A cardiac electromechanics model coupled with a lumped parameters model for closed-loop blood circulation. Part II: numerical approximation

Regazzoni, F.; Salvador, M.; Africa, P.c.; Fedele, M.; Dede', L.;

Quarteroni, A.

A cardiac electromechanics model coupled with a lumped parameters model for closed-loop blood circulation. Part II: numerical approximation

Francesco Regazzoni[1,*], Matteo Salvador[1,*], Pasquale Claudio Africa[1], Marco Fedele[1],
Luca Dedè[1], Alfio Quarteroni[1,2]

[1] MOX-Dipartimento di Matematica, Politecnico di Milano, Milan, Italy
[2] Professor Emeritus, École polytechnique fédérale de Lausanne, Lausanne, Switzerland

∗ These authors equally contributed to this work

Abstract

In the framework of accurate and efficient segregated schemes for 3D cardiac electromechanics and 0D cardiovascular models, we propose here a novel numerical approach to address the coupled 3D-0D problem introduced in Part I of this two-part series of papers. We combine implicit-explicit schemes to solve the different cardiac models in a multiphysics setting. We properly separate and manage the different time and space scales related to cardiac electromechanics and blood circulation. We employ a flexible and scalable intergrid transfer operator that enables to interpolate Finite Element functions among different meshes and, possibly, among different Finite Element spaces. We propose a numerical method to couple the 3D electromechanical model and the 0D circulation model in a numerically stable manner within a fully segregated fashion. No adaptations are required through the different phases of the heartbeat. We also propose a robust algorithm to reconstruct the stress-free reference configuration. Due to the computational cost associated with the numerical solution of this inverse problem, the reference configuration recovery algorithm comes along with a novel projection technique to precisely recover the unloaded geometry from a coarser representation of the computational domain. We show the convergence property of our numerical schemes by performing an accuracy study through grid refinement. To prove the biophysical accuracy of our computational model, we also address different scenarios of clinical interest in our numerical simulations by varying preload, afterload and contractility. Indeed, we simulate physiologically relevant behaviors and we reproduce meaningful results in the context of cardiac function.

Keywords Mathematical modeling, Cardiac electromechanics, Multiscale models, Multiphysics model, Energy balance

1 Introduction

This paper proposes novel numerical approaches for the simulation of cardiac electromechanics [12, 16, 20, 35, 44, 54, 58]. Following [49], we propose the numerical approximation and simulation of the 3D-0D model provided in (1). The computational model for cardiac electromechanics presented in this second part relies on segregated schemes [17, 19, 21]. These schemes, also called partitioned, often suffer from instabilities issues [19, 39, 43, 50], despite being computationally more attractive

than monolithic schemes [21, 22, 59]. Nonetheless, our numerical approach enable to couple the core models describing the different physics in a numerically stable manner, yet allowing to adopt different resolutions in space and time for each one, to reflect their characteristic scales [9, 20, 35, 44, 52, 56]. With this aim, we implement an efficient and flexible intergrid transfer operator [1, 23, 52] that allows to couple models defined on different computational meshes and possibly different Finite Element spaces. The single core models are approximated by suitable implicit-explicit (IMEX) schemes [42] in time, aimed at lowering the computational effort associated with each time step without introducing severe CFL restrictions on the time step size otherwise associated to fully explicit schemes [45]. The computational framework presented in this paper realizes a very favorable trade-off between the biophysical detail of the underlying mathematical models and computational efficiency of the corresponding solver. Mathematical rigor, model accuracy and computational efficiency are the landmarks of the proposed electromechanical solver.

We also propose a novel numerical approach to couple the 3D and the 0D models [8, 28] in the segregated scheme. We reinterpret the cavity pressure of the left ventricle as a Lagrange multiplier to enforce a volumetric constraint to couple together the 0D circulation model and the 3D electromechanics model. We end up with a saddle-point structure for the mechanical problem, which is numerically solved using the Schur complement reduction [7]. Our scheme is numerically stable and can be applied to all the phases of the heart cycle (filling phase, ejection phase and isovolumetric contraction and relaxation phases) without the need of switching among different 0D models and without considering any change in the parameters of the equations [34, 52].

Our model is supported by a strategy to reconstruct the stress-free (i.e. unloaded) configuration of a ventricle, starting from the loaded geometry acquired by cardiac imaging, by solving a suitable inverse problem [29]. This is essential for physically consistent patient-specific simulations since the elastodynamics equations must refer to such stress-free reference configuration. The numerical resolution of this inverse problem can be computationally demanding for highly refined meshes. For this reason, our algorithm for the recovery of the reference configuration is supplied by a projection technique that accurately recover the stress-free configuration from a coarser and independent representation of the computational domain.

In terms of numerical simulations, we perform a grid refinement study, using also different Finite Element spaces among the different core models, to show the effectiveness of our numerical approach. Moreover, we simulate the response of our electromechanical model to some physiologically relevant scenarios, namely varying preload, afterload and contractility.

1.1 Paper outline

In Sec. 2 we briefly recall the mathematical model of cardiac electromechanics considered in this paper. In Sec. 3 we present the numerical discretization of the different core models and, more importantly, the strategy followed for their coupling. In Sec. 4 we illustrate our algorithm to recover the reference configuration from a stressed geometry. In Sec. 5 we present the numerical results: we perform a grid refinement study and we analyze different numerical tests by varying preload, afterload and contractility, thus showing the suitability of our model to investigate different physiologically relevant scenarios. Finally, in Sec. 6, we draw our conclusions and outlook on future developments.

2 Mathematical models

We recall the models proposed in the Part I of this paper [49]:

$$
(\mathscr{E}) \begin{cases} \chi_{\mathrm{m}} \left[C_{\mathrm{m}} \dfrac{\partial u}{\partial t} + \mathcal{I}_{\mathrm{ion}}(u, \boldsymbol{w}, \boldsymbol{z}) \right] - \nabla \cdot (J \mathbf{F}^{-1} \boldsymbol{D}_{\mathrm{M}} \mathbf{F}^{-T} \nabla u) \\ \hspace{7cm} = \mathcal{I}_{\mathrm{app}}(t), \\[2mm] \left(J \mathbf{F}^{-1} \boldsymbol{D}_{\mathrm{M}} \mathbf{F}^{-T} \nabla u \right) \cdot \mathbf{N} = 0, \end{cases} \tag{1a}
$$

in $\Omega_0 \times (0, T)$, with $u = u_0$ in Ω_0 at time $t = 0$.

$$
(\mathscr{I}) \begin{cases} \dfrac{\partial \boldsymbol{w}}{\partial t} - \boldsymbol{H}(u, \boldsymbol{w}) = \mathbf{0}, \\[2mm] \dfrac{\partial \boldsymbol{z}}{\partial t} - \boldsymbol{G}(u, \boldsymbol{w}, \boldsymbol{z}) = \mathbf{0}, \end{cases} \tag{1b}
$$

in $\Omega_0 \times (0, T)$, with $\boldsymbol{w} = \boldsymbol{w}_0$ and $\boldsymbol{z} = \boldsymbol{z}_0$ in Ω_0 at time $t = 0$.

$$
(\mathscr{A}) \quad \frac{\partial \mathbf{s}}{\partial t} = \boldsymbol{K}(\mathbf{s}, [\mathrm{Ca}^{2+}]_{\mathrm{i}}, SL), \tag{1c}
$$

in $\Omega_0 \times (0, T)$, with $\mathbf{s}(0) = \mathbf{s}_0$ in Ω_0 at time $t = 0$.

$$
(\mathscr{M}) \begin{cases} \rho_{\mathrm{s}} \dfrac{\partial^2 \mathbf{d}}{\partial t^2} - \nabla \cdot \mathbf{P}(\mathbf{d}, T_{\mathrm{a}}(\mathbf{s})) = \mathbf{0}, \\[2mm] \mathbf{P}(\mathbf{d}, T_{\mathrm{a}}(\mathbf{s})) \mathbf{N} + \mathbf{K}^{\mathrm{epi}} \mathbf{d} + \mathbf{C}^{\mathrm{epi}} \dfrac{\partial \mathbf{d}}{\partial t} = \mathbf{0} & \text{on } \Gamma_0^{\mathrm{epi}} \times (0, T), \\[2mm] \mathbf{P}(\mathbf{d}, T_{\mathrm{a}}(\mathbf{s})) \mathbf{N} = p_{\mathrm{LV}}(t) \, |J \mathbf{F}^{-T} \mathbf{N}| \, \mathbf{v}^{\mathrm{base}}(t) & \text{on } \Gamma_0^{\mathrm{base}} \times (0, T), \\[2mm] \mathbf{P}(\mathbf{d}, T_{\mathrm{a}}(\mathbf{s})) \mathbf{N} = -p_{\mathrm{LV}}(t) \, J \mathbf{F}^{-T} \mathbf{N} & \text{on } \Gamma_0^{\mathrm{endo}} \times (0, T), \end{cases} \tag{1d}
$$

in $\Omega_0 \times (0, T)$, with $\mathbf{d} = \mathbf{d}_0$ and $\dfrac{\partial \mathbf{d}}{\partial t} = \dot{\mathbf{d}}_0$ in Ω_0 at time $t = 0$.

$$
(\mathscr{C}) \quad \frac{d\boldsymbol{c}_1(t)}{dt} = \widetilde{\boldsymbol{D}}(t, \boldsymbol{c}_1(t), p_{\mathrm{LV}}(t)), \tag{1e}
$$

for $t \in (0, T)$, with $\boldsymbol{c}_1(0) = \boldsymbol{c}_{1,0}$.

$$
(\mathscr{V}) \quad V_{\mathrm{LV}}^{\mathrm{0D}}(\boldsymbol{c}_1(t)) = V_{\mathrm{LV}}^{\mathrm{3D}}(\mathbf{d}(t)), \tag{1f}
$$

for $t \in (0, T)$.

In these models, $u \colon \Omega_0 \times (0, T) \to \mathbb{R}$ denotes the transmembrane potential, $\boldsymbol{w} \colon \Omega_0 \times (0, T) \to \mathbb{R}^{n_{\boldsymbol{w}}}$ and $\boldsymbol{z} \colon \Omega_0 \times (0, T) \to \mathbb{R}^{n_{\boldsymbol{z}}}$ the ionic variables (respectively describing the opening of ionic channels and the ionic concentrations). The vector-valued variable $\mathbf{s} \colon \Omega_0 \times (0, T) \to \mathbb{R}^{n_{\mathbf{s}}}$ collects the states variables of the force generation model. The variable $\mathbf{d} \colon \Omega_0 \times (0, T) \to \mathbb{R}^3$ denotes the displacement of the tissue. Finally, $\boldsymbol{c}_1 \colon (0, T) \to \mathbb{R}^{n_{\mathrm{c}}}$ is the state vector of the circulation models (collecting the pressures, volumes and fluxes in the different compartments of the vascular network) and $p_{\mathrm{LV}} \colon (0, T) \to \mathbb{R}$ denotes the left ventricle pressure.

In Eq. (1), cardiac electrophysiology is described by the monodomain equation $(\mathscr{E})$ [14], coupled with the ten Tusscher-Panfilov ionic model $(\mathscr{I})$ (henceforth denoted by TTP06) [55]. The generation of active force $(\mathscr{A})$ is described by means of an Artificial Neural Network (ANN) based model that surrogates the high-fidelity model proposed in [46] (RDQ18 model). This reduced model,

built by exploiting the Machine Learning algorithm proposed in [47, 48], drastically reduces the computational cost associated with the resolution of the RDQ18 model, still reproducing its results with a very good accuracy [48]. In ($\mathscr{M}$), we model the passive elastic properties of the cardiac tissue through the Guccione constitutive model [26, 27]. Finally, blood circulation is described by means of the lumped parameters model introduced in Part I of this paper [49].

3 Numerical approximation

We sketch in Fig. 1 the numerical approximation of problem (1). We use a segregated numerical scheme that allows to separate and properly manage the space and time scales related to cardiac electromechanics, by using different mesh sizes and different time steps according to the characteristic scale of each core model. Our scheme is thus segregated (i.e. the different core models are solved sequentially) and staggered (i.e. possibly different time step sizes are used for the different core models) [21, 52].

We perform the numerical approximation of the core models using the Finite Element Method (FEM) in space and Finite Difference schemes in time [42]. We employ an explicit 4th order Runge-Kutta method for the set of ODEs ($\mathscr{C}$). This high-order time discretization is motivated by the stiffness of the circulation model [42].

We consider a fine representation of the computational domain for both ($\mathscr{I}$) and ($\mathscr{E}$) models, whereas a coarser one is employed for both ($\mathscr{A}$) and ($\mathscr{M}$). This is motivated by the requirement of a higher resolution for ($\mathscr{I}$) and ($\mathscr{E}$), due to the sharp wavefronts characterizing electrophysiological solutions, whereas both ($\mathscr{A}$) and ($\mathscr{M}$) feature larger spatial scales [13, 15, 52]. Moreover, the nonlinearities of ($\mathscr{M}$) warrants for the use of a coarser space discretization to make the numerical solution less computationally intensive.

For what concerns time discretization, we use a staggered strategy based on the Godunov splitting scheme [24]. This approach introduces a first order splitting error [17, 21], which is compatible with the first order error associated with the backward Euler method used for ($\mathscr{I}$), ($\mathscr{E}$), ($\mathscr{A}$) and ($\mathscr{M}$)–($\mathscr{V}$). Moreover, the Godunov splitting scheme does not undermine the stability of the numerical scheme [17]. In our pipeline, we first update the variables of ($\mathscr{I}$) and ($\mathscr{E}$), then the variables of ($\mathscr{A}$) and finally, after updating the unknowns of ($\mathscr{C}$), we update the ones of ($\mathscr{M}$)–($\mathscr{V}$).

For electrophysiology, we employ a semi-implicit scheme, that we denote by ($\mathscr{I}_{\text{SI}}$)–($\mathscr{E}_{\text{SI}}$), where SI stands for semi-implicit. Mechanical activation ($\mathscr{A}_{\text{E}}$), is solved with an explicit method in time. Mechanics ($\mathscr{M}_{\text{I}}$) is instead implicitly discretized in time, due to the fact that the highly nonlinear (exponential) terms of the strain energy function $\mathcal{W}$ would need a restriction on the time step in both the semi-implicit and explicit contexts. Finally, we employ an explicit 4th order Runge-Kutta scheme for the circulation model, indicated as ($\mathscr{C}_{\text{E}}$). We use two different time steps for ($\mathscr{I}_{\text{SI}}$)–($\mathscr{E}_{\text{SI}}$)–($\mathscr{A}_{\text{E}}$)–($\mathscr{C}_{\text{E}}$) and for ($\mathscr{M}_{\text{I}}$)–($\mathscr{V}_{\text{I}}$).

We propose a novel strategy to perform the 3D-0D coupling, given by ($\mathscr{V}$), between ($\mathscr{M}$) and ($\mathscr{C}$), at the numerical level, within a segregated setting. Specifically, we solve the 3D mechanical problem under a volumetric constraint coming from the 0D circulation model. The cavity pressure acts as Lagrange multiplier associated with this constraint. Thus, we obtain a saddle-point problem that we address, at the algebraic level, by means of a Schur complement reduction.

