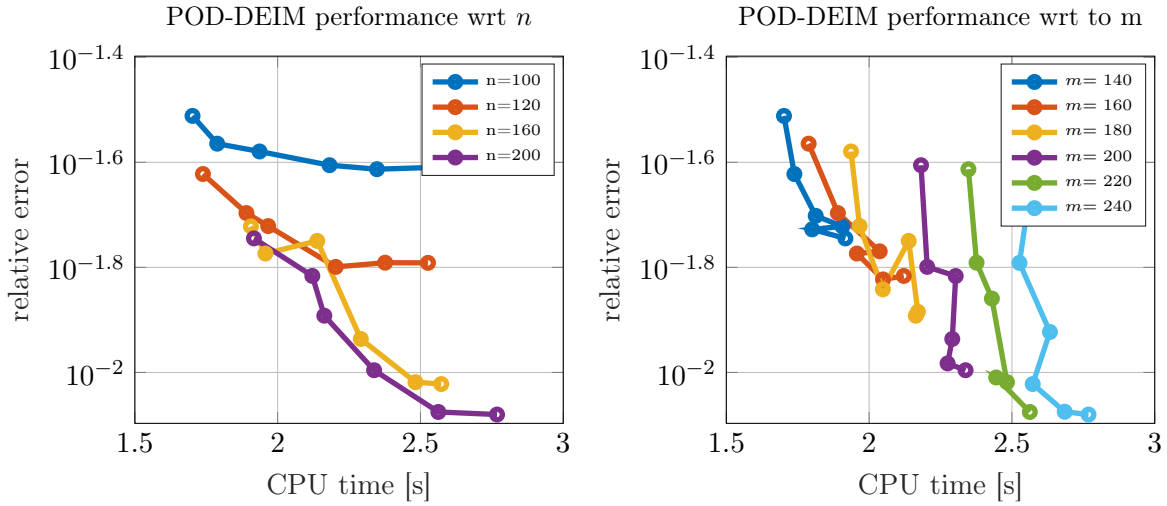


Figure 6: Relative mean H^1 -error over the *online* test sample vs the CPU time for different dimensions n and m of the ROM solution and the DEIM approximation of the nonlinear term.

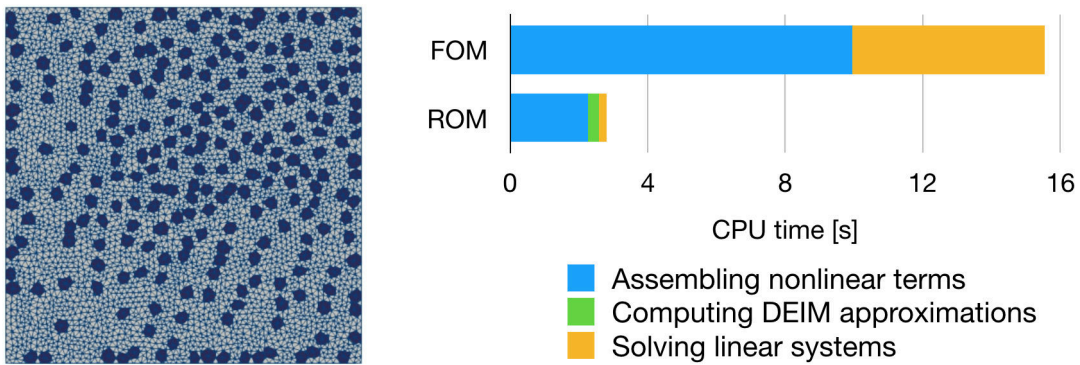


Figure 7: Left: sparse mesh employed by the POD-DEIM ROM based on a global reduced space. Right: comparison between CPU times required by the finite element FOM and the POD-DEIM ROM (*online*).

through the previous approach: we still find out a cluster with a large number of basis functions (see Figure 8, top) and the error evaluated on the test sample increases when a large number of clusters is considered (see Figure 9). This latter drawback is motivated by the fact that solutions corresponding to parameters which are on the boundary of a cluster \mathcal{P}_k might be poorly approximated by the local ROM. Also in this case, clusters overlapping could represent a possible way to fix the problem, sacrificing however the computational efficiency. Moreover, in this case we are not able to build a local ROM of substantially low-dimension: indeed, we end up respectively with a minimum number of basis functions for the DEIM approximation of 118 (46) when $N_c = 6$ (16) clusters are considered (see Figure 8, bottom) and we need to construct 16 local subspaces to reach a speedup of 9.1x. As a consequence, the computational resource that is mainly involved is memory, since all the RB matrices of dimensions $n_k \times n_k$ and $n_k \times m_k$ are not sparse. For this reason, parameter-base clustering seems not to be as promising as the previous one.

3. *State-based clustering*: we finally test the *k-means* and the *PEBL* algorithms for clustering the snapshots matrix \mathbf{S}_u . We highlight that both approaches lead to a smaller error on the test sample with respect to the time- and parameter-based local ROMs (see Figure 9); moreover, the state-based ROMs are less affected by error propagation.

The minimum (maximum) number of basis functions selected for a cluster provided by the *k-means* algorithm are always (often) smaller than the ones resulting from the other approaches (see Figure 8, bottom). In the case $N_c = 5$ the minimum number of basis functions for the solution is $n = 16$ for the *k-means* against $n = 55$ of the time-based, $n = 78$ of the *PEBL* and $n = 84$ of the parameter-based clustering; see Figure 8, top. As a consequence, the state-based approach employing the *k-means* algorithm is the one with the highest speedup (12.8x in the case $N_c = 16$) and the lowest memory storage (36.4 MB for $N_c = 10$). By looking at the centroids selected by the algorithm (see Figure 11), we notice that a primal subdivision is done with respect to time (case $N_c = \{4, 6, 8\}$). Nevertheless, this approach is more flexible with respect to the time-based one, because the front propagation velocity is taken into account automatically by the *k-means* algorithm since the cluster is assigned with respect to the current state. Also the pattern of the reduced mesh clearly reflects this subdivision for most of the clusters. Moreover, when considering a larger number of subdivisions, the variation of physical parameters starts playing an important role: for $N_c = 12$ we have two centroids that describe two situations where the ischemic region is not on the main diagonal.

The *PEBL* clustering does not provide the same performance of the *k-means* algorithm: the trade-off between accuracy, efficiency and memory storage is not as good as in the *k-means* case (see Table 1). As a matter of fact, the speedup ranges from 6.2x for $N_c = 2$ to 10x for $N_c = 16$, while memory storage goes from 22.9 MB for $N_c = 2$ to 43.8 MB for $N_c = 16$. By looking at the partitioning tree (see Figure 12), we observe in the case $N_c = 12$ that the subdivision is more focused on the location of the ischemia than on the time variable as done by the *k-means* clustering. The variability of the solutions related to the first 100 time-steps of the simulation is described only by one cluster (the fifth one), affecting in this way the overall performance. The speedup obtained by a faster online identification of the current cluster enabled by the tree structure is neutralized by the dimensions of the local ROM arrays. On the other hand, the remaining eleven clusters capture the variability of the solution induced by the different position of the ischemic region, leading to a more accurate ROM with respect to the previous approaches.

In conclusion, among the proposed local ROMs, the state-based strategy implemented in the