3.1 Space discretization and intergrid interpolation

We implemented an intergrid transfer operator between nested grids that can be generalized to the case of locally-refined non-conforming nested grids. More details about this strategy can be found
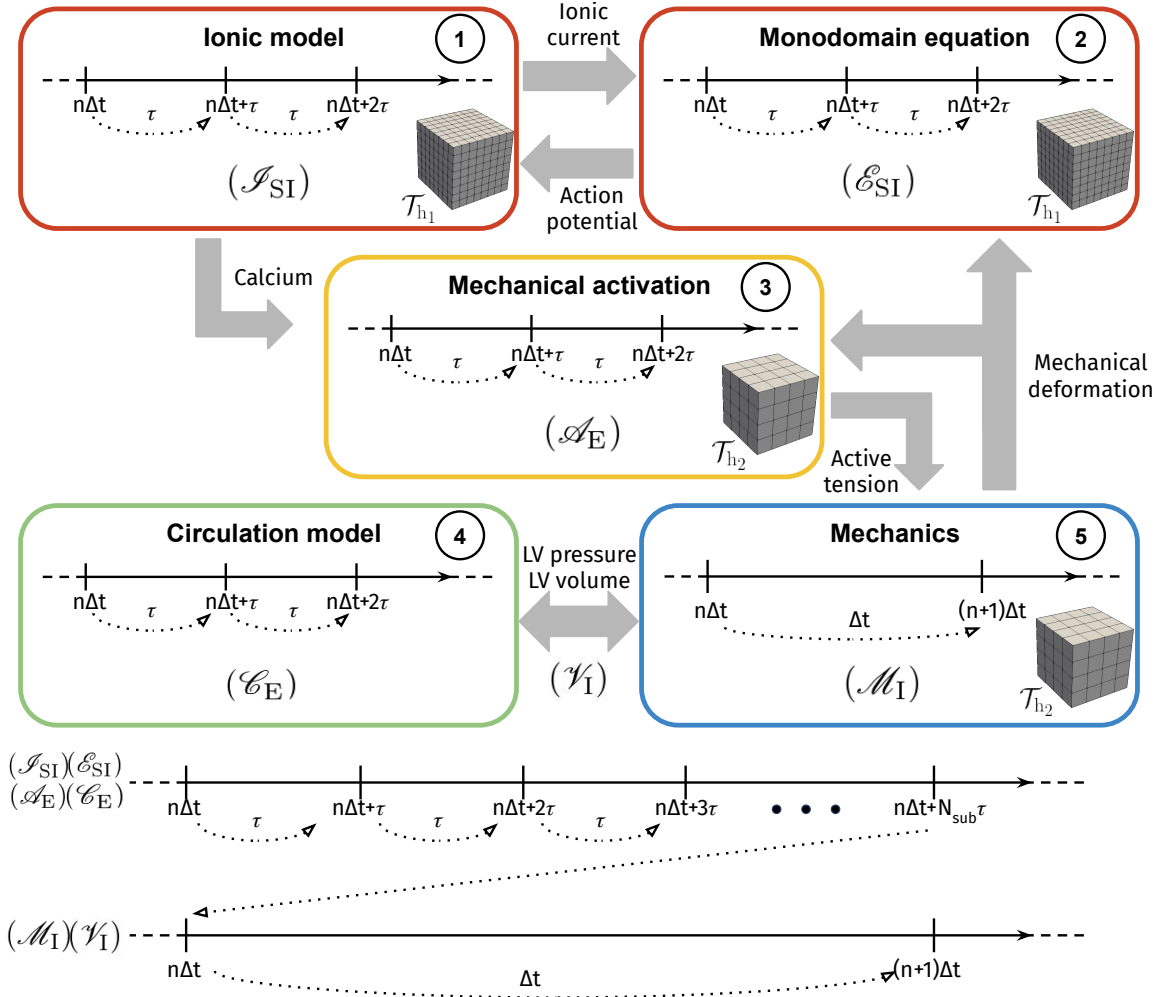
Figure 1: Sketch of the segregated-intergrid-staggered numerical scheme.

in [1]. We built an efficient and scalable interpolation data structure on top of the intergrid transfer operator that enables to evaluate the feedback coming from $(\mathscr{E})$ at the quadrature nodes defined on the mesh used for $(\mathscr{M})$ (and vice-versa), regardless of the polynomial degree used to discretize the two different models. This enables for a high level of numerical flexibility: different Finite Element degrees, as well as different levels of mesh refinement, can be effortlessly selected to tune the computational efficiency on the desired accuracy.

We consider two nested hexahedral meshes $\mathcal{T}_{h_1}$ and $\mathcal{T}_{h_2}$ of the computational domain Ω_0, where $\mathcal{T}_{h_1}$ has been generated by uniformly refining $\mathcal{T}_{h_2}$ according to an octree structure [1, 11], i.e. by recursively splitting each parent element of $\mathcal{T}_{h_2}$ into eight sub-elements for a prescribed number of times, i.e. until the desired geometrical detail is reached. Here h_1 and h_2 (with $h_1 < h_2$) represent the mesh sizes, which we compute as the mean of the maximum diameter of each element.

We denote by N_u, $N_{\boldsymbol{w,z}}$, N_s and $N_{\mathbf{d}}$ the number of degrees of freedom (DOFs, that is the number of variables) for the dimensionless transmembrane potential, gating and concentration variables, mechanical activation variables and displacement, respectively. We denote the set of tensor-products of polynomials with degree smaller than or equal to r over a mesh element K by $\mathbb{Q}_r(K)$, and we introduce the finite dimensional spaces $\mathcal{X}_{h_1}^r = \{v \in C^0(\bar{\Omega}_0) : v|_K \in \mathbb{Q}_r(K) \quad \forall K \in \mathcal{T}_{h_1}\}$ and $\mathcal{X}_{h_2}^s = \{v \in C^0(\bar{\Omega}_0) : v|_K \in \mathbb{Q}_s(K) \quad \forall K \in \mathcal{T}_{h_2}\}$, for $r, s \geq 1$.

We adopt the following notation. We denote by, e.g., $\mathbf{d}_{h_1}(t) \approx \mathbf{d}(t)$ the semi-discretized Finite Element approximation of the variable $\mathbf{d}(t)$, defined over the computational mesh $\mathcal{T}_{h_1}$. On the other hand, we denote by $\underline{\mathbf{d}}_{h_1}(t)$ the vector collecting the DOFs associated with $\mathbf{d}_{h_1}(t)$. Finally, we denote by $\underline{\mathbf{d}}_{h_1}^n \simeq \underline{\mathbf{d}}_{h_1}(t^n)$ the vector collecting the DOFs of the fully discretized Finite Element problem.

Monodomain equation. The set of basis functions for $\mathcal{X}_{h_1}^r$ with $N_u = \dim(\mathcal{X}_{h_1}^r)$ is given by $\{\phi_i\}_{i=1}^{N_u}$. The semi-discretized formulation of the monodomain equation reads: find $u_{h_1}(t) \in \mathcal{X}_{h_1}^r$ for all $t \in (0, T)$ such that

$$
\begin{aligned}
&\int_{\Omega_0} \dot{u}_{h_1}(t)\phi_i \, d\Omega_0 + \int_{\Omega_0} (J_{h_1} \mathbf{F}_{h_1}^{-1}(\mathbf{d}_{h_1}(t)) \widetilde{\boldsymbol{D}}_M \mathbf{F}_{h_1}^{-T}(\mathbf{d}_{h_1}(t)) \nabla u_{h_1}(t)) \cdot \nabla \phi_i \, d\Omega_0 \\
&+ \int_{\Omega_0} \widetilde{\mathcal{I}}_{\text{ion}}(u_{h_1}(t), \boldsymbol{w}_{h_1}(t), \boldsymbol{z}_{h_1}(t))\phi_i \, d\Omega_0 = \int_{\Omega_0} \widetilde{\mathcal{I}}_{\text{app}}(t)\phi_i \, d\Omega_0 \quad \forall i = 1, ..., N_u,
\end{aligned}
\tag{2}
$$

with $u_{h_1}(0) = \sum_{j=1}^{N_u} (u_0, \phi_j)_{L^2(\Omega_0)} \phi_j$. The functions $\boldsymbol{w}_{h_1}(t)$ and $\boldsymbol{z}_{h_1}(t)$ are the semi-discretized versions of the gating variables and of the concentration variables, whereas $u_{h_1}(t) = \sum_{j=1}^{N_u} u_{j,h_1}(t)\phi_j$ is the Finite Element solution that approximates $u = u(t)$. The tensor $\mathbf{F}_{h_1}$ is the interpolated deformation tensor, obtained through the following procedure:

- The approximate solution $\mathbf{d}_{h_2}$ to problem $(\mathscr{M})$ is interpolated from $\mathcal{T}_{h_2}$ to $\mathcal{T}_{h_1}$ to compute $\mathbf{d}_{h_1}$. The evaluation of $\mathbf{d}_{h_1}$ at the quadrature points used to approximate Eq. (2) is performed using our intergrid transfer operator between nested meshes and arbitrary Finite Element spaces.

- We build $\mathbf{F}_{h_1} = \mathbf{I} + \nabla \mathbf{d}_{h_1}$ directly on $\mathcal{T}_{h_1}$ after the interpolation procedure applied on the displacement field.

We rewrite Eq. (2) as a system of nonlinear ODEs by setting $\underline{\boldsymbol{u}}_{h_1}(t) = \{u_{j,h_1}(t)\}_{j=1}^{N_u}$:

$$
\begin{cases}
\mathcal{M}\dot{\underline{\boldsymbol{u}}}_{h_1}(t) + \mathcal{K}(\underline{\mathbf{d}}_{h_1}(t))\underline{\boldsymbol{u}}_{h_1}(t) + \boldsymbol{I}_{\text{ion}}(\underline{\boldsymbol{u}}_{h_1}(t), \underline{\boldsymbol{w}}_{h_1}(t), \underline{\boldsymbol{z}}_{h_1}(t)) = \boldsymbol{I}_{\text{app}}(t) \quad \forall t \in (0, T), \\
\underline{\boldsymbol{u}}_{h_1}(0) = \underline{\boldsymbol{u}}_{0,h_1},
\end{cases}
\tag{3}
$$

where we have defined the following matrices

$$\mathcal{M}_{ij} = \int_{\Omega_0} \phi_j \phi_i \, d\Omega_0, \quad \mathcal{K}_{ij}(\underline{\mathbf{d}}_{h_1}(t)) = \int_{\Omega_0} (J_{h_1} \mathbf{F}_{h_1}^{-1} \boldsymbol{D}_m \mathbf{F}_{h_1}^{-T} \nabla \phi_j) \cdot \nabla \phi_i \, d\Omega_0,$$

and the following vectors

$$\left(\boldsymbol{I}_{\text{ion}}(\underline{\boldsymbol{u}}_{h_1}(t), \underline{\boldsymbol{w}}_{h_1}(t), \underline{\boldsymbol{z}}_{h_1}(t)) \right)_i = \int_{\Omega_0} \widetilde{\mathcal{I}}_{\text{ion}}(u_{h_1}(t), \boldsymbol{w}_{h_1}(t), \boldsymbol{z}_{h_1}(t)) \phi_i \, d\Omega_0,$$

$$(\boldsymbol{I}_{\text{app}}(t))_i = \int_{\Omega_0} \widetilde{\mathcal{I}}_{\text{app}}(t) \phi_i \, d\Omega_0.$$

For the evaluation of the nonlinear term $\boldsymbol{I}_{\text{ion}}(\underline{\boldsymbol{u}}_{h_1}(t), \underline{\boldsymbol{w}}_{h_1}(t), \underline{\boldsymbol{z}}_{h_1}(t))$, three strategies are available [22, 33, 36–38]. In this work, we use the so-called ionic current interpolation (ICI) approach, which yields a faster assembly of the ionic term [22, 33]. By denoting by $\{\boldsymbol{x}_q^K\}_{q=1}^{N_q}$ and $\{\omega_q^K\}_{q=1}^{N_q}$ the quadrature nodes and weights of a generic mesh element of $K \in \mathcal{T}_{h_1}$, the term $\boldsymbol{I}_{\text{ion}}(\underline{\boldsymbol{u}}_{h_1}(t), \underline{\boldsymbol{w}}_{h_1}(t), \underline{\boldsymbol{z}}_{h_1}(t))$ is firstly evaluated at the DOFs and then interpolated at the quadrature nodes, i.e.:

$$\int_{\Omega_0} \widetilde{\mathcal{I}}_{\text{ion}}(u_{h_1}(t), \boldsymbol{w}_{h_1}(t), \boldsymbol{z}_{h_1}(t)) \phi_i \, d\Omega_0$$

$$\approx \sum_{K \in \mathcal{T}_{h_1}} \left(\sum_{q=1}^{N_q} \sum_{j=1}^{N_u} \widetilde{\mathcal{I}}_{\text{ion}} \left(u_{j,h_1}(t), \boldsymbol{w}_{j,h_1}(t) \boldsymbol{z}_{j,h_1}(t) \right) \phi_j(\boldsymbol{x}_q^K) \phi_i(\boldsymbol{x}_q^K) \omega_q^K \right). \tag{4}$$

Ionic model. The ionic model under consideration is a system of 18 ODEs (12 for the gating variables, 6 for the concentration variables), which indirectly depends on the space variable over the mesh $\mathcal{T}_{h_1}$ through the transmembrane potential u. The semi-discrete formulation can be written as follows:

$$\begin{cases} \underline{\dot{\boldsymbol{w}}}_{h_1}(t) = \overline{\boldsymbol{H}}(\underline{\boldsymbol{u}}_{h_1}(t), \underline{\boldsymbol{w}}_{h_1}(t)) & \forall t \in (0, T), \\ \underline{\dot{\boldsymbol{z}}}_{h_1}(t) = \overline{\boldsymbol{G}}(\underline{\boldsymbol{u}}_{h_1}(t), \underline{\boldsymbol{w}}_{h_1}(t), \underline{\boldsymbol{z}}_{h_1}(t)) & \forall t \in (0, T), \\ \underline{\boldsymbol{w}}_{h_1}(0) = \underline{\boldsymbol{w}}_{0,h_1}, \\ \underline{\boldsymbol{z}}_{h_1}(0) = \underline{\boldsymbol{z}}_{0,h_1}. \end{cases} \tag{5}$$

Mechanical activation model. The semi-discrete formulation, which is written on $\mathcal{T}_{h_2}$, reads:

$$\begin{cases} \underline{\dot{\mathbf{s}}}_{h_2}(t) = \overline{\boldsymbol{K}}(\underline{\mathbf{s}}_{h_2}(t), ([\text{Ca}^{2+}]_{i,h_2}(t), SL_{h_2}(t))^T) & \forall t \in (0, T), \\ \underline{\mathbf{s}}_{h_2}(0) = \underline{\mathbf{s}}_{0,h_2}. \end{cases} \tag{6}$$

where $[\text{Ca}^{2+}]_{i,h_2}(t)$ is obtained by interpolating the intracellular calcium concentration of TTP06 model from $\mathcal{T}_{h_1}$ to $\mathcal{T}_{h_2}$ as explained in the introduction to this section. The operator $\overline{\boldsymbol{K}}$ represents the element-wise application of the ANN associated with the activation model, previously trained from a collection of simulations obtained with the high-fidelity RDQ18 model [46, 48]. On the other hand, $SL_{h_2}(t)$ is obtained by solving, for any $t \in (0, T)$:

$$\begin{cases} \left(SL_{h_2}(t) - SL_0 \sqrt{\mathcal{I}_{4f,h_2}(t)} \right) - \delta_{SL}^2 \Delta SL_{h_2}(t) = 0 & \text{in } \Omega_0 \times (0, T) \\ \delta_{SL}^2 \nabla SL_{h_2}(t) \cdot \mathbf{N}_{h_2} = 0 & \text{on } \partial\Omega_0 \times (0, T) \end{cases} \tag{7}$$

where $\mathcal{I}_{4f,h_2}(t) = \mathbf{F}_{h_2}(t)\mathbf{f}_0 \cdot \mathbf{F}_{h_2}(t)\mathbf{f}_0$. Finally, $T_{a,h_2}(t)$ denotes the semi-discretized active tension, obtained by evaluating the function $T_a = T_a^{\max} G(\mathbf{s})$ at the nodes.