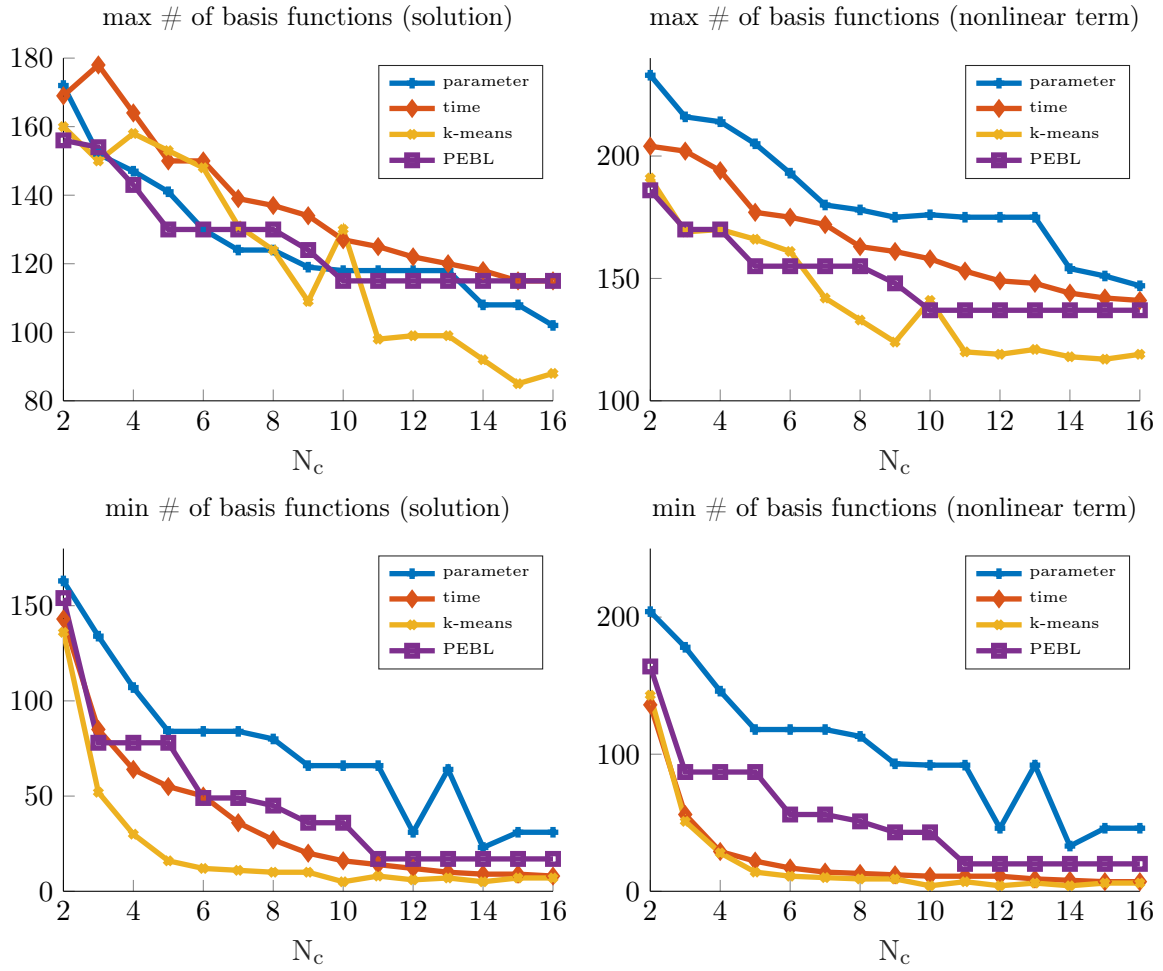


Figure 8: *Test case I*. Minimum and maximum number of basis functions for each local ROM.

Online performance

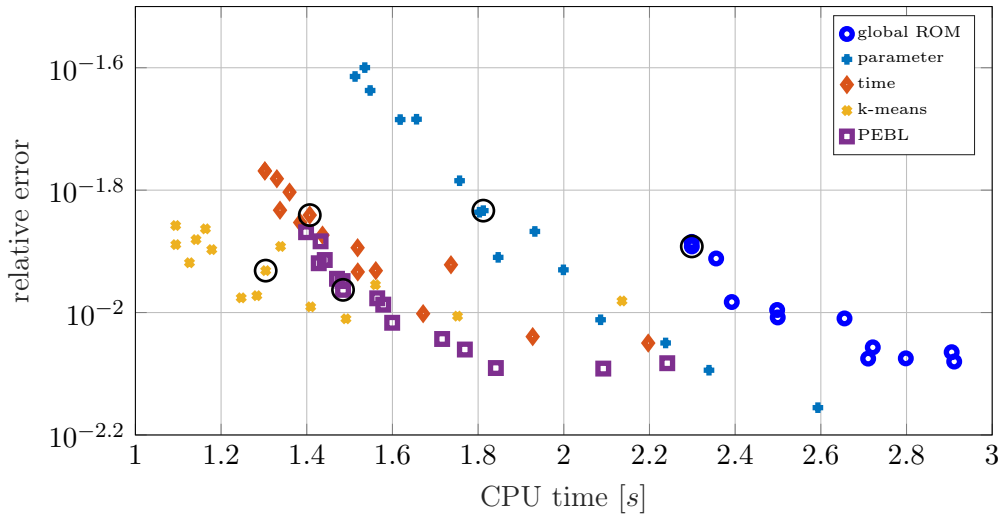


Figure 9: *Test case I*. Average *online* relative error over the test sample vs the CPU time for the different local ROM methods. The values in Table 1 are marked in this graph with black circles.

PEBL algorithm is the best option in terms of accuracy, but not in terms of CPU time and memory storage (see Table 1). This also has an impact on the evaluations of outputs of interest, such as the activation time, for which a full agreement between the FOM (see Figure 3) and the ROM (see Figure 10) quantities can be observed.

The state-based *k-means* ROM is overall the best option in the case at hand: the k-means algorithm provide a way to select automatically clusters of the snapshots set, for which the corresponding local ROMs feature a small number of basis functions. Moreover, the error propagation is minimized by the fact that there is a smooth transition from one cluster to another, and finally, it achieves the second best speedup among the considered schemes.

	speedup	mean relative error	memory storage
global POD-DEIM	6.1x	$1.3 \cdot 10^{-2}$	29.4 [MB]
time-based LROM ($N_c = 10$)	10.1x	$1.4 \cdot 10^{-2}$	47.6 [MB]
parameter-based LROM ($N_c = 10$)	8x	$1.64 \cdot 10^{-2}$	55.8 [MB]
<i>k-means</i> -based LROM ($N_c = 10$)	11.9x	$1.27 \cdot 10^{-2}$	36.4 [MB]
PEBL-based LROM ($N_c = 10$)	9.5x	$1.13 \cdot 10^{-2}$	39.2 [MB]

Table 1: *Test case I*. ROMs ranking. We highlight the performances using colors from red (worst) to green (best).

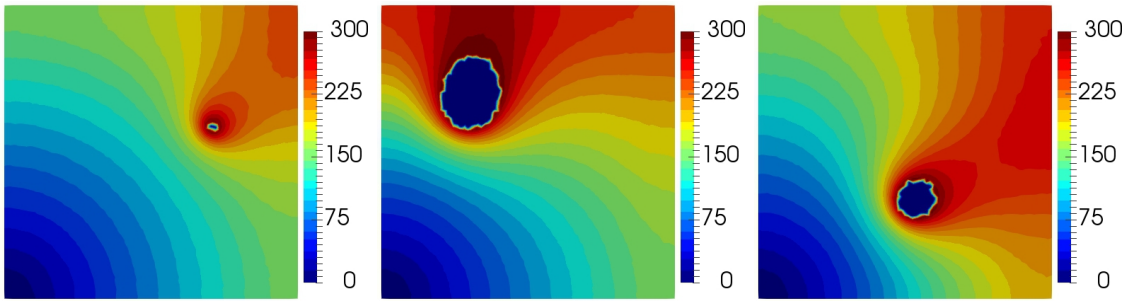


Figure 10: *Test case I*. Depolarization time [ms] computed with the *k-means* local ROM with $N_c = 10$ for the same parameter vectors of Figure 3.

5.2. Test case II: idealized left ventricle

In this second case we consider the monodomain model (1.1) with the Aliev-Panfilov model (2.1) to describe the electrical activation of an idealized left ventricle aiming at estimating the effect of fibers orientation on the electric signal conduction. As a matter of fact, the cardiac tissue is composed of fibers (the cardiomyocytes) whose orientation varies from the epicardium to the endocardium due to the laminar organization in sheets of the tissue [61]. At the macroscopic level, this structure yields preferential directions for the action potential traveling front [62, 7].

Therefore, at any point \mathbf{x} , it is possible to identify an orthonormal local reference system described by the principal axes $\mathbf{f}_0(\mathbf{x})$, $\mathbf{s}_0(\mathbf{x})$, and $\mathbf{n}_0(\mathbf{x})$, with $\mathbf{f}_0(\mathbf{x})$ parallel to the fiber direction, $\mathbf{s}_0(\mathbf{x})$ and $\mathbf{n}_0(\mathbf{x})$ orthogonal and tangent to the sheet direction. Denoting by σ_l and σ_t and σ_v the conductivity coefficients measured along the corresponding directions $\mathbf{f}_0(\mathbf{x})$, $\mathbf{s}_0(\mathbf{x})$, and $\mathbf{n}_0(\mathbf{x})$, the anisotropic conductivity tensor can be expressed as

$$\mathbf{D}(\mathbf{x}) = \sigma_l \mathbf{f}_0(\mathbf{x}) \otimes \mathbf{f}_0(\mathbf{x}) + \sigma_t \mathbf{s}_0(\mathbf{x}) \otimes \mathbf{s}_0(\mathbf{x}) + \sigma_v \mathbf{n}_0(\mathbf{x}) \otimes \mathbf{n}_0(\mathbf{x}).$$