Mechanical model. We denote by $[\mathcal{X}_{h_2}^s]^3$ the finite dimensional subspace of vector valued functions and by $\{\boldsymbol{\phi}_i\}_{i=1}^{N_d}$ its basis. The semi-discretized version of $(\mathcal{M})$ reads: given $T_{a,h_2}(t)$, find $\mathbf{d}_{h_2} = \mathbf{d}_{h_2}(t) \in [\mathcal{X}_{h_2}^s]^3$ for all $t \in (0,T)$ such that

$$
\int_{\Omega_0} \rho_s \ddot{\mathbf{d}}_{h_2}(t) \cdot \boldsymbol{\phi}_i \, d\Omega_0 + \int_{\Omega_0} \mathbf{P}(\mathbf{d}_{h_2}(t), T_{a,h_2}(t)) : \nabla\boldsymbol{\phi}_i d\Omega_0
$$
$$
+ \int_{\Gamma_0^{\text{epi}}} \left[(\mathbf{N}_{h_2} \otimes \mathbf{N}_{h_2}) \left(K_\perp^{\text{epi}} \mathbf{d}_{h_2}(t) + C_\perp^{\text{epi}} \dot{\mathbf{d}}_{h_2}(t) \right) \right.
$$
$$
\left. + (\mathbf{I} - \mathbf{N}_{h_2} \otimes \mathbf{N}_{h_2}) \left(K_\parallel^{\text{epi}} \mathbf{d}_{h_2}(t) + C_\parallel^{\text{epi}} \dot{\mathbf{d}}_{h_2}(t) \right) \right] \cdot \boldsymbol{\phi}_i \, d\Gamma_0 \qquad (8)
$$
$$
= -p_{\text{LV}}(t) \int_{\Gamma_0^{\text{endo}}} J_{h_2} \mathbf{F}_{h_2}^{-T} \mathbf{N} \cdot \boldsymbol{\phi}_i \, d\Gamma_0 + p_{\text{LV}}(t) \int_{\Gamma_0^{\text{base}}} |J_{h_2} \mathbf{F}_{h_2}^{-T} \mathbf{N}| \mathbf{v}_{h_2}^{\text{base}} \cdot \boldsymbol{\phi}_i \, d\Gamma_0
$$
$$
\forall i = 1, ..., N_{\mathbf{d}},
$$

with $\mathbf{d}_{h_2}(0) = \sum_{j=1}^{N_{\mathbf{d}}} (\mathbf{d}_0, \boldsymbol{\phi}_j)_{[L^2(\Omega_0)]^3} \boldsymbol{\phi}_j$, $\dot{\mathbf{d}}_{h_2}(0) = \sum_{j=1}^{N_{\mathbf{d}}} (\dot{\mathbf{d}}_0, \boldsymbol{\phi}_j)_{[L^2(\Omega_0)]^3} \boldsymbol{\phi}_j$ and where

$$
\mathbf{v}_{h_2}^{\text{base}} = \frac{\int_{\Gamma_0^{\text{endo}}} J_{h_2} \mathbf{F}_{h_2}^{-T} \mathbf{N} \, d\Gamma_0}{\int_{\Gamma_0^{\text{base}}} |J_{h_2} \mathbf{F}_{h_2}^{-T} \mathbf{N}| \, d\Gamma_0}.
$$

The corresponding algebraic formulation reads:

$$
\begin{cases}
\rho_s \mathcal{M} \ddot{\underline{\mathbf{d}}}_{h_2}(t) + \mathcal{F} \dot{\underline{\mathbf{d}}}_{h_2}(t) + \mathcal{G} \underline{\mathbf{d}}_{h_2}(t) + \boldsymbol{S}(\underline{\mathbf{d}}_{h_2}(t), T_{a,h_2}(t)) = p_{\text{LV}}(t) \, \boldsymbol{p}(\underline{\mathbf{d}}_{h_2}(t)) \\
\\
\underline{\mathbf{d}}_{h_2}(0) = \underline{\mathbf{d}}_{0,h_2}, \quad \dot{\underline{\mathbf{d}}}_{h_2}(0) = \dot{\underline{\mathbf{d}}}_{0,h_2},
\end{cases} \qquad \forall t \in (0,T), \quad (9)
$$

with:

$$
\boldsymbol{S}_i(\underline{\mathbf{d}}_{h_2}(t), T_{a,h_2}(t)) = \int_{\Omega_0} \mathbf{P}(\mathbf{d}_{h_2}(t), T_{a,h_2}(t)) : \nabla\boldsymbol{\phi}_i \, d\Omega_0,
$$

$$
\mathcal{F}_{i,j} = \int_{\Gamma_0^{\text{epi}}} \left[(\mathbf{N}_{h_2} \otimes \mathbf{N}_{h_2}) C_\perp^{\text{epi}} + (\mathbf{I} - \mathbf{N}_{h_2} \otimes \mathbf{N}_{h_2}) C_\parallel^{\text{epi}} \right] \boldsymbol{\phi}_j \cdot \boldsymbol{\phi}_i \, d\Gamma_0,
$$

$$
\mathcal{G}_{i,j} = \int_{\Gamma_0^{\text{epi}}} \left[(\mathbf{N}_{h_2} \otimes \mathbf{N}_{h_2}) K_\perp^{\text{epi}} + (\mathbf{I} - \mathbf{N}_{h_2} \otimes \mathbf{N}_{h_2}) K_\parallel^{\text{epi}} \right] \boldsymbol{\phi}_j \cdot \boldsymbol{\phi}_i \, d\Gamma_0,
$$

$$
\boldsymbol{p}_i(\underline{\mathbf{d}}_{h_2}(t)) = \int_{\Gamma_0^{\text{base}}} |J_{h_2} \mathbf{F}_{h_2}^{-T} \mathbf{N}| \mathbf{v}_{h_2}^{\text{base}} \cdot \boldsymbol{\phi}_i \, d\Gamma_0 - \int_{\Gamma_0^{\text{endo}}} J_{h_2} \mathbf{F}_{h_2}^{-T} \mathbf{N} \cdot \boldsymbol{\phi}_i \, d\Gamma_0,
$$

where $\underline{\mathbf{d}}_{h_2}(t) = \{\mathbf{d}_{j,h_2}(t)\}_{j=1}^{N_{\mathbf{d}}}$. Equations (2), (5), (6) and (8) provide a splitted semi-discretization of the entire electromechanical model.

3.2 Time discretization

We define Δt as the time step for $(\mathcal{M}_\text{I})$–$(\mathcal{V}_\text{I})$ and $\tau = \Delta t / N_{\text{sub}}$ as the time step for $(\mathcal{I}_{\text{SI}})$, $(\mathcal{E}_{\text{SI}})$, $(\mathcal{A}_\text{E})$ and $(\mathcal{C}_\text{E})$, being $N_{\text{sub}} \in \mathbb{N}$ the number of intermediate substeps that need to be solved by $(\mathcal{I}_{\text{SI}})$–$(\mathcal{E}_{\text{SI}})$–$(\mathcal{A}_\text{E})$–$(\mathcal{C}_\text{E})$ before a time step Δt of $(\mathcal{M}_\text{I})$–$(\mathcal{V}_\text{I})$ is performed.

Problem $(\mathcal{I}_{\text{SI}})$–$(\mathcal{E}_{\text{SI}})$–$(\mathcal{A}_\text{E})$–$(\mathcal{C}_\text{E})$ from t^n to t^{n+1}, once we set $t^{n+\frac{m}{N_{\text{sub}}}} = t^n + m\tau$, for $m = 1, ..., N_{\text{sub}}$, reads as follows:

8

- We find $\underline{\boldsymbol{w}}_{h_1}^{n+\frac{m}{N_{\text{sub}}}}$ and $\underline{\boldsymbol{z}}_{h_1}^{n+\frac{m}{N_{\text{sub}}}}$ defined on $\mathcal{T}_{h_1}$ by solving:

$$
\begin{cases}
\dfrac{1}{\tau}\underline{\boldsymbol{w}}_{h_1}^{n+\frac{m}{N_{\text{sub}}}} = \dfrac{1}{\tau}\underline{\boldsymbol{w}}_{h_1}^{n} + \overline{\boldsymbol{H}}(\underline{\boldsymbol{u}}_{h_1}^{n},\underline{\boldsymbol{w}}_{h_1}^{n+\frac{m}{N_{\text{sub}}}}), \\[2mm]
\dfrac{1}{\tau}\underline{\boldsymbol{z}}_{h_1}^{n+\frac{m}{N_{\text{sub}}}} = \dfrac{1}{\tau}\underline{\boldsymbol{z}}_{h_1}^{n} + \overline{\boldsymbol{G}}(\underline{\boldsymbol{u}}_{h_1}^{n},\underline{\boldsymbol{w}}_{h_1}^{n},\underline{\boldsymbol{z}}_{h_1}^{n}).
\end{cases}
\tag{10}
$$

  We adopt here the first order implicit-explicit (IMEX) scheme proposed in [45]. Specifically, we employ an explicit treatment of the ionic concentrations to avoid the solution of a nonlinear system (such choice does not compromise the stability of the scheme, thanks to the non-stiff dynamics of concentrations), and an implicit treatment of the gating variables, because of the severe CFL condition on the time step induced by an explicit scheme. We notice that, thanks to the linear dynamics of the gating variables, such implicit handling does not require the solution of a system of linear or nonlinear equations.

- We interpolate $\underline{\mathbf{d}}_{h_2}^{n}$ on the fine mesh $\mathcal{T}_{h_1}$ once per time step, at $t = t^n$. We use $\underline{\boldsymbol{z}}_{h_1}^{n+\frac{m}{N_{\text{sub}}}}$ from (10) and $\underline{\mathbf{d}}_{h_1}^{n}$ to find $\underline{\boldsymbol{u}}_{h_1}^{n+\frac{m}{N_{\text{sub}}}}$ over $\mathcal{T}_{h_1}$ by solving:

$$
\begin{aligned}
&\left(\frac{1}{\tau}\mathcal{M} + \mathcal{K}(\underline{\mathbf{d}}_{h_1}^{n}) + \boldsymbol{I}_{\mathrm{u}}^{\mathrm{ion}}\left(\underline{\boldsymbol{u}}_{h_1}^{n},\underline{\boldsymbol{z}}_{h_1}^{n+\frac{m}{N_{\text{sub}}}}\right)\right)\underline{\boldsymbol{u}}_{h_1}^{n+\frac{m}{N_{\text{sub}}}} = \\
&\quad \frac{1}{\tau}\mathcal{M}\underline{\boldsymbol{u}}_{h_1}^{n} - \tilde{\boldsymbol{I}}^{\mathrm{ion}}\left(\underline{\boldsymbol{u}}_{h_1}^{n},\underline{\boldsymbol{z}}_{h_1}^{n+\frac{m}{N_{\text{sub}}}}\right) + \boldsymbol{I}_{\mathrm{app}}\left(t^{n+\frac{m}{N_{\text{sub}}}}\right).
\end{aligned}
\tag{11}
$$

  $\boldsymbol{I}_{\mathrm{u}}^{\mathrm{ion}}$ is the derivative of the terms of $\boldsymbol{I}_{\mathrm{ion}}$ that linearly depends on $\underline{\boldsymbol{u}}_{h_1}$, while $\tilde{\boldsymbol{I}}^{\mathrm{ion}}$ collects all the other terms.

- We interpolate $[\mathrm{Ca}^{2+}]_{\mathrm{i},h_2}^{n+\frac{m}{N_{\text{sub}}}}$ from (10) on the coarse mesh $\mathcal{T}_{h_2}$, and we find $\underline{\boldsymbol{s}}_{h_2}^{n+\frac{m}{N_{\text{sub}}}}$ by solving:

$$
\underline{\boldsymbol{s}}_{h_2}^{n+\frac{m}{N_{\text{sub}}}} = \underline{\boldsymbol{s}}_{h_2}^{n} + \tau\overline{\boldsymbol{K}}(\underline{\boldsymbol{s}}_{h_2}^{n},([\mathrm{Ca}^{2+}]_{\mathrm{i},h_2}^{n+\frac{m}{N_{\text{sub}}}},SL_{h_2}^{n})^T).
\tag{12}
$$

  where $SL_{h_2}^{n}$ is obtained by solving problem (7).

- We find $\boldsymbol{c}_1^{n+\frac{m}{N_{\text{sub}}}}$, with the 4th order Runge-Kutta method:

$$
\boldsymbol{c}_1^{n+\frac{m}{N_{\text{sub}}}} = \boldsymbol{c}_1^{n} + \frac{1}{6}\left(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4\right),
\tag{13}
$$

  with $\mathbf{k}_1$, $\mathbf{k}_2$, $\mathbf{k}_3$ and $\mathbf{k}_4$ computed as follows:

$$
\begin{aligned}
\mathbf{k}_1 &= \tau\widetilde{\boldsymbol{D}}\left(t^n,\boldsymbol{c}_1^{n},p_{\mathrm{LV}}^{n}\right), & \mathbf{k}_2 &= \tau\widetilde{\boldsymbol{D}}\left(t^{n+\frac{1}{2}\frac{m}{N_{\text{sub}}}},\boldsymbol{c}_1^{n}+\frac{\mathbf{k}_1}{2},p_{\mathrm{LV}}^{n}\right), \\[2mm]
\mathbf{k}_3 &= \tau\widetilde{\boldsymbol{D}}\left(t^{n+\frac{1}{2}\frac{m}{N_{\text{sub}}}},\boldsymbol{c}_1^{n}+\frac{\mathbf{k}_2}{2},p_{\mathrm{LV}}^{n}\right), & \mathbf{k}_4 &= \tau\widetilde{\boldsymbol{D}}\left(t^{n+\frac{m}{N_{\text{sub}}}},\boldsymbol{c}_1^{n}+\mathbf{k}_3,p_{\mathrm{LV}}^{n}\right).
\end{aligned}
$$

After having solved (10), (11), (12) and (13) for N_{sub} steps, we treat $(\mathcal{M}_I)$–$(\mathcal{V}_I)$ at t^{n+1} by updating $\underline{\mathbf{d}}_{h_2}^{n+1}$ and p_{LV}^{n+1} with the following system:

$$
\begin{cases}
\left(\rho_{\mathrm{s}}\dfrac{1}{\Delta t^2}\mathcal{M} + \dfrac{1}{\Delta t}\mathcal{F} + \mathcal{G}\right)\underline{\mathbf{d}}_{h_2}^{n+1} + \boldsymbol{S}(\underline{\mathbf{d}}_{h_2}^{n+1},\mathbf{T}_{\mathrm{a},h_2}^{n+1}) \\[2mm]
\qquad = \rho_{\mathrm{s}}\dfrac{2}{\Delta t^2}\mathcal{M}\underline{\mathbf{d}}_{h_2}^{n} - \rho_{\mathrm{s}}\dfrac{1}{\Delta t^2}\mathcal{M}\underline{\mathbf{d}}_{h_2}^{n-1} + \dfrac{1}{\Delta t}\mathcal{F}\underline{\mathbf{d}}_{h_2}^{n} + p_{\mathrm{LV}}^{n+1}\boldsymbol{p}(\underline{\mathbf{d}}_{h_2}^{n},\underline{\mathbf{d}}_{h_2}^{n+1}), \\[2mm]
V_{\mathrm{LV}}^{\mathrm{3D}}(\underline{\mathbf{d}}_{h_2}^{n+1}) = V_{\mathrm{LV}}^{\mathrm{0D}}(\boldsymbol{c}_1^{n+1}).
\end{cases}
\tag{14}
$$

9

Equation (14) is a nonlinear saddle-point problem. In the next section (Sec. 3.3) we provide details about its numerical approximation at the algebraic level. An alternative to our approach for numerical discretization would be employing implicit monolithic schemes, which are known to be stable and accurate, but at the same time they present constraints in the choice of the time steps too and they are characterized by high computational costs [22, 52]. Indeed, we can only use one time step and one mesh with a monolithic approach. Therefore, we are forced to choose a small time step and a fine representation of the computational domain due to the requirements of cardiac electrophysiology. Our approach is instead accurate and computationally efficient, thanks to the flexibility in the choice of both space and time resolution among the different core models.

3.3 Algorithm for the resolution of Eq. (14)

We approximate the solution of Eq. (14) by means of a quasi-Newton strategy [42], as we proposed in [48]. Specifically, in the computation of the Jacobian matrix, we neglect the derivative of the nonlocal term $\mathbf{v}_{h_2}^{base}$ in the pressure variable, and we update the Jacobian at each time step, but not through the iterations of Newton's loop. By moving all the terms in Eq. (14) to the left hand side and by rewriting its first and second line as $\mathbf{r}_{\mathbf{d}}^{n+1}(\underline{\mathbf{d}}_{h_2}^{n+1}, p_{LV}^{n+1}) = \mathbf{0}$ and $r_{p}^{n+1}(\underline{\mathbf{d}}_{h_2}^{n+1}) = 0$, respectively, the quasi-Newton algorithm reads as follows:

- We set $\underline{\mathbf{d}}_{h_2}^{n+1,0} = \underline{\mathbf{d}}_{h_2}^{n}$ and $p_{LV}^{n+1,0} = p_{LV}^{n}$.