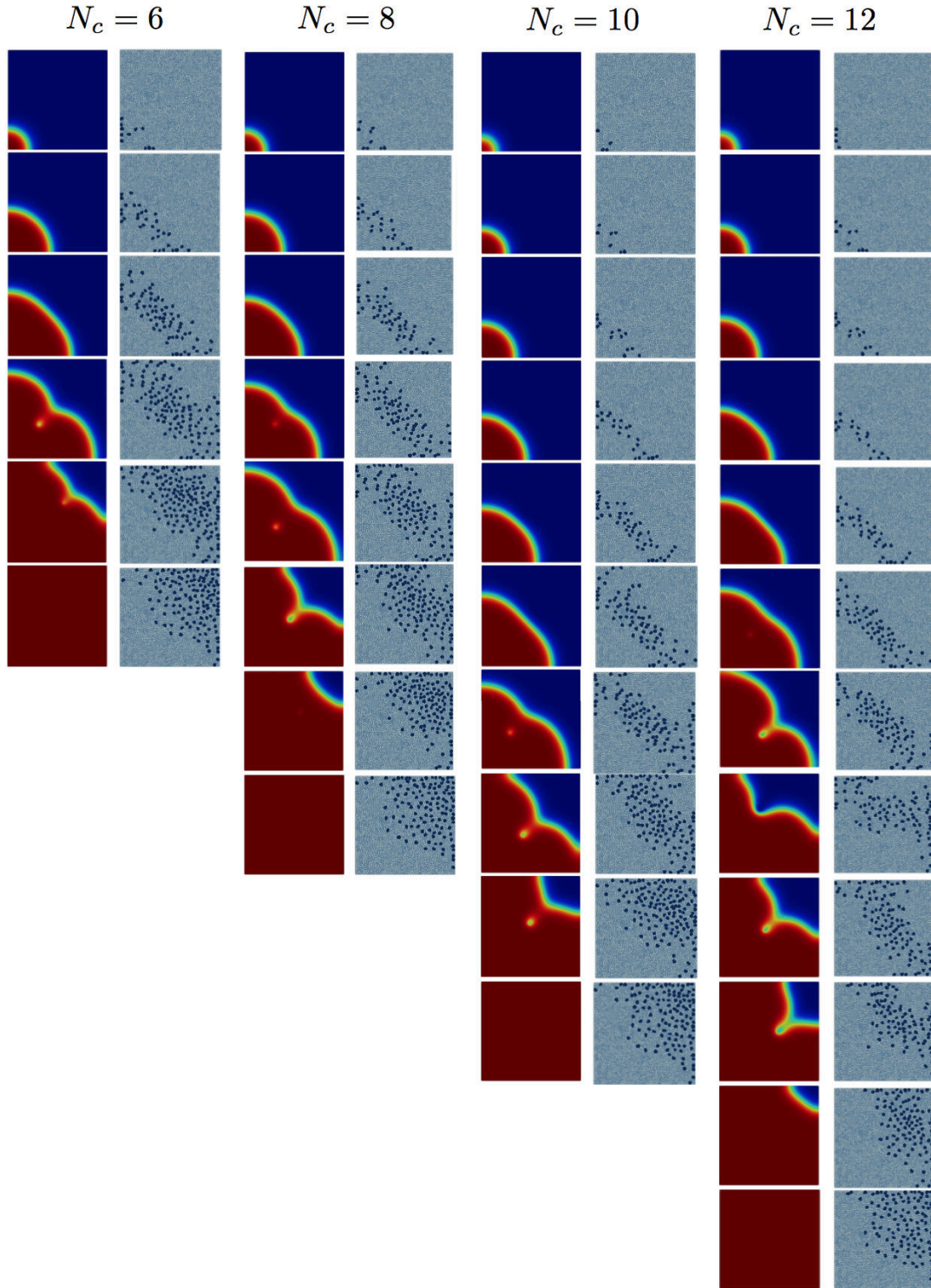


Figure 11: *Test case I.* Centroids obtained with the *k-means* algorithm applied to the snapshot matrix \mathbf{S}_u and associated reduced meshes for $N_c = 6, 8, 10, 12$ clusters (from left to right). We observe a perfect match between the outputs calculated with FOM and the local ROM.

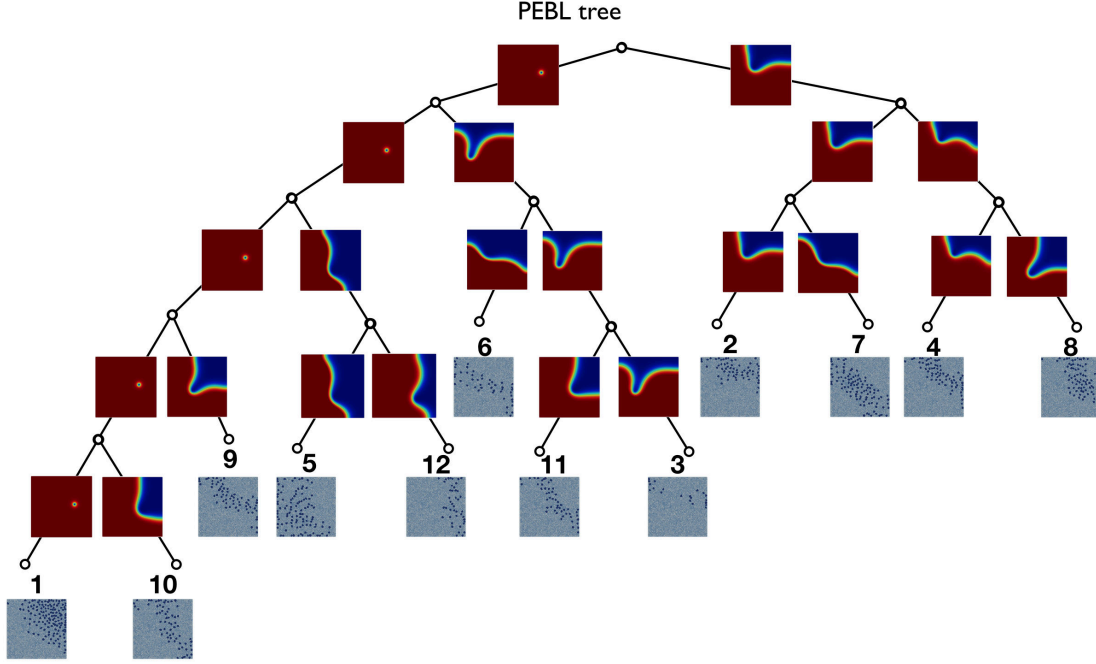


Figure 12: Snapshot subdivision tree related to the PEBL algorithm

Under the assumption that the left-ventricle tissue is an axisymmetric anisotropic medium ($\sigma_t = \sigma_v$), the previous relation simplifies as follows,

$$\mathbf{D}(\mathbf{x}) = \sigma_t \mathbf{I} + (\sigma_l - \sigma_t) \mathbf{f}_0(\mathbf{x}) \otimes \mathbf{f}_0(\mathbf{x}).$$

In this case we have considered an idealized left ventricle domain, whose inner and outer surface are described by the ellipsoids

$$\{\mathbf{x} = (x, y, z) \in \mathbb{R}^3 : x = r_i \cos(\xi) \cos(\omega), \quad y = r_i \cos(\xi) \sin(\omega), \quad z = r_i \sin(\xi), \quad i = 1, 2\},$$

where $r_1 = 2.7 \text{ cm}$ for the endocardium and $r_2 = 3.6 \text{ cm}$ for the epicardium. The final geometry is obtained by truncating the ellipsoid to have an apex-to-base distance of 7 cm . We then represent the sheets direction as

$$\mathbf{s}_0(\mathbf{x}) = \left(\frac{x}{\sqrt{x^2 + y^2}}, \frac{y}{\sqrt{x^2 + y^2}}, 0 \right),$$

while the fibers direction \mathbf{f}_0 is obtained imposing the orthogonality constraints to \mathbf{s}_0 in the planes ($\{\mathbf{x} \in \mathbb{R}^3 : z = c_z\}$), with c_z constant. Finally, $\mathbf{f}_0(\mathbf{x})$ is rotated from an angle θ_{epi} on the epicardium to an angle θ_{endo} on the endocardium with the following relationship:

$$\theta = (\theta_{epi} - \theta_{endo}) \frac{r - r_1}{r_2 - r_1} + \theta_{endo}.$$

We consider four parameters for this test case: the two conductivities $\sigma_l \in [0.1, 0.25]$ and $\sigma_t \in [0.01, 0.1]$ and the two angles $\theta_{epi} \in [30, 70]$ and $\theta_{endo} \in [-70, -30]$. The fiber angles determine the preferential directions for the signal propagation (see Figure 13), while the ratio between the values of the two conductivities is directly related to the anisotropic behaviour of the conduction (see Figure 14). The other physical parameters have been fixed, accordingly to [63], to $K = 8$, $a = 0.15$, $c_1 = 0.1$, $c_2 = 0.3$ and $\varepsilon_0 = 0.013$. The geometry has been discretized using a three-dimensional mesh with $N_h = 31765$ vertices and 140271 elements. In this example

the full-order numerical approximation, based on linear finite elements, takes 64 minutes to be computed. This CPU time is mainly motivated by the fact that the semi-implicit time-advancing scheme requires a large number of time-steps ($N_t = 1800$), each of which entails the assembling of the nonlinear term.

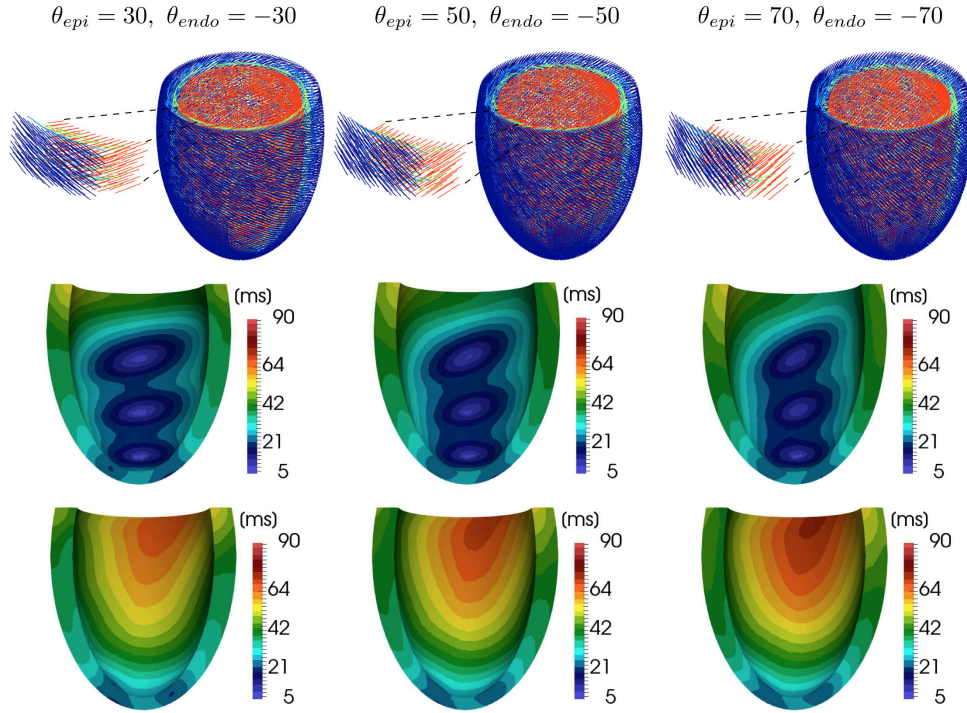


Figure 13: Fibers orientation for different values of θ_{epi} and θ_{endo} (first row) and activation times measured over the idealized left ventricle (second and third row). Here $\sigma_l = 0.25$ and $\sigma_t = 0.025$. The numerical approximation of the electric potential is performed using the full-order finite element model.