- For $j = 0, 1, \ldots$, until a convergence criterion is not fulfilled, we solve the following linear system:
$$\begin{pmatrix} J_{\mathbf{d},\mathbf{d}}^{n+1} & J_{\mathbf{d},p}^{n+1} \\ J_{p,\mathbf{d}}^{n+1} & 0 \end{pmatrix} \begin{pmatrix} \Delta \underline{\mathbf{d}}_{h_2}^{n+1,j} \\ \Delta p_{LV}^{n+1,j} \end{pmatrix} = - \begin{pmatrix} \mathbf{r}_{\mathbf{d}}^{n+1,j} \\ r_{p}^{n+1,j} \end{pmatrix}, \tag{15}$$
where $J_{\mathbf{d},\mathbf{d}}^{n+1} \simeq \frac{\partial}{\partial \underline{\mathbf{d}}} \mathbf{r}_{\mathbf{d}}^{n+1}(\underline{\mathbf{d}}_{h_2}^{n}, p_{LV}^{n})$ (wherein we neglected the derivative with respect to the nonlocal term $\mathbf{v}_{h_2}^{base}$), $J_{\mathbf{d},p}^{n+1} = \frac{\partial}{\partial p} \mathbf{r}_{\mathbf{d}}^{n+1}(\underline{\mathbf{d}}_{h_2}^{n}, p_{LV}^{n})$, $J_{p,\mathbf{d}}^{n+1} = \frac{\partial}{\partial \underline{\mathbf{d}}} r_{p}^{n+1}(\underline{\mathbf{d}}_{h_2}^{n})$, $\mathbf{r}_{\mathbf{d}}^{n+1,j} = \mathbf{r}_{\mathbf{d}}^{n+1}(\underline{\mathbf{d}}_{h_2}^{n+1,j}, p_{LV}^{n+1,j})$ and $r_{p}^{n+1,j} = r_{p}^{n+1}(\underline{\mathbf{d}}_{h_2}^{n+1,j})$.

- We update $\underline{\mathbf{d}}_{h_2}^{n+1,j+1} = \underline{\mathbf{d}}_{h_2}^{n+1,j} + \Delta \underline{\mathbf{d}}_{h_2}^{n+1,j}$ and $p_{LV}^{n+1,j+1} = p_{LV}^{n+1,j} + \Delta p_{LV}^{n+1,j}$.

- When the convergence criterion is satisfied, we set $\underline{\mathbf{d}}_{h_2}^{n+1} = \underline{\mathbf{d}}_{h_2}^{n+1,j}$ and $p_{LV}^{n+1} = p_{LV}^{n+1,j}$.

From the algebraic viewpoint, we solve the saddle-point problem (15) via Schur complement reduction [7]. Specifically, we solve the two linear systems
$$J_{\mathbf{d},\mathbf{d}}^{n+1} \mathbf{v}^{n+1,j} = \mathbf{r}_{\mathbf{d}}^{n+1,j}, \qquad J_{\mathbf{d},\mathbf{d}}^{n+1} \mathbf{w}^{n+1,j} = J_{\mathbf{d},p}^{n+1}$$

and we set:
$$\Delta p_{LV}^{n+1,j} = \frac{r_{p}^{n+1,j} - J_{p,\mathbf{d}}^{n+1} \mathbf{v}^{n+1,j}}{J_{p,\mathbf{d}}^{n+1} \mathbf{w}^{n+1,j}}, \quad \Delta \underline{\mathbf{d}}_{h_2}^{n+1,j} = - \left(\mathbf{v}^{n+1,j} + \mathbf{w}^{n+1,j} \Delta p_{LV}^{n+1,j} \right). \tag{16}$$

We remark that, thanks to the reuse of the Jacobian matrix throughout the Newton loop, $\mathbf{w}^{n+1,j}$ becomes independent of j and thus it does not need to be recomputed at each iteration. Our scheme only involves the following operations: for each time step, we assemble the matrix $J_{\mathbf{d},\mathbf{d}}^{n+1}$ and the vectors $J_{p,\mathbf{d}}^{n+1}$ and $J_{\mathbf{d},p}^{n+1}$ and we solve the linear system $J_{\mathbf{d},\mathbf{d}}^{n+1} \mathbf{w}^{n+1,0} = J_{\mathbf{d},p}^{n+1}$; at each Newton iteration, we only need to solve the linear system $J_{\mathbf{d},\mathbf{d}}^{n+1} \mathbf{v}^{n+1,j} = \mathbf{r}_{\mathbf{d}}^{n+1,j}$ and perform a couple of

matrix-vector multiplications and a vector-vector sum of Eq. (16). We later show, through several numerical simulations, that this approach is numerically stable. Even more importantly, our scheme is appropriate for the whole heartbeat, as it does not require adaptations according to the specific cardiac phase.

Moreover, our approach allows for a segregated solution of the 3D-0D coupled model, which is instead typically solved through a monolithic strategy [28]. As a matter of fact, segregated schemes available in literature that couple a mechanical problem with a model describing the dynamics of a fluid (even when described through a 0D circulation model) generally fail when the fluid domain is fully enclosed by the solid structure, because the incompressibility constraint of the fluid is no longer satisfied after the structure update [28]. This issue, known as *balloon dilemma*, affects also cardiac chambers, either when coupled with a 3D circulation model – within a fluid-structure interaction framework [5, 25] – or with a 0D one. To overcome this issue, the 3D-0D cardiac circulation models available in literature rely either on a monolithic solution of the two models, where the 0D and 3D models are simultaneously discretized as a unique system and then typically solved by a Newton method [28] or iterative methods that progressively update the cavity pressures and the solid displacement, until convergence is reached. For instance, in [32], the authors proposed a method where the cavity pressure is initially estimated by extrapolating from previous time steps. Then, the cavity compliance (i.e. $\partial V / \partial p$, where p and V are the cavity pressure and volume, respectively) is estimated by finite differences and is used to update the pressure until the blood flux of the 3D model matches that of the 0D model within a prescribed tolerance. Similarly, in [17, 21], during the isovolumic phases, the cavity pressure is iteratively updated by a fixed point scheme. However, convergence of this scheme depends upon a relaxation parameter, whose optimal value needs to be manually assessed from case by case [21].

With our approach, instead, the mechanical 3D model ($\mathscr{M}$) and the circulation 0D model ($\mathscr{C}$) are solved in both a segregated and staggered manner. Indeed, we do not solve the ($\mathscr{M}$) model simultaneously to the ($\mathscr{C}$) model, but coupled to the volume-consistency condition ($\mathscr{V}$) instead. In this way we end up with the saddle-point problem (14). Thanks to the above algorithm, finally, at each Newton iteration, we only need to solve a linear system involving the Jacobian matrix of the ($\mathscr{M}$) problem, and perform a couple of matrix-vector operations.

3.4 Stresses computation

Starting from [21], we define the following indicator for evaluating the components of the mechanical stress:

$$S_{\mathrm{ab}} = (\mathbf{Pa}_0) \cdot \frac{\mathbf{Fb}_0}{|\mathbf{Fb}_0|}, \qquad (17)$$

where $a, b \in \{f, s, n\}$ indicate the fibers (f), the sheets (s) and the crossfibers (n) direction respectively. The metric S_{ab} measures the stress component in the $\mathbf{b}$ direction (where $\mathbf{b} = \frac{\mathbf{Fb}_0}{|\mathbf{Fb}_0|}$ denotes the direction $\mathbf{b}_0$ in the current configuration) across a surface normal to the direction $\mathbf{a}_0$. Hence, we refer to axial stresses when $a = b$, whereas we get shear stresses when $a \neq b$. We remark that in the active stress framework, $\mathbf{P}$ incorporates an additive decomposition between the passive and the active terms, the latter coming from the active tension T_{a}.

Even though a possible strategy to handle the calculation of the stress tensor would be to solve a L^2-projection problem [21], we represent each component as a piece-wise constant ($\mathbb{Q}_0$) Finite Element vector, where the average of the values over the quadrature points of each cell is associated with the only local degree of freedom corresponding to the cell centroid. Since the stresses are only processed for visualization purposes, our strategies turns out to be very efficient while not hampering the accuracy of the computation.

11

4 Recovering the reference stress-free configuration

Here we present the algorithm for recovering the reference configuration Ω_0 from the deformed configuration $\widetilde{\Omega}$, knowing that the latter is obtained from the former by applying a pressure $\widetilde{p}$ and an active tension $\widetilde{T}_a$. The steady state version of the PDE for cardiac mechanics reads:

$$\begin{cases} \nabla \cdot \mathbf{P}(\mathbf{d}, T_a) = \mathbf{0} & \text{in } \Omega_0, \\ \mathbf{P}(\mathbf{d}, T_a)\mathbf{N} + \mathbf{K}^{\text{epi}}\mathbf{d} = \mathbf{0} & \text{on } \Gamma_0^{\text{epi}}, \\ \mathbf{P}(\mathbf{d}, T_a)\mathbf{N} = p_{\text{LV}}\,|J\mathbf{F}^{-T}\mathbf{N}|\,\mathbf{v}^{\text{base}}(t) & \text{on } \Gamma_0^{\text{base}}, \\ \mathbf{P}(\mathbf{d}, T_a)\mathbf{N} = -p_{\text{LV}}\,J\mathbf{F}^{-T}\mathbf{N} & \text{on } \Gamma_0^{\text{endo}}. \end{cases} \tag{18}$$

In what follows, we denote by $\mathbf{d} = \mathbf{d}_{\text{eq}}(\mathbf{x}_0, p_{\text{LV}}, T_a)$ the equilibrium solution of Eq. (18) obtained on the computational domain of coordinate $\mathbf{x}_0$. Hence, our aim is finding a coordinate $\mathbf{x}_0$ such that $\mathbf{x}_0 + \mathbf{d}_{\text{eq}}(\mathbf{x}_0, p_{\text{LV}}, T_a) = \widetilde{\mathbf{x}}$.

4.1 Algorithm for the recovery of the reference configuration

A representation of this algorithm is shown in Fig. 2. We start by setting the coordinate of the reference configuration $\mathbf{x}_0$ equal to the coordinate of the deformed one (i.e. $\mathbf{x}_0^{(0)} = \widetilde{\mathbf{x}}$). Then, we solve the elastostatic problem of Eq. (18), and we get the displacement $\mathbf{d}^{(0)} = \mathbf{d}_{\text{eq}}(\mathbf{x}_0^{(0)}, p_{\text{LV}}, T_a)$ (Fig. 2, top-left). Since, when the configuration $\widetilde{\Omega}$ is recorded the active tension $\widetilde{T}_a$ is almost zero, in the deformed configuration $\mathbf{x}^{(0)} = \mathbf{x}_0^{(0)} + \mathbf{d}^{(0)}$ the ventricle is inflated compared to the configuration $\widetilde{\mathbf{x}}$. Thus, with the aim of correcting the mismatch between $\mathbf{x}^{(0)}$ and $\widetilde{\mathbf{x}}$, we deflate the ventricle by setting $\mathbf{x}_0^{(1)} = \mathbf{x}_0^{(0)} + (\widetilde{\mathbf{x}} - \mathbf{x}^{(0)}) = \widetilde{\mathbf{x}} - \mathbf{d}^{(0)}$ (Fig. 2, top-right). Then, we proceed by iterating the above steps. More precisely, for $k \geq 1$, we compute $\mathbf{d}^{(k)} = \mathbf{d}_{\text{eq}}(\mathbf{x}_0^{(k)}, p_{\text{LV}}, T_a)$ (Fig. 2, bottom-left) and we set $\mathbf{x}_0^{(k+1)} = \widetilde{\mathbf{x}} - \mathbf{d}^{(k)}$ (Fig. 2, bottom-right), stopping when the difference between two consecutive iterations is sufficiently small.

The whole procedure is reported in Algorithm 1, where we denote the function that solves problem (18) on the geometry with coordinates $\mathbf{x}_0$ as STEADYSTATEMECHANICS. The latter function solves the nonlinear system by means of the Newton method. In case the Newton iterations do not reach convergence (according to a criterion based both on the residual and on the difference between consecutive iterations), it returns a flag to indicate failure of the algorithm. More precisely, the function signature reads

$$(\texttt{converged\_SSM}, \mathbf{d}) = \textsc{SteadyStateMechanics}(\mathbf{x}_0, p_{\text{LV}}, T_a)$$

where, in case of convergence, `converged_SSM` is true and $\mathbf{d} = \mathbf{d}_{\text{eq}}(\mathbf{x}_0, p_{\text{LV}}, T_a)$, while in case of non convergence `converged_SSM` is false and $\mathbf{d}$ is not used. We remark that an algorithm similar to Algorithm 1 is presented in [43].

Algorithm 1 can be interpreted as a fixed-point iteration scheme. Indeed, the fixed-point $\mathbf{x}_0$ of the map defined by the Algorithm 1 iteration satisfies $\mathbf{x}_0 = \widetilde{\mathbf{x}} - \mathbf{d}_{\text{eq}}(\mathbf{x}_0, p_{\text{LV}}, T_a)$. The fixed-point iterations of Algorithm 1 are sketched in Fig. 3. The solution is obtained as the intersection between the line $\widetilde{\mathbf{x}} = \mathbf{x}_0 + \mathbf{d}$ and the manifold $\mathbf{d} = \mathbf{d}_{\text{eq}}(\mathbf{x}_0, p_{\text{LV}}, T_a)$. The algorithm proceeds iteratively in the space $(\mathbf{x}_0, \mathbf{d})$: the first variable is updated by the fixed-point iterations (horizontal axis of Fig. 3), while the second variable is updated by Newton iterations (vertical axis of Fig. 3).

However, the implementation shown in Algorithm 1 has several limitations when applied to realistic heart geometries and to highly nonlinear constitutive laws, such as in the case of cardiac
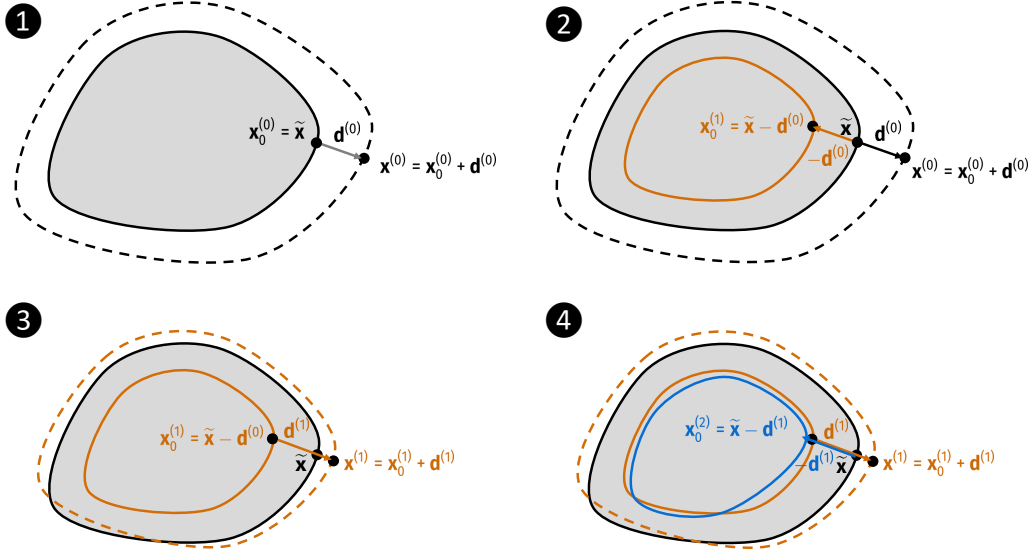
Figure 2: Visual representation of the basic version of the fixed-point algorithm (see Algorithm 1).