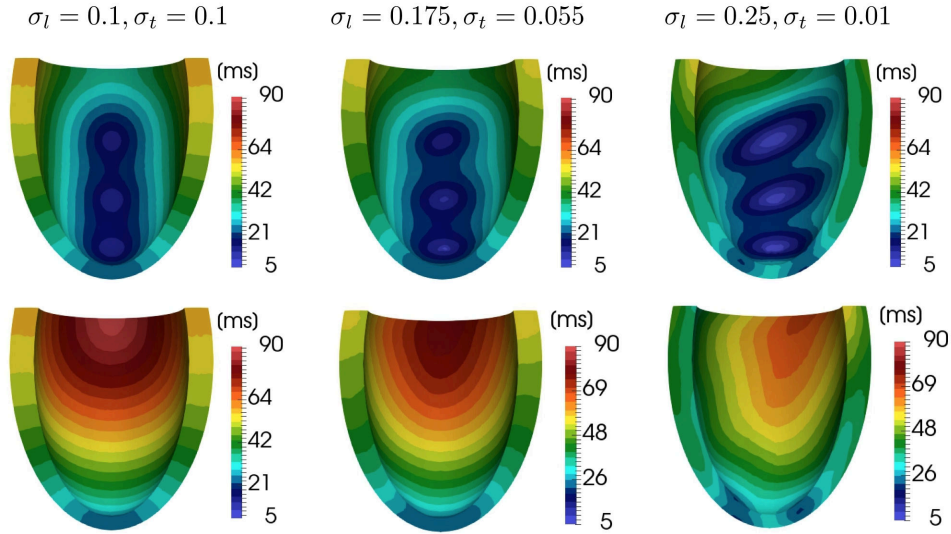


Figure 14: Activation map for different values of the conductivities σ_l and σ_t (here $\theta_{epi} = 50$ and $\theta_{endo} = -50$). The numerical approximation of the electric potential is performed using the full-order finite element model.

The complexity of the parametrized solution of the problem is increased with respect to the previous test case by the presence of the polarization front shaped by the recovery variable w (1.1). We consider a training set formed by $N_{train} = 50$ parameter vectors (the corresponding snapshots matrix \mathbf{S}_u is formed by $N_s = N_t N_{train}$ full-order vectors), while additional $N_{test} = 30$ values are used to test the *online* performances of the ROMs.

In the case of the POD-DEIM ROM, based on a unique global reduced space, we require $n = 275$ basis functions for the solution u and $m = 1956$ basis functions for the nonlinear term I_{ion} . This large value of m compromises the efficiency of the ROM: the reduced mesh does not present a sparse pattern, making the construction of DEIM nonlinear term very cumbersome; as a result, only a modest speedup of $3.4x$ is reached. Local ROMs are mandatory to improve the modest performance of such a global ROM.

First, we compare the local ROMs built by considering the four options described in Section 4 at fixed $N_c = 15$, by looking at the trade-off between accuracy, efficiency and memory storage (see Table 2). In this test case we observe that the parameter-based clustering yields the worst overall performance, with a maximum memory storage and the second largest speedup and relative error among all the strategies. In this case it is not possible to build a local ROM of substantially low-dimension due to the fact that for each new parameter we use a unique reduced subspace to approximate the evolution of the front. The *k-means* clustering leads instead to the best overall performance in terms of speedup ($35.5x$) and the second smallest relative error on the test sample ($9.6 \cdot 10^{-3}$). Also in this test case the centroids selected by the *k-means* algorithm (see Figure 15) are principally divided with respect to time (this also motivates the good performances of the time-based local ROMs). Time is pivotal also in constructing the partitioning tree for the PEBL clustering (see Figure 16, case $N_c = 15$), with 11 leafs related to the depolarization phase and only 4 to the polarization phase. This unbalanced subdivision enhances the performance of the local ROM in the depolarization phase, however by decreasing the efficiency of the numerical approximation in the polarization phase (with an overall performance lower than the one obtained with the *k-means* clustering).

	speedup	mean relative error	memory storage
global POD-DEIM	$3.4x$	$0.99 \cdot 10^{-2}$	108.1 [MB]
time-based LROM ($N_c = 15$)	$28.8x$	$1.86 \cdot 10^{-2}$	192.1 [MB]
parameter-based LROM ($N_c = 15$)	$21.2x$	$1.18 \cdot 10^{-2}$	224.8 [MB]
<i>k-means</i> -based LROM ($N_c = 15$)	$35.5x$	$0.96 \cdot 10^{-2}$	213.1 [MB]
PEBL-based LROM ($N_c = 15$)	$21.6x$	$0.81 \cdot 10^{-2}$	214.4 [MB]

Table 2: *Test case II*. ROMs ranking. We highlight the performances using colors from red (worst) to green (best).

Next, we consider the effect of changing the number of clusters N_c on the *online* performance. With this goal, we compare the mean relative error on the test sample and the CPU time obtained with the different reduction strategies described in Section 4 by using $N_c = 5, 10, 15, 20$ clusters (see Figure 17). The state-based clustering based on the *k-means* algorithm is the best in terms of the trade-off between efficiency and accuracy. State-based local ROMs show more robustness with respect to changes in the number of clusters: the relative errors on the test sample are comparable and the speedup increase progressively with N_c . Instead, note that for $N_c = 5$ the time- and parameter-based local ROMs exhibit a large CPU time of 342.7 and 384.7 seconds, respectively. These numerical results emphasize the fact that state-based local ROMs are better suited for clustering snapshots of time-dependent problems with parameter dependent moving

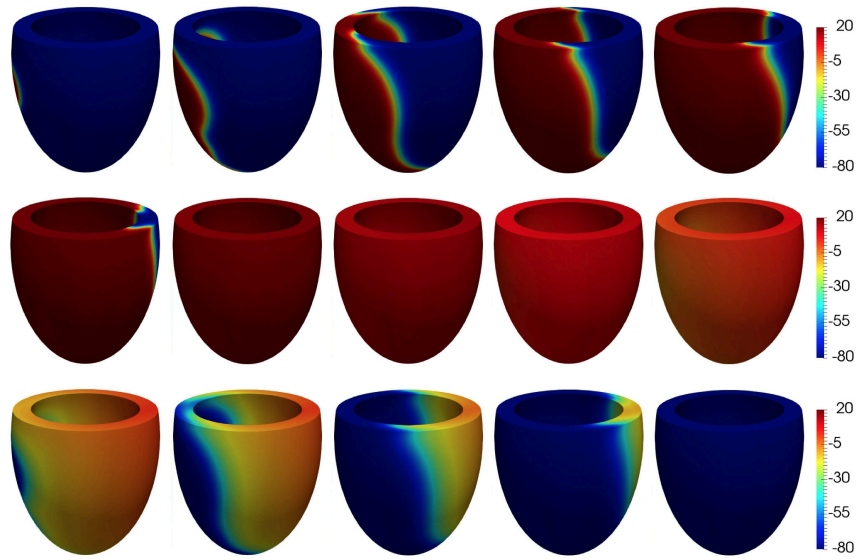


Figure 15: *Test case II*. Centroids obtained with the *k-means* algorithm applied to the snapshot matrix S_u