Algorithm 1 Reference configuration recovery (basic version)

---

parameters: $k_{\max}, \epsilon_{\text{tol}}$
output: converged_RCB, $\mathbf{x}_0$

procedure REFERENCECONFIGURATIONBASE($\widetilde{\mathbf{x}}, \widetilde{p}, \widetilde{T}_{\text{a}}$)
    $\mathbf{x}_0^{(0)} \leftarrow \widetilde{\mathbf{x}}$
    for $k = 0, \ldots, k_{\max}$ **do**
        (converged_SSM, $\mathbf{d}^{(k)}$) $\leftarrow$ STEADYSTATEMECHANICS($\mathbf{x}_0^{(k)}, \widetilde{p}, \widetilde{T}_{\text{a}}$)
        if not converged_SSM **then**
            return (false, $\mathbf{0}$)　　　　　　　　　　　　▷ Newton method does not converge.
        end if
        $\mathbf{x}^{(k)} \leftarrow \mathbf{x}_0^{(k)} + \mathbf{d}^{(k)}$
        if $\|\mathbf{x}^{(k)} - \widetilde{\mathbf{x}}\| \leq \epsilon_{\text{tol}}\|\mathbf{d}^{(k)}\|$ **then**
            return (true, $\mathbf{x}_0^{(k)}$)　　　　　　　　　　　　▷ Fixed-point converged.
        end if
        $\mathbf{x}_0^{(k+1)} \leftarrow \widetilde{\mathbf{x}} - \mathbf{d}^{(k)}$　　　　　　　　　　　　▷ Fixed-point update.
    end for
    return (false, $\mathbf{0}$)　　　　　　　▷ Maximum number of iterations reached.
end procedure

Figure 3: Representation of the basic version of the reference configuration recovery algorithm (Algorithm 1).

mechanics. In fact, both the Newton method employed in STEADYSTATEMECHANICS and the fixed-point scheme employed in REFERENCECONFIGURATIONBASE do not converge if the initial guess is not "sufficiently" close to the solution. In particular, our experience revealed that the attraction basin of the fixed-point iterations scheme gets smaller and smaller as the value of p_{LV} grows, which makes the base formulation of Algorithm 1 not suitable for realistic cardiac geometries.

To tackle the above criticisms, we improved Algorithm 1 by increasing its robustness through several strategies. We propose in Algorithm 2 the enhanced version of Algorithm 1. First, we introduced a relaxation parameter $\alpha \in (0,1]$ in the fixed-point iteration, by rewriting it as $\mathbf{x}_0^{(k)} \leftarrow \mathbf{x}_0^{(k-1)} + \alpha(\widetilde{\mathbf{x}} - \mathbf{x}^{(k-1)})$. We update α adaptively as written in Algorithm 3: in case of non-convergence of the Newton iterations, we repeat the last fixed-point iteration for a smaller value of α; in case of convergence, we increase it in the next iteration. Moreover, the fixed-point iterations are nested inside an outer loop, in which we progressively increase the value of p_{LV} and T_{a} until the target value $\widetilde{p}$ and $\widetilde{T}_{\text{a}}$ is reached (continuation method). In this outer loop, we adaptively change the step $\omega^{(k)}$, by decreasing it in case of failure of the inner fixed-point loop and increasing it in case of success.

We remark that after each failure of either STEADYSTATEMECHANICS or FIXEDPOINT function, we reset the value of $\mathbf{d}$ to the last solution of Eq. (18). This ensures the success of such continuation strategy, by which we move in the space $(\mathbf{x}_0, \mathbf{d}, \omega)$ staying close to the intersection of the curves $\mathbf{d} = \widetilde{\mathbf{x}} - \mathbf{x}_0$ and $\mathbf{d} = \mathbf{d}_{\text{eq}}(\mathbf{x}_0, \omega p_{\text{LV}}, \omega T_{\text{a}})$ (see Fig. 4).

4.2 Projection of the reference configuration from a coarser mesh

The procedure to recover the reference configuration (Algorithm 2) implies to numerically solve the elastostatic problem of Eq. (18) multiple times, until the fixed point algorithm converges. According to our experience, this procedure can be very computational demanding in cardiac applications, especially if realistic human heart geometries and fine representations of the computational domain need to be considered. To overcome this issue, the reference configuration recovery algorithm can be run on a coarser mesh compared to the couple of nested meshes $\mathcal{T}_{\text{h}_2}$-$\mathcal{T}_{\text{h}_1}$ used for the electromechanical model. A natural choice can be to exploit once again the efficient hierarchical octree structure, but starting from a coarser level h_3, generating accordingly the triad of nested meshes $\mathcal{T}_{\text{h}_3}$-$\mathcal{T}_{\text{h}_2}$-$\mathcal{T}_{\text{h}_1}$. However, a further level of coarsening would imply a loss of geometric accuracy that is propagated also into the finer couple of meshes $\mathcal{T}_{\text{h}_2}$-$\mathcal{T}_{\text{h}_1}$, affecting the electromechanical model.

14

Algorithm 2 Reference configuration recovery (enhanced version)

---

parameters: $k_{\max}, \epsilon_{\text{tol}}^{\text{ramp}}, \epsilon_{\text{tol}}^{\text{final}}, \gamma_\omega^+, \gamma_\omega^-, \Delta\omega_{\max}$
output: `converged_RC`, $\mathbf{x}_0$

procedure REFERENCECONFIGURATION$(\widetilde{\mathbf{x}}, \widetilde{p}, \widetilde{T}_{\mathrm{a}})$
    $\mathbf{x}_0^{(0)} \leftarrow \widetilde{\mathbf{x}}$
    $\omega^{(0)} \leftarrow 0$
    $\Delta\omega \leftarrow \Delta\omega_{\max}$
    for $k = 0, \ldots, k_{\max}$ **do**
        $\omega^{(k)} \leftarrow \min(\omega^{(k-1)} + \Delta\omega, 1)$
        if $\omega^{(k)} = 1$ **then**
            $\epsilon_{\text{tol}} \leftarrow \epsilon_{\text{tol}}^{\text{final}}$
        else
            $\epsilon_{\text{tol}} \leftarrow \epsilon_{\text{tol}}^{\text{ramp}}$
        end if
        $(\texttt{converged\_FP}, \mathbf{x}_0^{(k)}) \leftarrow$ FIXEDPOINT$(\widetilde{\mathbf{x}}, \omega^{(k)}\widetilde{p}, \omega^{(k)}\widetilde{T}_{\mathrm{a}}\mathbf{x}_0^{(k-1)}, \epsilon_{\text{tol}})$
        if `converged_FP` **then**
            if $\omega^{(k)} = 1$ **then**
                return (true, $\mathbf{x}_0^{(k)}$) $\triangleright$ Ramp converged.
            end if
            $\Delta\omega \leftarrow \min(\gamma_\omega^+ \Delta\omega, \Delta\omega_{\max})$
        else
            $\Delta\omega \leftarrow \gamma_\omega^- \Delta\omega$
        end if
    end for
    return (false, $\mathbf{0}$) $\triangleright$ Maximum number of iterations reached.
end procedure

Algorithm 3 Inner fixed-point loop of the reference configuration recovery algorithm

---

parameters: $k_{\max}, \alpha_{\min}, \alpha_{\max}, \gamma_\alpha^+, \gamma_\alpha^-$
output: $\texttt{converged\_FP}, \mathbf{x}_0$

procedure FIXEDPOINT$(\widetilde{\mathbf{x}}, p_{\mathrm{LV}}, T_{\mathrm{a}}, \mathbf{x}_0, \epsilon_{\mathrm{tol}})$
    $\mathbf{x}_0^{(0)} \leftarrow \mathbf{x}_0$
    $\alpha \leftarrow \alpha_{\max}$
    $(\texttt{converged\_SSM}, \mathbf{d}^{(0)}) \leftarrow$ STEADYSTATEMECHANICS$(\mathbf{x}_0^{(0)}, p_{\mathrm{LV}}, T_{\mathrm{a}})$
    if not $\texttt{converged\_SSM}$ **then**
        return (false, $\mathbf{0}$)
    end if
    for $k = 0, \ldots, k_{\max}$ **do**
        $\mathbf{x}_0^{(k)} \leftarrow \mathbf{x}_0^{(k-1)} + \alpha(\widetilde{\mathbf{x}} - \mathbf{x}^{(k-1)})$ $\triangleright$ Fixed-point update.
        $(\texttt{converged\_SSM}, \mathbf{d}^{(k)}) \leftarrow$ STEADYSTATEMECHANICS$(\mathbf{x}_0^{(k)}, \widetilde{p}, \widetilde{T}_{\mathrm{a}})$
        if $\texttt{converged\_SSM}$ **then**
            $\mathbf{x}^{(k)} \leftarrow \mathbf{x}_0^{(k)} + \mathbf{d}^{(k)}$
            if $\|\mathbf{x}^{(k)} - \widetilde{\mathbf{x}}\| \leq \epsilon_{\mathrm{tol}} \|\mathbf{d}^{(k)}\|$ **then**
                return (true, $\mathbf{x}_0^{(k)}$) $\triangleright$ Fixed-point converged.
            end if
            $\alpha \leftarrow \min(\gamma_\alpha^+ \alpha, \alpha_{\max})$
        else
            $\alpha \leftarrow \max(\gamma_\alpha^- \alpha, \alpha_{\min})$
        end if
    end for
    return (false, $\mathbf{0}$) $\triangleright$ Maximum number of iterations reached.
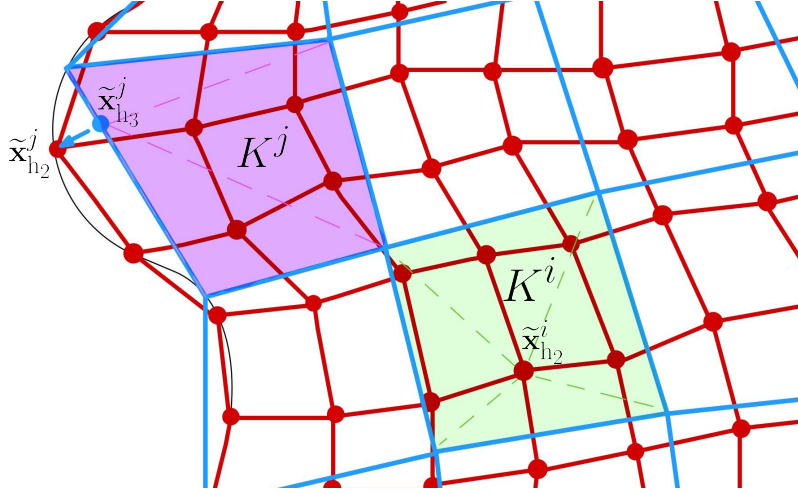**end procedure**

Figure 4: Representation of the enhanced version of the reference configuration recovery algorithm (see Algorithms 2 and 3).

For this reason, we propose a projection technique which enables to map to the reference configuration Ω_0 from a coarser non-nested mesh. This mesh - named $\mathcal{T}_{\widetilde{h}_3}$ - can be independently generated with a mesh size $\widetilde{h}_3$ such that $h_3 < \widetilde{h}_3 < h_2$. Such strategy provides the advantages of having more flexibility on the choice of the mesh size $\widetilde{h}_3$ and of preserving the geometric accuracy for the electromechanical meshes $\mathcal{T}_{h_2}$-$\mathcal{T}_{h_1}$.

The complete procedure consists of the following steps:

1. generate two non-nested computational meshes $\mathcal{T}_{h_2}$ and $\mathcal{T}_{\widetilde{h}_3}$ from the deformed configuration $\widetilde{\Omega}$, i.e. the one reconstructed from the medical images, such that $h_2 < \widetilde{h}_3$. The former is characterized by the target mesh size for the mechanical simulation.

2. solve the reference configuration recovery (Algorithm 2) on the coarser mesh $\mathcal{T}_{\widetilde{h}_3}$, obtaining the displacement field $\mathbf{d}_{\widetilde{h}_3}$;

3. project the displacement $\mathbf{d}_{\widetilde{h}_3}$ on the finer mesh $\mathcal{T}_{h_2}$ obtaining the field $\widehat{\mathbf{d}}_{h_2}$, which approximates the displacement $\mathbf{d}_{h_2}$, i.e. the one that would be computed if the reference configuration recovery (Algorithm 2) is applied directly on the finer mesh $\mathcal{T}_{h_2}$;

4. move each vertex of the mesh $\mathcal{T}_{h_2}$ according to $\mathbf{x}_0 = \widetilde{\mathbf{x}} - \widehat{\mathbf{d}}_{h_2}$, recovering the computational mesh $\mathcal{T}_{h_2}$ that describes the reference configuration Ω_0;

5. hierarchically refine $\mathcal{T}_{h_2}$ to generate the fine mesh $\mathcal{T}_{h_1}$ for the electrophysiology.

We remark that the projection (step 3) is necessary despite both $\mathcal{T}_{h_2}$ and $\mathcal{T}_{\widetilde{h}_3}$ describe the same domain $\widetilde{\Omega}$. Indeed, in practice, their boundaries do not match, since they are independent polygonal surfaces made of piecewise linear elements. Thus, some vertexes of $\mathcal{T}_{h_2}$ can lie outside $\mathcal{T}_{\widetilde{h}_3}$, as

17

Figure 5: A sketch of the projection procedure from a coarse mesh $\mathcal{T}_{\widetilde{h}_3}$ (in blue) into a finer mesh $\mathcal{T}_{h_2}$ (in red): on the internal points $\widetilde{\mathbf{x}}_{h_2}^i \in \mathcal{T}_{h_2}$ we recover the value exploiting the basis functions of the element $K^i \in \mathcal{T}_{h_3}$ (in green); on the external points $\widetilde{\mathbf{x}}_{h_2}^j \in \mathcal{T}_{h_2}$ we project the value of the closest point $\widetilde{\mathbf{x}}_{h_3}^j \in \mathcal{T}_{\widetilde{h}_3}$ evaluated on the closest element $K^j \in \mathcal{T}_{\widetilde{h}_3}$ (in purple).

illustrated in the 2D sketch of Fig. 5. As a consequence, to recover $\widehat{\mathbf{d}}_{h_2}$ in all the vertexes of the mesh $\mathcal{T}_{h_2}$, we proceed in a different way if these vertexes lie inside or outside $\mathcal{T}_{\widetilde{h}_3}$. In particular:

- for the internal vertexes – denoted as $\widetilde{\mathbf{x}}_{h_2}^i$, $i = 0, 1, 2, \ldots$ – it is sufficient to find the element $K^i \in \mathcal{T}_{h_3}$ such that $K^i \ni \widetilde{\mathbf{x}}_{h_2}^i$ and evaluate $\mathbf{d}_{h_3}(\widetilde{\mathbf{x}}_{h_2}^i)$ exploiting its Finite Element expansion on K^i (see Fig. 5, green element);

- conversely, for each external point – denoted as $\widetilde{\mathbf{x}}_{h_2}^j$, $j = 0, 1, 2, \ldots$ – more sub-steps are necessary, (see Fig. 5, purple element):

    1. we find the closest element $K^j \in \mathcal{T}_{\widetilde{h}_3}$ from the external point $\widetilde{\mathbf{x}}_{h_2}^j$;

    2. on K^j we find the closest point $\widetilde{\mathbf{x}}_{h_3}^j$ to $\widetilde{\mathbf{x}}_{h_2}^j$;

    3. we evaluate $\mathbf{d}_{h_3}(\widetilde{\mathbf{x}}_{h_3}^j)$ projecting the resulting value into the external point $\widetilde{\mathbf{x}}_{h_2}^j$.

From the implementation point of view, this projection is performed by exploiting the *VTK* library [53] in our Finite Element library `life`ˣ (`https://lifex.gitlab.io/lifex`). *VTK* filtering utilities allow to locate all internal points, which are the majority, in a really fast way. Moreover, the *VTK* library is efficient in performing closest points interpolation, leading to a very fast projection procedure. However, we remark that the hierarchical octree structure [1, 11] still remains more effective for electromechanical simulations, where the exchange of information between (nested) meshes occurs at each time step. Indeed, the projection presented here is intended to be a single pre-processing step to be performed before the non-stationary electromechanical simulation. An example of this projection in a left ventricle is shown in Sect. 5.1.