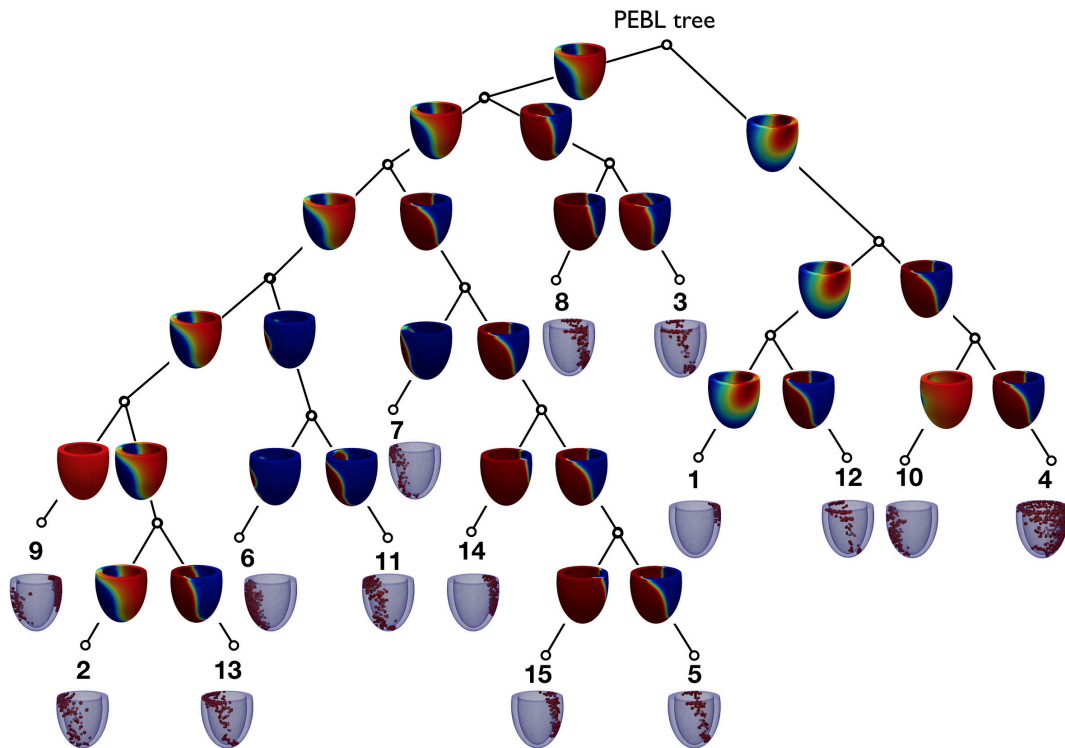


Figure 16: *Test case II*. Snapshot subdivision tree resulting from the PEBL algorithm

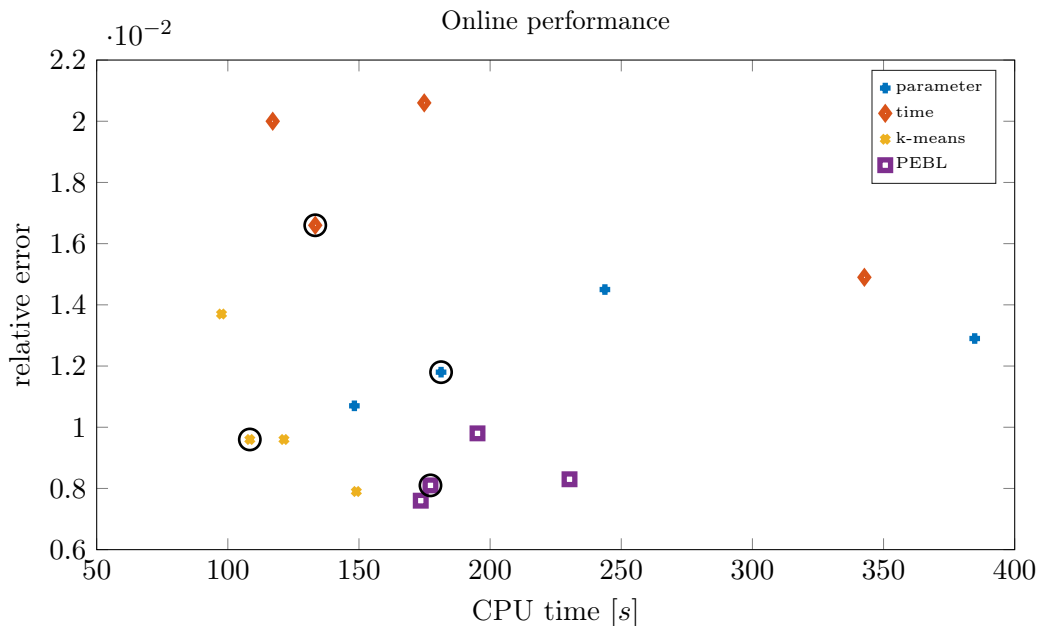


Figure 17: *Test case II*. Average *online* relative error over the test sample vs the CPU time for the different local ROM methods. The values in Table 2 are marked in this graph with black circles.

fronts.

Finally, we compute the depolarization time map for two parameter vectors in the test sample (see Figure 18) to compare the overall activation pattern generated by the finite element full-order model and the *k-means*-based local ROM with $N_c = 15$. The propagation of the signal along the idealized left ventricle appears similar between the two models: the depolarization time difference map is indeed smaller than 0.5 ms in every points of the computational mesh.

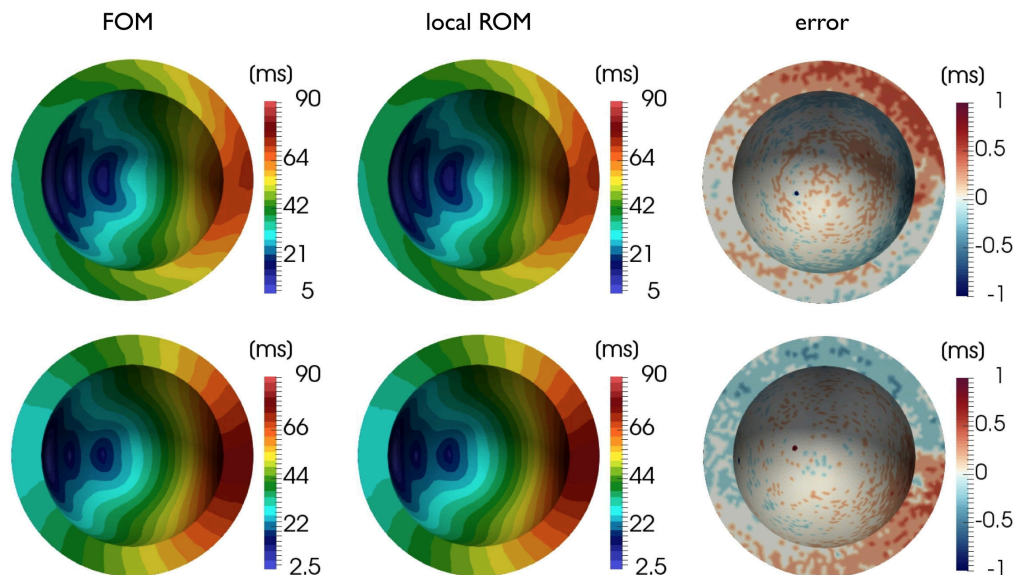


Figure 18: *Test case II*. Depolarization time [ms] computed with the finite element FOM (left) and the *k-means* local ROM with $N_c = 15$ (center) for the two parameter vectors from the test sample. Right: depolarization time difference map between the FOM and the *k-means* local ROM.

6. Conclusions

A detailed comparison of different local ROMs applied parameter-dependent problems arising in cardiac electrophysiology is presented in this work. Special emphasis has been put on the enhancement of the computational efficiency of local ROMs, built on clustering techniques for partitioning the snapshot matrix, and a POD-DEIM-Galerkin strategy for the construction of the reduced arrays. The *online* efficiency of local ROMs is ensured by the relatively small dimension of local reduced arrays and the ability to switch between local reduced spaces by evaluating only quantities which depend on the reduced dimension n and m of the POD state space and the DEIM approximation of nonlinear terms, respectively.

Our numerical tests show that the *k-means* clustering outperforms the other clustering approaches. In particular, the local ROM based on *k-means* clustering realizes the best speedup, improving the performance of a global ROM of a factor 2 in the first test case and 10 in the second one. Increasing the number of clusters has two competing effects: it can improve the speedup, but also lead to an increase of error propagation. This trade-off is clearly visible in the results related to parameter- and time-based local ROMs, while the state-based clustering techniques enable to better manage the error propagation drawback, justifying their better overall performance.

On the basis of the results on cardiac electrophysiology problems we expect local ROMs, and more in general nonlinear dimensionality reduction techniques, to be a viable strategy for efficiently solving nonlinear unsteady parametrized problems showing complex dynamics.

References

References

- [1] Quarteroni A, Manzoni A, Vergara C. The cardiovascular system: Mathematical modeling, numerical algorithms, clinical applications. *Acta Numerica* 2017; **26**:365–590.
- [2] FitzHugh R. Impulses and physiological states in theoretical models of nerve membrane. *Biophysical journal* 1961; **1**(6):445–466.
- [3] Nagumo J, Arimoto S, Yoshizawa S. An active pulse transmission line simulating nerve axon. *Proceedings of the IRE* 1962; **50**(10):2061–2070.
- [4] Aliev RR, Panfilov AV. A simple two-variable model of cardiac excitation. *Chaos Solitons Fractals* 1996; **7**(3):293–301.
- [5] Nash MP, Panfilov AV. Electromechanical model of excitable tissue to study reentrant cardiac arrhythmias. *Prog. Biophys. Mol. Biol.* 2004; **85**:501–522.
- [6] Mitchell CC, Schaeffer DG. A two-current model for the dynamics of cardiac membrane. *Bulletin of Mathematical Biology* 2003; **65**(5):767–793.
- [7] Colli Franzone P, Pavarino L, Taccardi B. Simulating patterns of excitation, repolarization and action potential duration with cardiac bidomain and monodomain models. *Mathematical Biosciences* 2005; **197**(1):35–66.
- [8] Plank G, Liebmann M, dos Santos RW, Vigmond EJ, Haase G. Algebraic multigrid preconditioner for the cardiac bidomain model. *IEEE Transactions on Biomedical Engineering* 2007; **54**(4):585–596.