	Number of elements	Number of vertices	h_{mean}
Mesh1			
Electrophysiology	1'008'896	1'046'641	1 mm
Activation/Mechanics	15'764	19'099	4 mm
Mesh2			
Electrophysiology	2'590'464	2'663'817	0.75 mm
Activation/Mechanics	40'476	47'529	3 mm
Mesh3			
Electrophysiology	8'939'776	9'116'741	0.5 mm
Activation/Mechanics	139'684	159'149	2 mm

Table 1: Details of the Zygote left ventricle meshes used for the numerical results (Sec. 5). For each of the three configuration (Mesh1, Mesh2, and Mesh3) two nested meshes are generated: the finer - used for the Electrophysiology model - is obtained by two recursive splitting of each element of the coarser - adopted for both the Activation and Mechanics models.

5 Numerical results

We present the numerical simulations of left ventricle electromechanics. The geometry is preprocessed from the Zygote Solid 3D heart model [30], which represents the 50th percentile of a healthy caucasian male in the U.S., reconstructed from an high resolution computed tomography scan. In particular, in Tab. 1 we list the three configurations of the computational meshes under consideration, from the coarsest (Mesh1) to the finest one (Mesh3). For all the cases, we employ for the electrophysiology problem a mesh size four times smaller than for the mechanical problem. The listed meshes were generated using vmtk (www.vmtk.org) [2], in particular by exploiting the recently proposed tools for cardiac mesh generation [18].

First, we show an example of projection of the reference configuration from a coarser mesh, exploiting the method detailed in Sec. 4.2. Then, we provide a grid refinement study, by considering increasingly refined meshes, employing in all cases different grids for electrophysiology and activation/mechanics. We depict activation maps generated using either $\mathbb{Q}_1$ or $\mathbb{Q}_2$ elements for electrophysiology and only $\mathbb{Q}_1$ element for activation and mechanics: in this way we show that our intergrid transfer operator can handle different Finite Element spaces on different core models, permitting to explore high-order methods in future works. Finally, we show that our model can generate several scenarios according to the chosen parameter sets: specifically, we vary preload, afterload and contractility, and we evaluate the effects of these changes on left ventricle PV loops.

The numerical methods of Sec. 3 have been implemented in life$^{\mathrm{x}}$ (https://lifex.gitlab.io/lifex), a high-performance C++ library developed within the iHEART project[1] and based on the deal.II (https://www.dealii.org) Finite Element core [3]. The numerical simulations have been obtained by employing life$^{\mathrm{x}}$ in a parallel setting, using either a HPC resource available at MOX (48 Intel Xeon ES-2640 CPUs) or the GALILEO supercomputer from Cineca (7 nodes endowed with 36 Intel Xeon E5-2697 v4 2.30 GHz CPUs, for a total number of 252 cores).

In terms of the settings our the numerical simulations, we apply a current $\widetilde{\mathcal{I}}_{\mathrm{app}}(\boldsymbol{x}, t)$, distributed in space as a Gaussian with peak $\widetilde{\mathcal{I}}_{\mathrm{app}}^{\mathrm{max}}$, for a duration of t_{app}, in three different regions of the
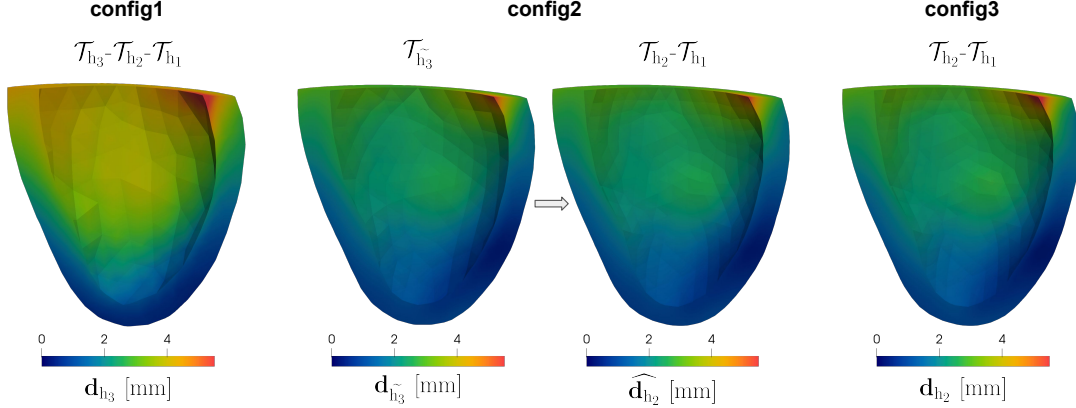
Figure 6: Solving the reference configuration recovery (Algorithm 2) for three different settings, with a common target mesh size for the mechanical simulation ($\mathcal{T}_{h_2}$, $h = 3\,\text{mm}$): in *config1* we apply the recovery algorithm on the coarsest mesh of a triad of hierarchical nested meshes ($\mathcal{T}_{h_3}$, $h = 6\,\text{mm}$); in *config2* the algorithm runs on an independent coarse mesh ($\mathcal{T}_{\tilde{h}_3}$, $h = 4\,\text{mm}$) and the displacement is projected onto $\mathcal{T}_{h_2}$; in *config3* we compute the reference configuration directly on $\mathcal{T}_{h_2}$.

myocardium to trigger the electrical signal in the left ventricle. These regions are located at the central distance between the apex and the base of the ventricle. Even if we do not model the Purkinje network [51, 57], such electrical activation of the tissue that we propose is known to provide meaningful results [21]. We use the Bayer-Blake-Plank-Trayanova algorithm [6, 41] to generate the fibers distribution (field $\mathbf{f}_0$) for our geometry, using $\alpha_{\text{epi}} = -60°$, $\alpha_{\text{endo}} = 60°$, $\beta_{\text{epi}} = 20°$ and $\beta_{\text{endo}} = -20°$. In all numerical tests, we set for simplicity $p_{\text{EX}}(t) = 0$. All the parameters related to the monodomain equation and passive mechanics are reported in Tabs. 2 and 3 respectively. We use a time step $\Delta t_1 = 50\,\mu\text{s}$ for electrophysiology, activation and circulation, whereas mechanics is solved in time with a time step $\Delta t_2 = 250\,\mu\text{s}$. Further information about the parameters of both linear and nonlinear solvers are reported in Appendix B.

5.1 Projection of the reference configuration

We discuss the results of the linear projection of the reference configuration (Sect. 4.2) by running the recovery of reference configuration (Algorithm 2) for the Zygote left ventricle geometry. In particular, we consider the following mesh configurations:

- *config1*: a triad of nested meshes $\mathcal{T}_{h_3}$-$\mathcal{T}_{h_2}$-$\mathcal{T}_{h_1}$; the coarsest one for the recovery algorithm, the intermediate one for mechanics and the finer one for electrophysiology;

- *config2*: an independent coarser mesh $\mathcal{T}_{\tilde{h}_3}$ for the recovery algorithm, and a couple of nested meshes $\mathcal{T}_{h_2}$-$\mathcal{T}_{h_1}$ for the electromechanical model;

- *config3*: a couple of nested meshes $\mathcal{T}_{h_2}$-$\mathcal{T}_{h_1}$ for the electromechanical model, where the mesh $\mathcal{T}_{h_2}$ is used also for the recovery algorithm.

In Fig. 6, we show the results of the recovery algorithm on these three configurations, choosing $h = 3\,\text{mm}$ as target mesh size for mechanics (i.e. $\mathcal{T}_{h_2}$). More in detail, we create an ad-hoc hierarchical triad $\mathcal{T}_{h_3}$ ($h = 6\,\text{mm}$)-$\mathcal{T}_{h_2}$($h = 3\,\text{mm}$)-$\mathcal{T}_{h_1}$($h = 0.75\,\text{mm}$) for *config1*. Instead, *config2*

20

and *config3* exploit some meshes listed in Tab. 1: Mesh2 is selected as the couple $\mathcal{T}_{h_2}$-$\mathcal{T}_{h_1}$, and Mesh1, activation/mechanics as $\mathcal{T}_{\widetilde{h}_3}$.

The displacement field $\mathbf{d}_{h_2}$ of *config3* represents the reference result since it is obtained by running directly the recovery algorithm on $\mathcal{T}_{h_2}$, i.e. the finer mesh that we consider. On the contrary, *config1* presents a large loss of geometric accuracy, which is propagated into the two meshes $\mathcal{T}_{h_2}$-$\mathcal{T}_{h_1}$ of the electromechanical model. Indeed, $\mathcal{T}_{h_3}$ appears too sharp and not realistic, especially at the base and at the apex. Moreover, the displacement field $\mathbf{d}_{h_3}$ computed on $\mathcal{T}_{h_3}$ is qualitative different from the reference one ($\mathbf{d}_{h_2}$, *config3*).

In *config2*, we first compute the displacement field $\mathbf{d}_{\widetilde{h}_3}$ on the independent coarse mesh $\mathcal{T}_{\widetilde{h}_3}$, and then we obtain the projected field $\widehat{\mathbf{d}}_{h_2}$ on $\mathcal{T}_{h_2}$ by exploiting the linear projection procedure (Sect. 4.2). The projected field $\widehat{\mathbf{d}}_{h_2}$ features a good qualitative match with the field $\mathbf{d}_{h_2}$. More quantitatively, we obtain an average error $|\mathbf{d}_{h_2} - \widehat{\mathbf{d}}_{h_2}| = 0.105\,\text{mm}$. This quantity is lower than the standard resolution of cardiac medical images (about $0.5\,\text{mm}$ for high-quality acquisition). Thus, we can consider this error lower than the geometric uncertainty when dealing with patient-specific simulations. Moreover, being $\mathcal{T}_{\widetilde{h}_3}$ made of a considerable lower number of elements if compared to $\mathcal{T}_{h_2}$ (see Tab. 1), *config2* saves a large amount of computational time for the execution of Algorithm 2, compared to *config3*.

Config2 is the cheapest procedure in terms of total computational cost, moreover it presents a negligible error and does not affect the geometrical accuracy of the electromechanical model. We conclude by recommending this setting when a fine patient-specific mesh is employed for the mechanical problem.

5.2 Mesh convergence

To study the results of our numerical model when the space discretization is refined, we run simulations on Mesh1, Mesh2 and Mesh3 of Tab. 1. In Fig. 7 we display the evolution in time of pressure, solid volume, blood volume, and the PV loop, for the three proposed mesh settings. We show that convergence properties in the electromechanical framework are not only influenced by the average diameter h_{mean} [4], but also by the effective number of DOFs of the mesh. Indeed, we see that Mesh2 with $h_{\text{mean}} \approx 0.75$ mm for electrophysiology and $h_{\text{mean}} \approx 3$ mm for activation and mechanics already provides a good convergence of both mechanical and fluid dynamics relevant properties for the left ventricle. Under the assumption of properly distributed mesh elements, the number of DOFs itself can potentially be an indicator to evaluate the accuracy of the discretization for cardiac simulations, unifying approximations coming from different mesh elements (e.g., tetrahedra, hexaedra and prisms) and different techniques, such as Finite Element Method, Spectral Element Method or Isogeometric Analysis [10, 40, 42].

In Fig. 8 we depict the activation maps for the Mesh3 setting in Tab. 1. We use either $\mathbb{Q}_1$ or $\mathbb{Q}_2$ Finite Elements for electrophysiology, and $\mathbb{Q}_1$ for both activation and mechanics. We show one simulation with $\mathbb{Q}_2$ Finite Elements for electrophysiology to underline that our mathematical discretization can be extended to high order methods, which are known to be more suitable than standard FEM for wave propagation problems [10, 40]. We observe that the activation map resulting from $\mathbb{Q}_2$ elements has slightly more pronounced anisotropy of the isochrones. Indeed, the additional DOFs provided by $\mathbb{Q}_2$ elements allow to better outline the different roles of conductivities σ_l, σ_t and σ_n, as shown in [41].
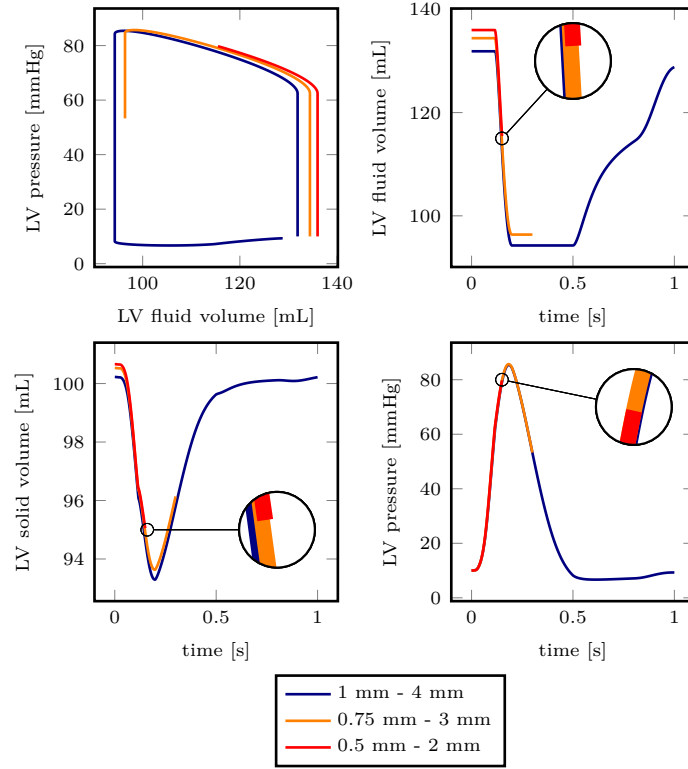
Figure 7: Time evolution of pressure and volumes, and PV loops, related to the Zygote left ventricle, considering different mesh resolutions (Mesh1, Mesh2, Mesh3 of Tab. 1) and $\mathbb{Q}_1$ Finite Element spaces. We report a full heartbeat only for the coarsest mesh (Mesh1).
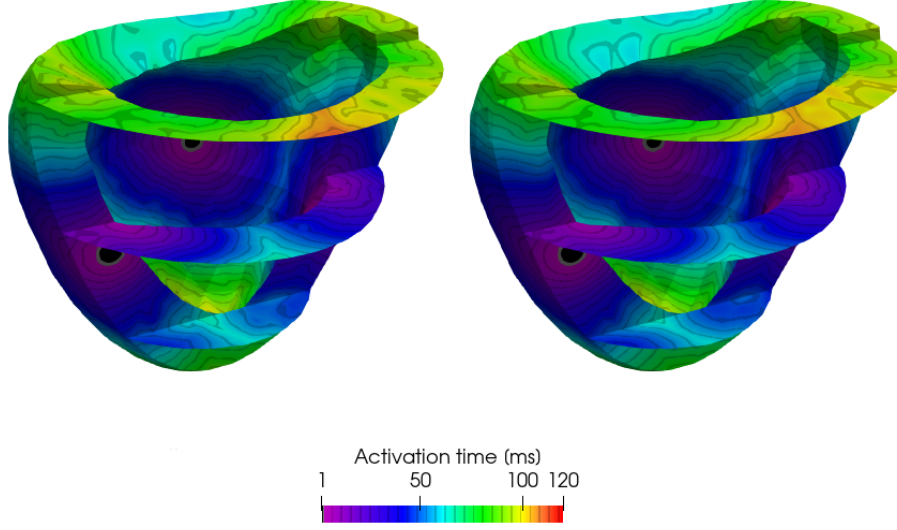
Figure 8: Activation times on the Zygote left ventricle considering different Finite Element spaces ($\mathbb{Q}_1$, 1'046'641 DOFs, on the left, and $\mathbb{Q}_2$, 8'211'745 DOFs, on the right) for electrophysiology.

5.3   Tests Cases varying preload, afterload and contractility

We test the response of our electromechanical model to some scenarios of clinical interest. The parameters of the baseline simulation are reported in Appendix A. Starting from this setting, we consider three physiologically relevant scenarios, aimed at investigating the effects of changes in preload, afterload and myocardial contractility, respectively. In all the cases, we simulate several heartbeats, until we reach a periodic regime, but we report only the PV loops related to the last cycle.

5.3.1   Baseline simulation

In Fig. 10 we report the time evolution of both the transmembrane potential u and the displacement magnitude $|\mathbf{d}|$ in the Zygote left ventricle, considering the first heartbeat of the simulation. We use mesh configuration 2 in Tab. 1. We compute the initial displacement $\mathbf{d}_0$ by inflating Ω_0 until the desired end-diastolic pressure is reached. In practice we accomplish this task by performing a pressure ramp on the quasi-static approximation of the mechanical problem.