- [9] Goktepe S, Kuhl E. Computational modeling of cardiac electrophysiology: a novel finite element approach. *Int. J. Numer. Methods Engng.* 2009; **79**(2):156–178.
- [10] Miller CE, Henriquez CS. Finite element analysis of bioelectric phenomena. *Critical Reviews in Biomedical Engineering* 1989; **18**(3):207–233.
- [11] Patelli AS, Dede' L, Lassila T, Bartzaghi A, Quarteroni A. Isogeometric approximation of cardiac electrophysiology models on surfaces: An accuracy study with application to the human left atrium. *Computer Methods in Applied Mechanics and Engineering* 2017; **317**:248–273.
- [12] Sundnes J, Lines GT, Cai X, Nielsen BF, Mardal KA, Tveito A. *Computing the electrical activity in the heart*, vol. 1. Springer Science & Business Media, 2007.
- [13] Vigmond EJ, Aguel F, Trayanova NA. Computational techniques for solving the bidomain equations in three dimensions. *IEEE Transactions on Biomedical Engineering* 2002; **49**(11):1260–1269.
- [14] Vigmond E, Dos Santos RW, Prassl A, Deo M, Plank G. Solvers for the cardiac bidomain equations. *Progress in Biophysics and Molecular Biology* 2008; **96**(1):3–18.
- [15] Colli Franzone P, Pavarino LF. A parallel solver for reaction–diffusion systems in computational electrocardiology. *Mathematical Models and Methods in Applied Sciences* 2004; **14**(06):883–911.
- [16] Colli Franzone P, Pavarino LF, Scacchi S. *Mathematical cardiac electrophysiology, MS&A*, vol. 13. Springer, 2014.
- [17] Geneser SE, Kirby RM, MacLeod RS. Application of stochastic finite element methods to study the sensitivity of ECG forward modeling to organ conductivity. *IEEE Transactions on Biomedical Engineering* 2008; **55**(1):31–40.
- [18] Swenson DJ, Geneser SE, Stinstra JG, Kirby RM, MacLeod RS. Cardiac position sensitivity study in the electrocardiographic forward problem using stochastic collocation and boundary element methods. *Annals of biomedical engineering* 2011; **39**(12):2900–2910.
- [19] Mirams G, Pathmanathan P, Gray R, Challenor P, Clayton R. Uncertainty and variability in computational and mathematical models of cardiac physiology. *J. Physiol.* 2016; **594.23**:6833–6847.
- [20] Johnstone RH, Chang ET, Bardenet R, de Boer TP, Gavaghan DJ, Pathmanathan P, Clayton RH, Mirams GR. Uncertainty and variability in models of the cardiac action potential: Can we build trustworthy models? *Journal of Molecular and Cellular Cardiology* 2016; **96**:49 – 62, doi:<http://dx.doi.org/10.1016/j.yjmcc.2015.11.018>. Special Issue: Computational Modelling of the Heart.
- [21] Hurtado D, Castro S, Madrid P. Uncertainty quantification of two models of cardiac electromechanics. *Int. J. Numer. Methods Biomed. Engrg.* 2017; doi:10.1002/cnm.2894. In press.
- [22] Boulakia M, Cazeau S, Fernández MA, Gerbeau JF, Zemzemi N. Mathematical modeling of electrocardiograms: a numerical study. *Annals of Biomedical Engineering* 2010; **38**(3):1071–1097.

- [23] Schenone E, Collin A, Gerbeau JF. Numerical simulation of electrocardiograms for full cardiac cycles in healthy and pathological conditions. *Int. J. Numer. Meth. Biomed. Engng.* 2016; **32**(5):27–44, doi:10.1002/cnm.2744.
- [24] Boulakia M, Schenone E, Gerbeau JF. Reduced-order modeling for cardiac electrophysiology. application to parameter identification. *Int. J. Numer. Meth. Biomed. Engng.* 2012; **28**(6-7):727–744.
- [25] Camara O, Sermesant M, Lamata P, Wang L, Pop M, Relan J, Craene MD, Delingette H, Liu H, Niederer S, *et al.*. Inter-model consistency and complementarity: Learning from ex-vivo imaging and electrophysiological data towards an integrated understanding of cardiac physiology. *Progress in Biophysics and Molecular Biology* 2011; **107**(1):122 – 133, doi:http://dx.doi.org/10.1016/j.pbiomolbio.2011.07.007.
- [26] Quarteroni A, Manzoni A, Negri F. *Reduced Basis Methods for Partial Differential Equations. An Introduction.* La Matematica per il 3+2, Springer International Publishing, 2016.
- [27] Barrault M, Maday Y, Nguyen NC, Patera AT. An empirical interpolation method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathematique* 2004; **339**(9):667–672.
- [28] Chaturantabut S, Sorensen DC. Nonlinear model reduction via discrete empirical interpolation. *SIAM J. Sci. Comp.* 2010; **32**(5):2737–2764.
- [29] Carlberg K, Farhat C, Cortial J, Amsallem D. The GNAT method for nonlinear model reduction: effective implementation and application to computational fluid dynamics and turbulent flows. *J. Comput. Phys.* 2013; **242**:623–647.
- [30] Negri F, Manzoni A, Amsallem D. Efficient model reduction of parametrized systems by matrix discrete empirical interpolation. *J. Comput. Phys.* 2015; **303**:431–454.
- [31] Yang H, Veneziani A. Efficient estimation of cardiac conductivities via POD-DEIM model order reduction. *Applied Numerical Mathematics* 2017; **115**:180–199.
- [32] Gerbeau JF, Lombardi D. Approximated Lax pairs for the reduced order integration of nonlinear evolution equations. *J. Comput. Phys.* 2014; **265**:246–269.
- [33] Gerbeau JF, Lombardi D, Schenone E. Reduced order model in cardiac electrophysiology with approximated Lax pairs. *Adv. Comput. Math.* 2015; **41**(5):1103–1130.
- [34] Chapelle D, Gariah A, Sainte-Marie J. Galerkin approximation with proper orthogonal decomposition: new error estimates and illustrative examples. *ESAIM: Mathematical Modelling and Numerical Analysis* 2012; **46**(4):731–757.
- [35] Corrado C, Lassoued J, Mahjoub M, Zemzemi N. Stability analysis of the POD reduced order method for solving the bidomain model in cardiac electrophysiology. *Mathematical biosciences* 2016; **272**:81–91.
- [36] Henriquez CS. Simulating the electrical behavior of cardiac tissue using the bidomain model. *Critical Reviews in Biomedical Engineering* 1992; **21**(1):1–77.

- [37] Clayton R, Bernus O, Cherry E, Dierckx H, Fenton F, Mirabella L, Panfilov A, Sachse F, Seemann G, Zhang H. Models of cardiac tissue electrophysiology: Progress, challenges and open questions. *Progress in Biophysics and Molecular Biology* 2011; **104**(1):22–48. Cardiac Physiome project: Mathematical and Modelling Foundations.
- [38] Franzone PC, Savaré G. Degenerate evolution systems modeling the cardiac electric field at micro-and macroscopic level. *Evolution equations, semigroups and functional analysis*, Lorenzi A, Ruf B (eds.). Springer, 2002; 49–78.
- [39] Bourgault Y, Coudiere Y, Pierre C. Existence and uniqueness of the solution for the bidomain model used in cardiac electrophysiology. *Nonlinear analysis: Real world applications* 2009; **10**(1):458–482.
- [40] Bendahmane M, Karlsen K. Analysis of a class of degenerate reaction-diffusion systems and the bidomain model of cardiac tissue. *Networks and Heterogeneous Media* 2006; **1**(1):185–218.
- [41] Boulakia M, Fernández MA, Gerbeau JF, Zenzemi N. A coupled system of PDEs and ODEs arising in electrocardiograms modeling. *Applied Mathematics Research eXpress* 2008; .
- [42] Ethier M, Bourgault Y. Semi-implicit time-discretization schemes for the bidomain model. *SIAM Journal on Numerical Analysis* 2008; **46**(5):2443–2468.
- [43] Maday Y, Nguyen NC, Patera AT, Pau SH. A general multipurpose interpolation procedure: the magic points. *Communications on Pure and Applied Analysis* 2009; **8**(1):383–404.
- [44] Bonomi D, Manzoni A, Quarteroni A. A matrix deim technique for model reduction of nonlinear parametrized problems in cardiac mechanics. *Comput. Methods Appl. Mech. Engrg.* 2017; (324):300–326.
- [45] Amsallem D, Zahr MJ, Farhat C. Nonlinear model order reduction based on local reduced-order bases. *Int. J. Numer. Methods Engrg.* 2012; **92**(10):891–916.
- [46] Amsallem D, Haasdonk B. PEBL-ROM: Projection-error based local reduced-order models. *Advanced Modeling and Simulation in Engineering Sciences* 2016; **3**(1):6.
- [47] Washabaugh K, Amsallem D, Zahr M, Farhat C. Nonlinear model reduction for cfd problems using local reduced order bases. *AIAA Paper 2012–2686* 2012; :1–16.
- [48] Amsallem D, Zahr MJ, Washabaugh K. Fast local reduced basis updates for the efficient reduction of nonlinear systems with hyper-reduction. *Adv. Comput. Math.* 2015; **41**(5):1187–1230.
- [49] Peherstorfer B, Butnaru D, Willcox K, Bungartz HJ. Localized discrete empirical interpolation method. *SIAM Journal on Scientific Computing* 2014; **36**(1):168–192.
- [50] Dihlmann M, Drohmann M, Haasdonk B. Model reduction of parametrized evolution problems using the reduced basis method with adaptive time-partitioning. *Proceedings of ADMOS 2011, V International Conference on Adaptive Modeling and Simulation*, 2011.
- [51] Drohmann M, Haasdonk B, Ohlberger M. Adaptive reduced basis methods for non-linear convection-diffusion equations. *Springer Proceedings in Mathematics*, vol. 4, 2011; 369–377.
- [52] Borggaard J, Wang Z, Zietsman L. A goal-oriented reduced-order modeling approach for nonlinear systems. *Computers & Mathematics with Applications* 2016; **71**(11):2155–2169.