In Fig. 11 we depict the space and time evolutions of the axial stresses S_{ff}, S_{ss} and S_{nn}, whereas in Fig. 12 we show the shear stresses S_{fs}, S_{fn} and S_{sf}. We observe that there is no significant generation of stresses (i.e. $S_{\mathrm{ab}} \approx 0$) elsewhere on some specific location at the base, when crossfibers directions are involved. Indeed, the dominant role of fibers and sheets stresses is evident during both the ejection phase (i.e. the second part of systole) and isovolumetric relaxation (i.e. early diastole).

5.3.2   Test Case 1

We change the ventricles preload (i.e. their end-diastolic pressure) by modifying the value of the atrial contractility with respect to the baseline setting of Appendix A. More precisely, we consider
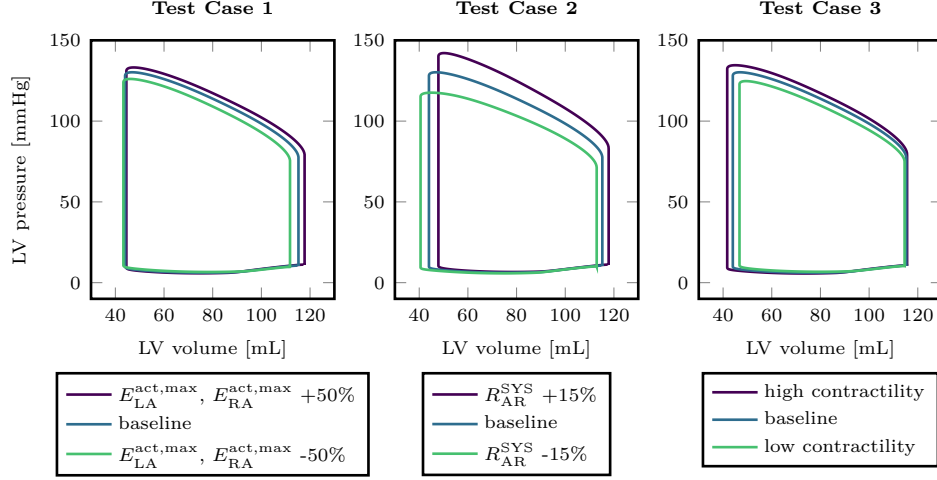
Figure 9: Left ventricle PV loops obtained in the three Test Cases of Sec. 5.3 compared with baseline.

the two cases when $E_{LA}^{act,max}$ and $E_{RA}^{act,max}$ are respectively increased and decreased by 50%. As shown in Fig. 9, the larger the atrial contractility, the more blood is injected in the ventricle, thus increasing preload. Moreover, our electromechanical model predicts a larger stroke volume for a larger preload. This is consistent with the so-called Frank-Starling effect, a self-regulatory mechanism that guarantees the balance between venous return and cardiac output [31].

5.3.3 Test Case 2

We investigate the effects of changing the resistance of the arterial circulation. This corresponds to scenarios of a patient affected by diseases associated with hypertension, such as arteriosclerosis, or to the effect of vasodilator-vasoconstrictor drugs. Starting from the baseline setting, we perform two additional simulations where we respectively increase and decrease by 15% the value of R_{AR}^{SYS}. In both cases, we modify the value of C_{AR}^{SYS} accordingly, so that the product $R_{AR}^{SYS} C_{AR}^{SYS}$, corresponding to the characteristic time constant of the arterial system, is preserved. The results in Fig. 9 show that an increase of the arterial resistance yields larger values of both the aortic valve opening pressure and the maximal LV pressure (hypertensive effect).

5.3.4 Test Case 3

We consider the response of our electromechanical model to either positive or negative change to inotropic state of the muscle, whose effect is that of increasing and, respectively, decreasing the myocardial contractility. Specifically, starting from the baseline, we first increment and then decrement the atrial contractility by 35% ($E_{LA}^{act,max}$ and $E_{RA}^{act,max}$) by 35% and the ventricular contractility (T_a^{max} and $E_{RV}^{act,max}$). The results in Fig. 9 show that an increase in myocardial contractility generates an increase of both the maximal LV pressure and the stroke volume, but it does not affect preload. As a matter of fact, the end diastolic pressure is unaltered.
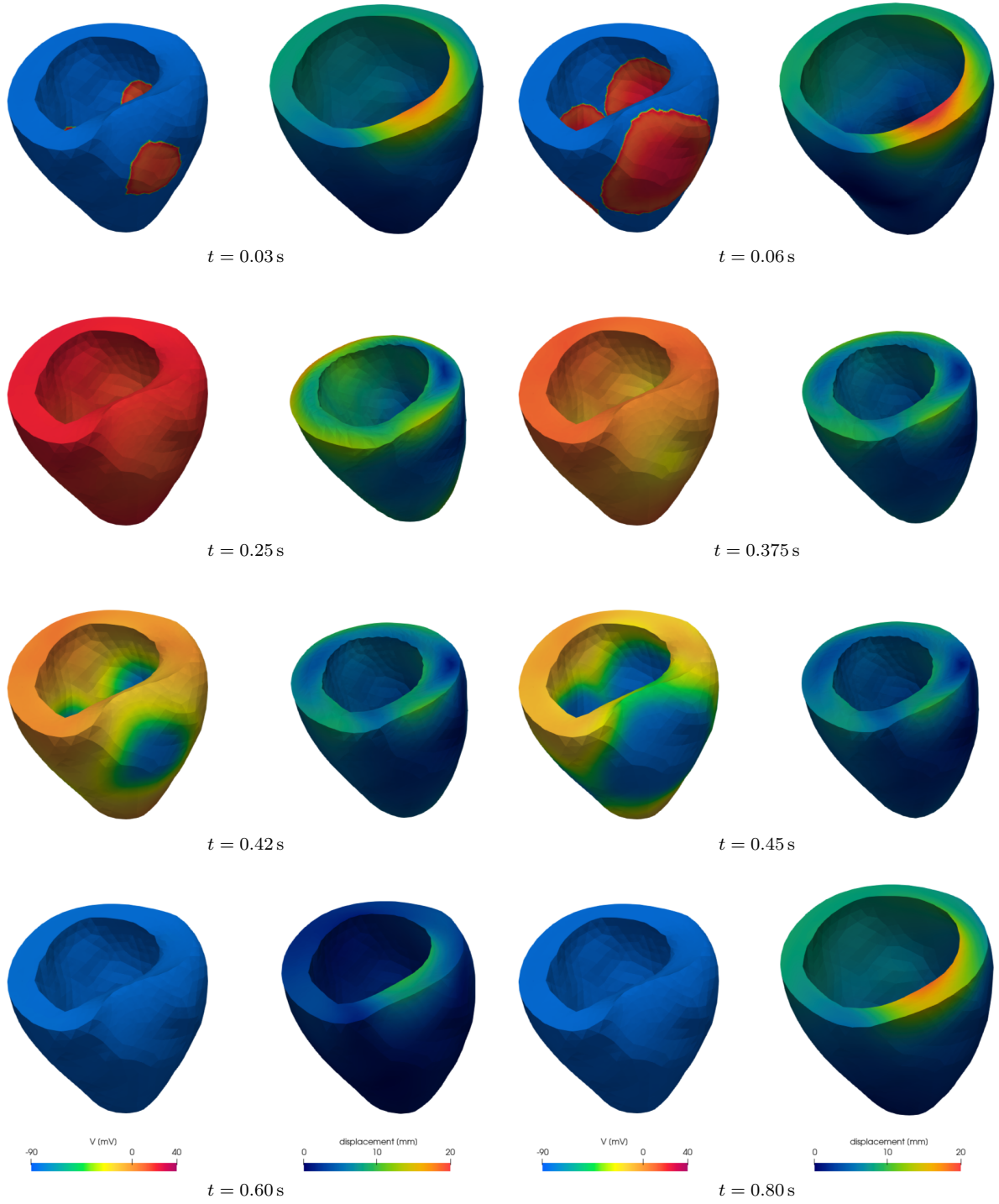
24

Figure 10: Evolution of the transmembrane potential u and the displacement magnitude $|\mathbf{d}|$ in the Zygote left ventricle over time. The right view of each picture is warped by the displacement vector. Conversely, the potential u is displayed on the reference configuration Ω_0.
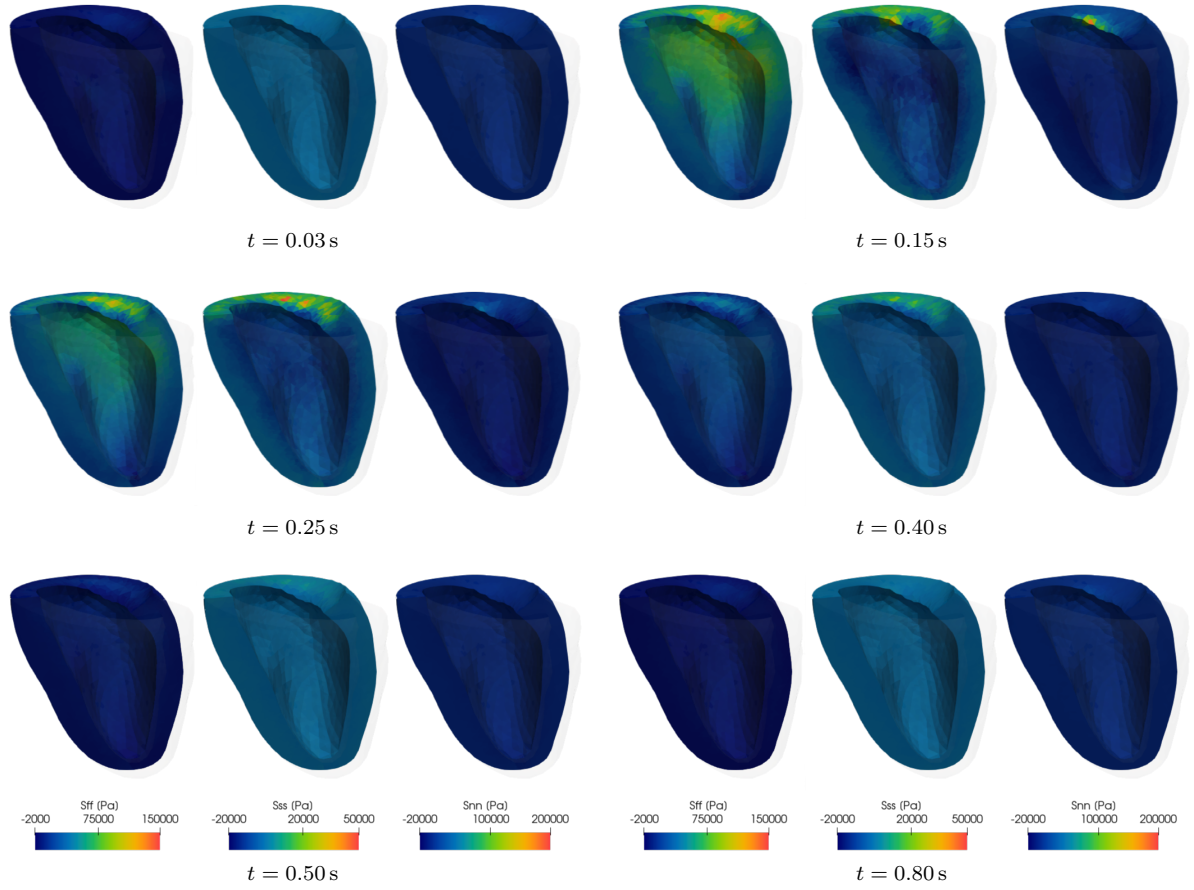
25

Figure 11: Evolution of S_{ff}, S_{ss}, S_{nn} during the first heartbeat of the simulation.
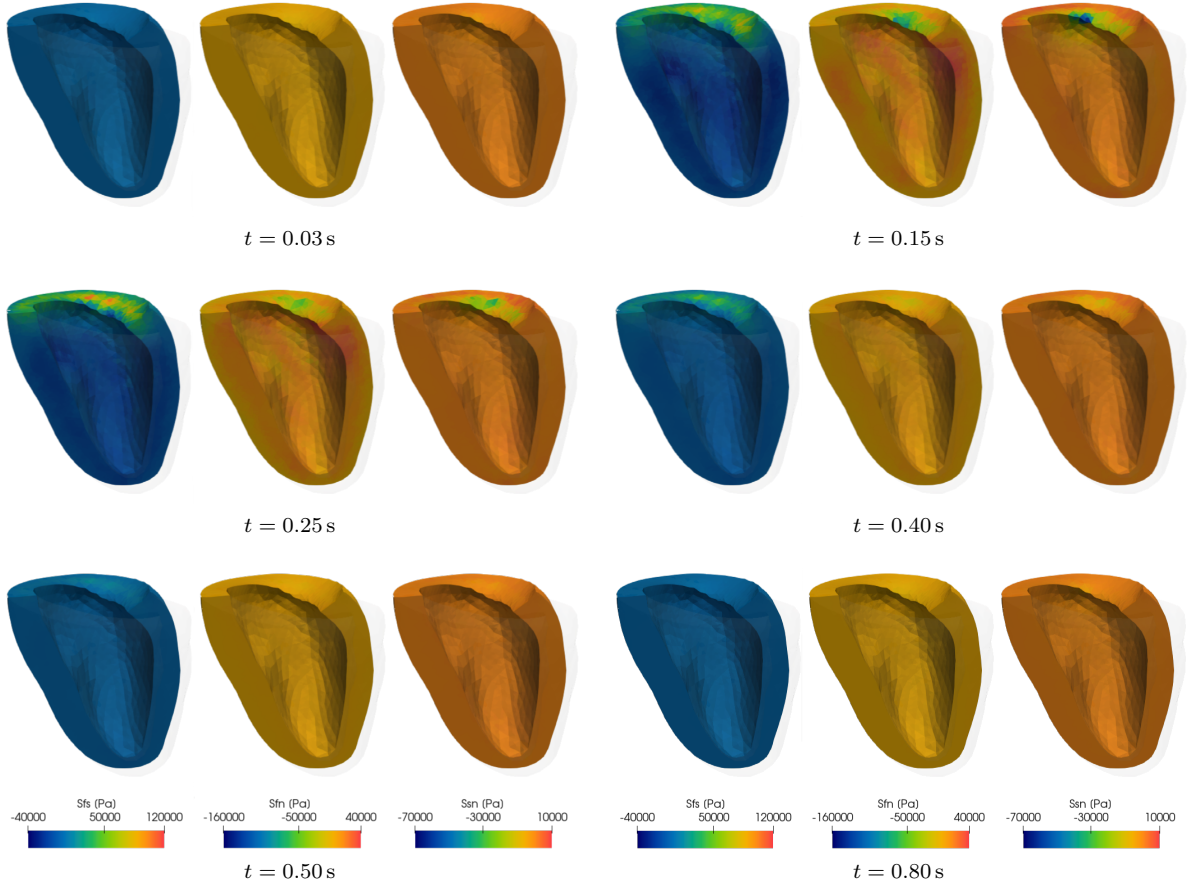
Figure 12: Evolution of S_{fs}, S_{fn} and S_{sn} during the first heartbeat of the simulation.

6 Conclusions

In this paper, we presented a new numerical model for simulating cardiac electromechanics. The numerical schemes here proposed are aimed at coupling, in a computationally efficient and accurate manner, mathematical models that are significantly biophysically detailed. Our computational framework is based on two main pillars, namely (1) IMEX schemes to approximate the single core models; (2) a fully segregated coupling of the different physics. The advantages yielded by these features are the following.

- We designed IMEX schemes to minimize the number of nonlinear systems that need to be solved, using implicit solvers only for those core models that would otherwise lead to severe CFL restriction on the time step. In particular, our numerical scheme allows to update the variables of the ionic and activation models without the need of solving any algebraic system (neither nonlinear nor linear), and it allows to update the electrical potential by solving a single linear system at each time iteration. The unique nonlinear system of our mathematical model is associated with cardiac mechanics, for which explicit or semi-implicit schemes are unstable, unless a very fine time step is employed, because of to the strong nonlinearities contained in the constitutive law [21].