- [53] Haasdonk B, Dihlmann M, Ohlberger M. A training set and multiple bases generation approach for parameterized model reduction based on adaptive grids in parameter space. *Mathematical and Computer Modelling of Dynamical Systems* 2011; **17**(4):423–442.
- [54] Eftang JL, Patera AT, Rønquist EM. An “hp” certified reduced basis method for parametrized elliptic partial differential equations. *SIAM J. Sci. Comput.* 2010; **32**(6):3170–3200.
- [55] Eftang JL, Knezevic DJ, Patera AT. An “hp” certified reduced basis method for parametrized parabolic partial differential equations. *Math. Comput. Model. Dynam. Syst.* 2011; **17**(4):395–422.
- [56] Eftang JL, Stamm B. Parameter multi-domain *hp* empirical interpolation. *Int. J. Numer. Methods Engrg.* 2012; **90**:412–428.
- [57] Maday Y, Stamm B. Locally adaptive greedy approximations for anisotropic parameter reduced basis spaces. *SIAM J. Sci. Comput.* 2013; **35**.
- [58] Xu R, Wunsch D. Survey of clustering algorithms. *IEEE Transactions on Neural Networks* 2005; **16**(3):645–678.
- [59] Friedman J, Hastie T, Tibshirani R. *The elements of statistical learning*, vol. 1. Springer series in statistics Springer, Berlin, 2001.
- [60] Negri F. redbKIT Version 1.0. <http://redbkit.github.io/redbKIT/> 2016.
- [61] Sengupta PP, Korinek J, Belohlavek M, Narula J, Vannan MA, Jahangir A, Khandheria BK. Left ventricular structure and function: basic science for cardiac imaging. *Journal of the American College of Cardiology* 2006; **48**(10):1988–2001.
- [62] Knisley SB, Trayanova N, Aguel F. Roles of electric field and fiber structure in cardiac electric stimulation. *Biophysical journal* 1999; **77**(3):1404–1417.
- [63] Aliev RR, Panfilov AV. A simple two-variable model of cardiac excitation. *Chaos, Solitons & Fractals* 1996; **7**(3):293–301.

7. Appendix

In this Appendix we provide further numerical results⁵ on the comparison of the proposed techniques on a simple one-dimensional parameter-dependent problem, whose solutions show moving sharp fronts. We aim at better investigating accuracy and efficiency of the methodologies addressed in the paper.

⁵For the sake of reproducibility of the proposed results, the `Matlab` code used to generate these results is the package `LocalROM` has been released and is freely available at <https://stefanopagani.github.io/LocalROM/>.

We consider the following coupled PDE-ODE model set on a one-dimensional spatial domain $(0, L)$, with $L = 1$:

$$\left\{ \begin{array}{ll} \varepsilon \frac{\partial v}{\partial t} - \varepsilon^2 \frac{\partial^2 v}{\partial x^2} + f(v) + w = 0, & x \in (0, L), t \in (0, 2) \\ \frac{\partial w}{\partial t} + (\gamma w - bv) = 0, & x \in (0, L), t \in (0, 2) \\ \frac{\partial v}{\partial t}(0, t) = -i_0(t) = 50000t^3 e^{-15t}, & t \in (0, 2) \\ \frac{\partial v}{\partial x}(L, t) = 0, & t \in (0, 2) \\ \frac{\partial v}{\partial x}(x, 0) = 0, \quad w(x, 0) = 0, & x \in (0, L). \end{array} \right. \quad (7.1)$$

Here

$$f(v) = v(v - 0.1)(v - 1), \quad \gamma = 2, \quad b = 0.5$$

whereas

$$\mu = \varepsilon \in (0.005, 0.05)$$

is a parameter which can be varied. This system represents a parametrized version of the Fitzhugh-Nagumo membrane model, in which the ionic current is a cubic function in v and is linear in the recovery variable w . Such a model is widely used to describe excitation-relaxation of a cardiac cell, as well as neurons, and can be derived from a simplified electrical circuit model of the cell membrane (see, e.g., [16, Sect. 2.9.8], [2, 3]); in this respect, v plays the role of electrical potential. Problem (7.1) has been discretized in space using linear finite elements on a partition of $N_h = 1024$ elements of the domain $(0, 1)$. Regarding the treatment of nonlinear terms and time discretization, we use a semi-implicit, first order, one-step scheme, similar to the one discussed in Sect. 2. The solutions of the system above are characterized by parameter-dependent traveling waves which exhibit sharp fronts (see Figure 19). Note that for very small values of ε the solutions to the system can be seen as an approximation to the one of the corresponding purely hyperbolic system, in the vanishing viscosity limit.

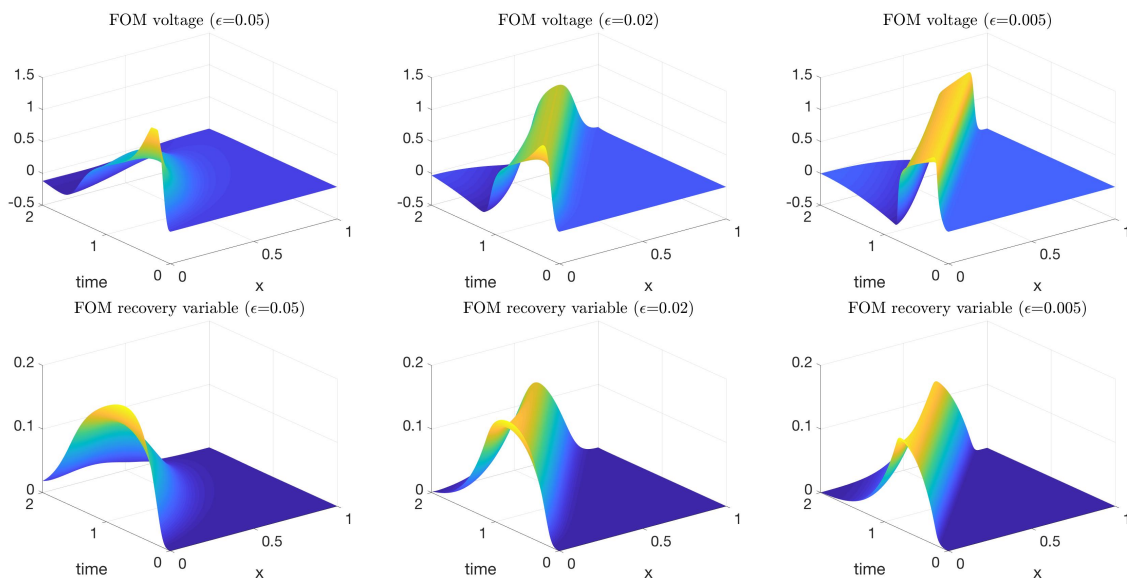


Figure 19: *Test case 1D*. Full-order model solution for different values of the parameter ε

We have compared the proposed techniques in terms of error decay with respect to the maximum number of basis functions in each cluster, also in presence of different levels of hyper-reduction. Numerical results show that also in this case state-based clustering methods, such as k -means and PEBL clustering, yield a remarkable error reduction when the number of clusters N_c increases. As shown in Fig. 20, state-based clustering techniques allow to reduce of about three times the number of basis functions in each cluster, decreasing from 246 (in the case of a global basis) to about 90 (when $N_c = 10$ clusters are employed) and an accuracy of 10^{-5} on the state solution is achieved. From the convergence analysis, it clearly results that more remarkable gains in terms of basis dimension are obtained when reaching smaller errors.

The time-based clustering also shows a good performance in this case, due to the substantial time-driven nature of the problem; a reduction, similar to the one provided by state-based clustering techniques, is obtained regarding the basis dimensions. On the other hand, the parameter-based clustering does not provide any computational gain when increasing the number of clusters, because of the mild parameter dependence of the problem at hand.

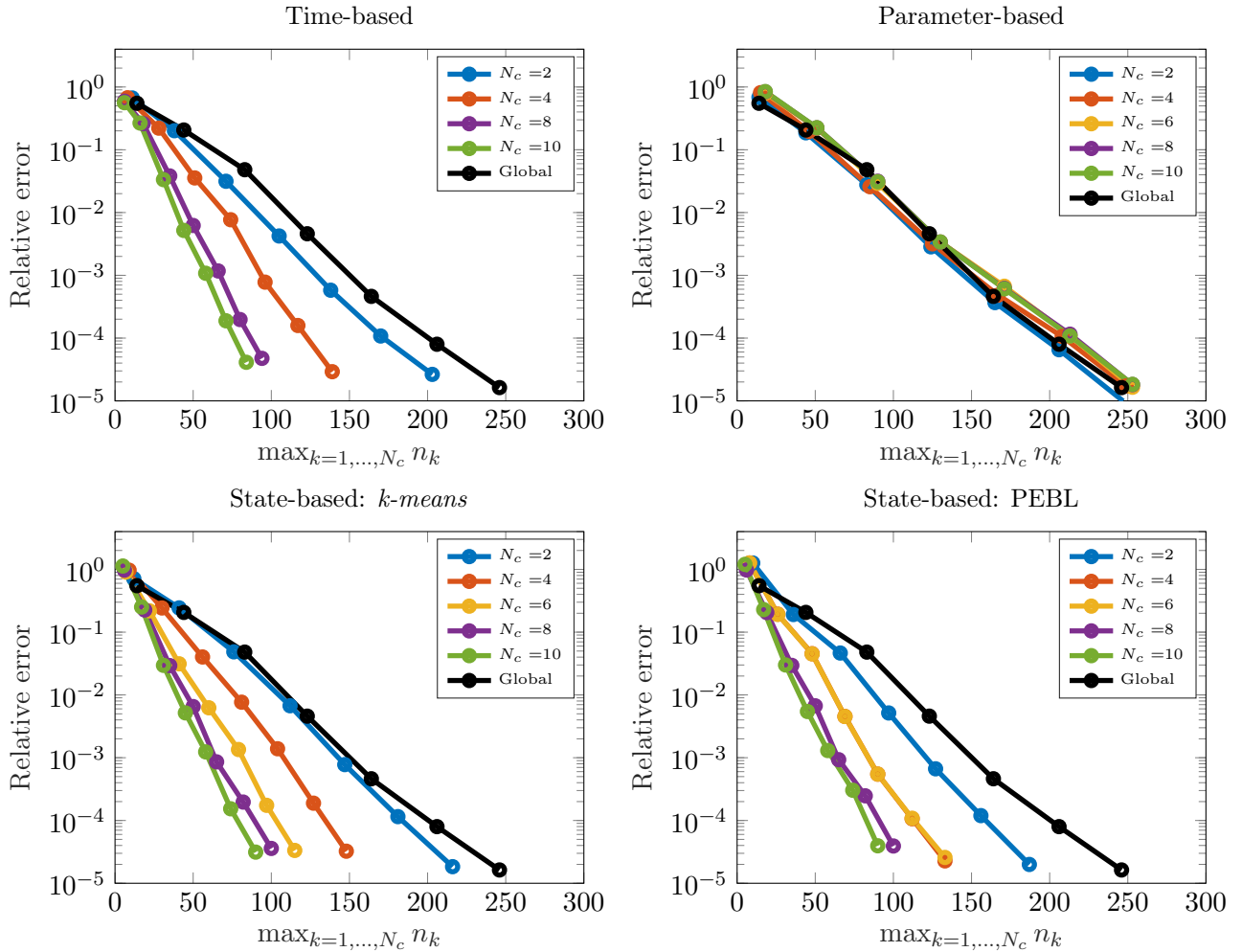


Figure 20: *Test case 1D*. H^1 -error decay with respect to the maximum number of basis functions among all clusters.

A further aspect taken into consideration deals with the effect of hyper-reduction on the error decay. Indeed, it is well-known in the framework of ROM techniques for (nonaffine and/or nonlinear) parameter-dependent PDE systems that a large number of (M)DEIM terms and RB functions has to be considered to achieve a good accuracy with respect to the high-fidelity approximation. For the case at hand, we apply DEIM to approximate the nonlinear term $f(v)$.

At some extent, hyper-reduction requires an even higher level of accuracy compared to state reduction. This is evident from Fig. 21, showing that DEIM must be performed by selecting basis functions (and corresponding interpolation points in the physical domain) until the approximation error on $f(v)$ reaches about 10^{-5} , if we pursue a decay of reduction errors which is not hampered by the further level of approximation introduced by DEIM. It is also evident that this fact is independent of the clustering technique used to perform both state reduction and hyper-reduction of nonlinear terms – we recall that the same clustering technique employed on solutions’ snapshots (before performing POD) is then used on snapshots of nonlinear terms before performing DEIM as well.

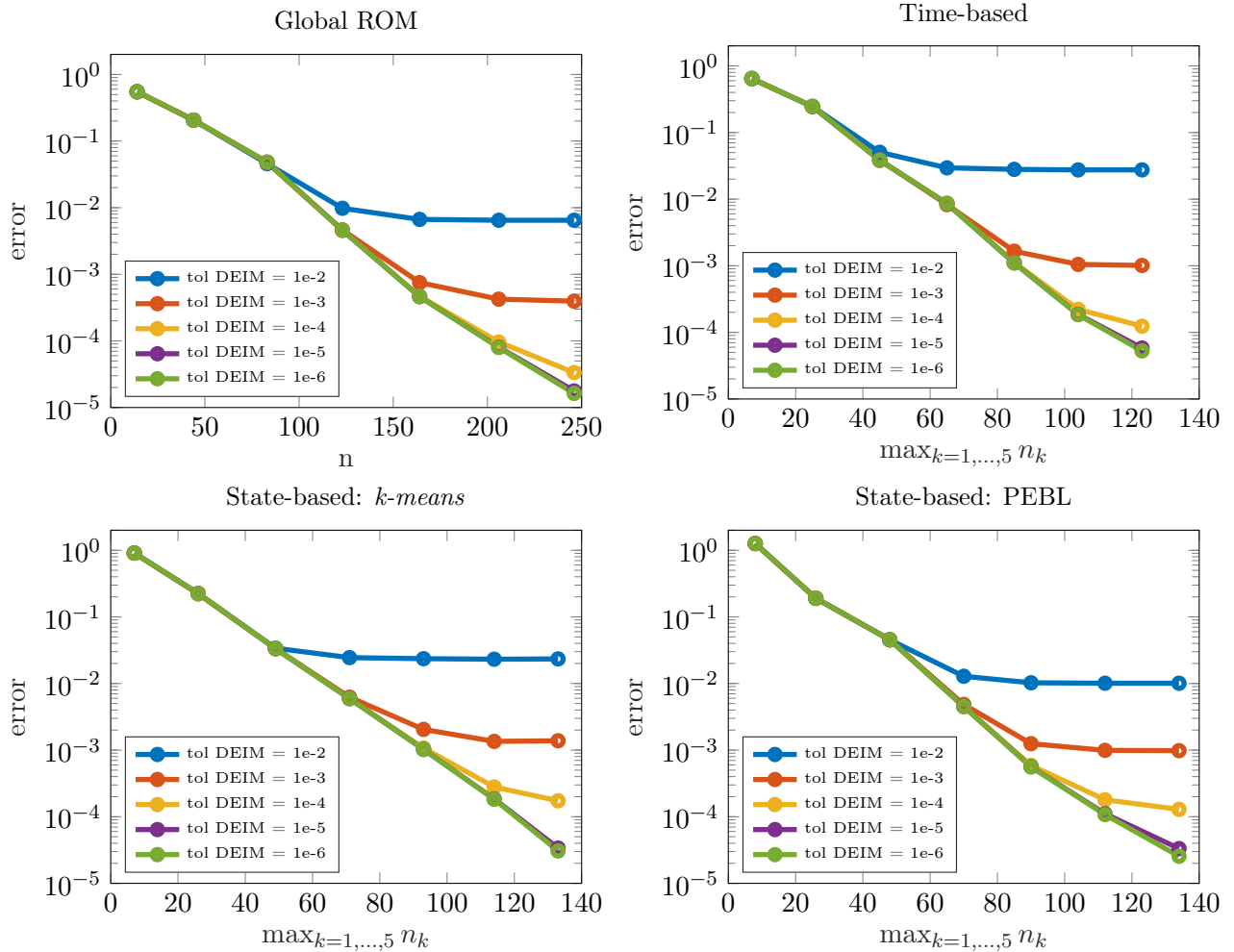


Figure 21: *Test case 1D*. H^1 -error decay with respect to the maximum number of basis functions among all clusters on varying the POD tolerance for the DEIM approximation of the non-linear term.

MOX Technical Reports, last issues

Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

- 03/2018** Antonietti, P. F.; Houston, P.; Pennesi, G.
Fast numerical integration on polytopic meshes with applications to discontinuous Galerkin finite element methods
- 04/2018** Ekin, T.; Ieva, F.; Ruggeri, F.; Soyer, R.
Statistical Medical Fraud Assessment: Exposition to an Emerging Field
- 02/2018** Canuto, C.; Nochetto, R. H.; Stevenson, R.; Verani, M.
A saturation property for the spectral-Galerkin approximation of a Dirichlet problem in a square
- 01/2018** Berrone, S.; Bonito, A.; Stevenson, R.; Verani, M.
An optimal adaptive Fictitious Domain Method
- 67/2017** Esterhazy, S.; Schneider, F.; Mazzieri, I; Bokelmann, G.
Insights into the modeling of seismic waves for the detection of underground cavities
- 68/2017** Paolucci, R.; Infantino, M.; Mazzieri, I; Özcebe, A.G.; Smerzini, C.; Stupazzini, M.
3D physics-based numerical simulations: advantages and current limitations of a new frontier to earthquake ground motion prediction. The Istanbul case study.
- 66/2017** Tamellini, M.; Parolini, N.; Verani, M.
An optimal control problem of two-phase compressible-incompressible flows
- 64/2017** Fumagalli, I.; Parolini, N.; Verani, M.
Optimal control in ink-jet printing via instantaneous control
- 65/2017** Regazzoni, F.;Parolini, N.;Verani M.
Topology optimization of multiple anisotropic materials, with application to self-assembling diblock copolymers
- 62/2017** Barbarotta, L.; Rossi, S.; Dede', L.; Quarteroni, A.
A Transmurally Heterogeneous Orthotropic Activation Model for Ventricular Contraction and its Numerical Validation