- We developed a novel scheme to couple the 3D electromechanical model with a 0D circulation model in a fully segregated manner. Our scheme does not present the *balloon dilemma* issue [28] – which affects segregated schemes wherein the displacement update is not aware of the incompressibility constraint of the enclosed fluid, thus possibly leading to the failure of the scheme [5, 25] – by the introduction of a volumetric constraint on the solution of the mechanical problem. In this way, the cavity pressure can be reinterpreted as a Lagrange multiplier associated with the constraint, that enforces the coupling between the 0D circulation model and the 3D electromechanics model. At the algebraic level, we end up with a saddle-point problem, that we solve by means of Schur complement reduction [7].

- The fully segregated scheme proposed in this paper allows to employ different discretizations in space and time to approximate the variables associated with the different physics. This better reflects, at the numerical level, the typical space and time scales characterizing the associated physical phenomena. Specifically, we employ coarser spatial and temporal discretizations for the elastodynamics problem, which is characterized by larger characteristic temporal and spatial scales than the electrophysiological problem [52]. Moreover, it is the most demanding core model from the computational viewpoint, as it involves the solution of a nonlinear system and the time-consuming Jacobian matrix assembly [22].

Regarding the latter aspect, we employ a parallel and flexible intergrid transfer operator [1, 23, 52] that permits to interpolate the solution of a core model among nested meshes and between possibly different FEM spaces [1]. As a matter of fact, among the numerical results presented in this paper, we considered the case where we employ different polynomial orders for the electrophysiology variables, on one hand, and for the activation and mechanics variables, on the other hand. This can be considered as a first step towards possible investigation of high-order methods in our cardiac electromechanics computational framework. Indeed, wave propagation problems, such as the one arising in cardiac electrophysiology, can be more suitably represented using high-order basis functions than classical FEM [10, 40].

As a further novel contribution, we presented an algorithm to reconstruct the reference (i.e. stress-free) configuration of the left ventricle starting from a stressed configuration coming from

imaging, by solving a suitable inverse problem. Determining such configuration is essential to correctly initialize electromechanical simulations, which is especially useful in patient-specific scenarios where the end diastolic pressure and/or the end diastolic volume are possibly known.

Finally, we presented several simulations varying different parameters of the model, to affect preload, afterload and contractility, thus investigating the response of our model to scenarios of clinical interest. Our model correctly reproduces the increase of stroke volume as a consequence of increased preload, coherently with the Frank-Starling law [31], thus guaranteeing the matching between the venous return and the cardiac output.

Albeit in this paper we focused on left ventricle simulations, the 3D-0D coupling that we describe (along with its numerical treatment) can be extended to a full four chambers representation of the human heart, a model that we will pursue in forthcoming publications. Similarly, the pipeline to reconstruct the reference geometry can be generalized and employed for both atria and ventricles.

Acknowledgements

Appendices

A Model parameters

We provide the full list of parameters adopted for the simulations referred as baseline in Sec. 5.3. Specifically, Tab. 2 contains the parameters related to the electrophysiological model, Tab. 3 those related to the mechanical model and, finally, Tab. 4 contains the parameters of the circulation model. For the TTP06 model, we adopt the parameters reported in the original paper (for M cells) [55]. For the RDQ18 model, we employ the parameters of the original paper [46].

Variable	Value	Unit	Variable	Value	Unit
Conductivity tensor			**Applied current**		
σ_l	$0.7643 \cdot 10^{-3}$	$m^2\,s^{-1}$	$\widetilde{\mathcal{I}}_{app}^{max}$	35	$V\,s^{-1}$
σ_t	$0.3494 \cdot 10^{-3}$	$m^2\,s^{-1}$	t_{app}	$3 \cdot 10^{-3}$	s
σ_n	$0.1125 \cdot 10^{-3}$	$m^2\,s^{-1}$			

Table 2: Parameters of the electrophysiological model.

Variable	Value	Unit	Variable	Value	Unit
Constitutive law			**Boundary conditions**		
B	$50 \cdot 10^3$	Pa	$K_{\perp}^{\mathrm{epi}}$	$2 \cdot 10^5$	$\mathrm{Pa\,m^{-1}}$
C	$0.88 \cdot 10^3$	Pa	$K_{\parallel}^{\mathrm{epi}}$	$2 \cdot 10^4$	$\mathrm{Pa\,m^{-1}}$
b_{ff}	8	–	$C_{\perp}^{\mathrm{epi}}$	$2 \cdot 10^4$	$\mathrm{Pa\,s\,m^{-1}}$
b_{ss}	6	–	$C_{\parallel}^{\mathrm{epi}}$	$2 \cdot 10^3$	$\mathrm{Pa\,s\,m^{-1}}$
b_{nn}	3	–			
b_{fs}	12	–	**Activation**		
b_{fn}	3	–	$T_{\mathrm{a}}^{\mathrm{max}}$	$180 \cdot 10^3$	Pa
b_{sn}	3	–	SL_0	2	μm
ρ_{s}	10^3	$\mathrm{kg\,m^{-3}}$			

Table 3: Parameters of the mechanical model.

Variable	Value	Unit	Variable	Value	Unit
External circulation			**Cardiac chambers**		
$R_{\mathrm{AR}}^{\mathrm{SYS}}$	0.8	$\mathrm{mmHg\,s\,mL^{-1}}$	$E_{\mathrm{LA}}^{\mathrm{pass}}$	0.09	$\mathrm{mmHg\,mL^{-1}}$
$R_{\mathrm{AR}}^{\mathrm{PUL}}$	0.1625	$\mathrm{mmHg\,s\,mL^{-1}}$	$E_{\mathrm{RA}}^{\mathrm{pass}}$	0.07	$\mathrm{mmHg\,mL^{-1}}$
$R_{\mathrm{VEN}}^{\mathrm{SYS}}$	0.26	$\mathrm{mmHg\,s\,mL^{-1}}$	$E_{\mathrm{RV}}^{\mathrm{pass}}$	0.05	$\mathrm{mmHg\,mL^{-1}}$
$R_{\mathrm{VEN}}^{\mathrm{PUL}}$	0.1625	$\mathrm{mmHg\,s\,mL^{-1}}$	$E_{\mathrm{LA}}^{\mathrm{act,max}}$	0.07	$\mathrm{mmHg\,mL^{-1}}$
$C_{\mathrm{AR}}^{\mathrm{SYS}}$	1.2	$\mathrm{mL\,mmHg^{-1}}$	$E_{\mathrm{RA}}^{\mathrm{act,max}}$	0.06	$\mathrm{mmHg\,mL^{-1}}$
$C_{\mathrm{AR}}^{\mathrm{PUL}}$	10.0	$\mathrm{mL\,mmHg^{-1}}$	$E_{\mathrm{RV}}^{\mathrm{act,max}}$	0.55	$\mathrm{mmHg\,mL^{-1}}$
$C_{\mathrm{VEN}}^{\mathrm{SYS}}$	60.0	$\mathrm{mL\,mmHg^{-1}}$	$V_{0,\mathrm{LA}}$	4.0	mL
$C_{\mathrm{VEN}}^{\mathrm{PUL}}$	16.0	$\mathrm{mL\,mmHg^{-1}}$	$V_{0,\mathrm{RA}}$	4.0	mL
$L_{\mathrm{AR}}^{\mathrm{SYS}}$	$5 \cdot 10^{-3}$	$\mathrm{mmHg\,s^2\,mL^{-1}}$	$V_{0,\mathrm{RV}}$	10.0	mL
$L_{\mathrm{AR}}^{\mathrm{PUL}}$	$5 \cdot 10^{-4}$	$\mathrm{mmHg\,s^2\,mL^{-1}}$	**Cardiac valves**		
$L_{\mathrm{VEN}}^{\mathrm{SYS}}$	$5 \cdot 10^{-4}$	$\mathrm{mmHg\,s^2\,mL^{-1}}$	$R_{\min}$	0.0075	$\mathrm{mmHg\,s\,mL^{-1}}$
$L_{\mathrm{VEN}}^{\mathrm{PUL}}$	$5 \cdot 10^{-4}$	$\mathrm{mmHg\,s^2\,mL^{-1}}$	$R_{\max}$	75006.2	$\mathrm{mmHg\,s\,mL^{-1}}$

Table 4: Parameters of the circulation model. We always consider a heartbeat period $T = 0.8\,\mathrm{s}$.

B   Parameters of the linear and nonlinear solvers

We report the setting used for the linear (Tab. 5) and nonlinear (Tab. 6) solvers to produce the results shown in this paper.

Physics/Fields	Linear solver	Abs. tol.
Monodomain model	CG	10^{-10}
Activation	GMRES	10^{-10}
Mechanics	GMRES	10^{-8}

Table 5: Tolerances about the linear solver for the different physics.

Physics/Fields	nonlinear solver	Rel. tol.	Abs. tol.
Mechanics	Quasi-Newton	10^{-10}	10^{-8}
Reference configuration	Newton	10^{-10}	10^{-8}

Table 6: Tolerances about the nonlinear solver for the mechanical problem.

References

[1] P. C. Africa. "Scalable adaptive simulation of organic thin-film transistors". PhD thesis. Politecnico di Milano, 2019.

[2] L. Antiga, M. Piccinelli, L. Botti, B. Ene-Iordache, A. Remuzzi, and D. A. Steinman. "An image-based modeling framework for patient-specific computational hemodynamics". In: *Medical & biological engineering & computing* 46.11 (2008), p. 1097.

[3] D. Arndt, W. Bangerth, B. Blais, T. C. Clevenger, M. Fehling, A. V. Grayver, T. Heister, L. Heltai, M. Kronbichler, M. Maier, P. Munch, J.-P. Pelteret, R. Rastak, I. Thomas, B. Turcksin, Z. Wang, and D. Wells. "The `deal.II` Library, Version 9.2". In: *Journal of Numerical Mathematics* 28.3 (2020), pp. 131–146.

[4] C. M. Augustin, A. Neic, M. Liebmann, and et al. "Anatomically accurate high resolution modeling of human whole heart electromechanics: a strongly scalable algebraic multigrid solver method for nonlinear deformation". In: *Journal of Computational Physics* 305 (2016), pp. 622–646.

[5] Y. Bao, A. Donev, and B. E. Griffith. "An immersed boundary method with divergence-free velocity interpolation". In: *Journal of Computational Physics* 347 (2017), pp. 183–206.

[6] J. D. Bayer, R. C. Blake, G. Plank, and N. Trayanova. "A novel rule-based algorithm for assigning myocardial fiber orientation to computational heart models". In: *Annals of Biomedical Engineering* 40 (2012), pp. 2243–2254.

[7] M. Benzi, G. H. Golub, and J. Liesen. "Numerical solution of saddle point problems". In: *Acta numerica* 14 (2005), pp. 1–137.

[8] P. J. Blanco and R. A. Feijóo. "A 3D-1D-0D Computational Model for the Entire Cardiovascular System". In: *Computational Mechanics* 24 (2010), pp. 5887–5911.

[9] D. Boffi, L. F. Pavarino, G. Rozza, S. Scacchi, and C. Vergara. *Mathematical and Numerical Modeling of the Cardiovascular System and Applications*. Vol. 16. Springer, 2018.

[10] M. Bucelli, M. Salvador, L. Dede', and A. Quarteroni. "Multipatch Isogeometric Analysis for Electrophysiology: Simulation in a Human Heart". In: *MOX Report, Politecnico di Milano* 46 (2020).

[11] C. Burstedde, L. C. Wilcox, and O. Ghattas. "`p4est`: Scalable Algorithms for Parallel Adaptive Mesh Refinement on Forests of Octrees". In: *SIAM Journal on Scientific Computing* 33.3 (2011), pp. 1103–1133.

[12] R. Chabiniok, V. Wang, M. Hadjicharalambous, L. Asner, J. Lee, M. Sermesant, E. Kuhl, A. Young, P. Moireau, M. Nash, D. Chapelle, and D. Nordsletten. "Multiphysics and multiscale modelling, data–model fusion and integration of organ physiology in the clinic: ventricular cardiac mechanics". In: *Interface Focus* 6.2 (2016), p. 20150083.

[13] D. Chapelle, M. A. Fernández, J. F. Gerbeau, and et al. "Numerical simulation of the electromechanical activity of the heart". In: *International Conference on Functional Imaging and Modeling of Heart* 5528 (2009), pp. 357–365.

[14] P. Colli Franzone, L. F. Pavarino, and G. Savaré. "Computational electrocardiology: mathematical and numerical modeling". In: *Complex systems in Biomedicine*. Springer, 2006, pp. 187–241.

[15] P. Colli Franzone, L. F. Pavarino, and S. Scacchi. "A Numerical Study of Scalable Cardiac Electro-Mechanical Solvers on HPC Architectures". In: *Frontiers in Physiology* 9 (2018), p. 268.

[16] E. J. Crampin, M. Halstead, P. Hunter, P. Nielsen, D. Noble, N. Smith, and M. Tawhai. "Computational physiology and the physiome project". In: *Experimental Physiology* 89.1 (2004), pp. 1–26.

[17] L. Dede', A. Gerbi, and A. Quarteroni. "Segregated Algorithms for the Numerical Simulation of Cardiac Electromechanics in the Left Human Ventricle". In: *The Mathematics of Mechanobiology: Cetraro, Italy 2018*. Cham: Springer International Publishing, 2020, pp. 81–116.

[18] M. Fedele and A. Quarteroni. "Polygonal surface processing and mesh generation tools for numerical simulations of the complete cardiac function". In: *MOX report, Politecnico di Milano* 32 (2019).

[19] C. Felippa, K. Park, and C. Farhat. "Partitioned analysis of coupled mechanical systems". In: *Computer Methods in Applied Mechanics and Engineering* 190.24 (2001), pp. 3247–3270.

[20] M. Fink, S. Niederer, E. Cherry, F. Fenton, J. Koivumäki, G. Seemann, R. Thul, H. Zhang, F. Sachse, D. Beard, E. Crampin, and N. Smith. "Cardiac cell modelling: observations from the heart of the cardiac physiome project". In: *Progress in Biophysics and Molecular Biology* 104.1 (2011), pp. 2–21.

[21] A. Gerbi. "Numerical approximation of cardiac electro-fluid-mechanical models: coupling strategies for large-scale simulation". PhD thesis. EPFL, 2018.

[22] A. Gerbi, L. Dede', and A. Quarteroni. "A monolithic algorithm for the simulation of cardiac electromechanics in the human left ventricle". In: *Mathematics in Engineering* 1 (2018), pp. 1–37.

[23] P. Gervasio and A. Quarteroni. "The INTERNODES Method for Non-conforming Discretizations of PDEs". In: *Communications on Applied Mathematics and Computation* 1 (2019), pp. 361–401.

[24] S. Godunov. "A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics". In: *Matematicheskii Sbornik* 89 (1959), pp. 271–306.

[25] B. E. Griffith and N. A. Patankar. "Immersed methods for fluid-structure interaction". In: *Annual Review of Fluid Mechanics* 52 (2020), pp. 421–448.

[26] J. M. Guccione and A. D. McCulloch. "Finite element modeling of ventricular mechanics". In: *Theory of Heart*. Springer, 1991, pp. 121–144.

[27] J. M. Guccione, A. D. McCulloch, and L. K. Waldman. "Passive material properties of intact ventricular myocardium determined from a cylindrical model". In: *Journal of Biomechanical Engineering* 113 (1991), pp. 42–55.

[28] M. Hirschvogel, M. Bassilious, L. Jagschies, and et al. "A monolithic 3D-0D coupled closed-loop model of the heart and the vascular system: Experiment-based parameter estimation for patient-specific cardiac mechanics". In: *International Journal for Numerical Methods in Biomedical Engineering* 33.8 (2017), e2842.

[29] M. C. Hsu and Y. Bazilevs. "Blood vessel tissue prestress modeling for vascular fluid-structure interaction simulation". In: *Finite Elements in Analysis and Design* 47 (2011), pp. 593–599.

[30] Z. M. G. Inc. *Zygote Solid 3D heart Generation II Development Report*. Technical Report. 2014.

[31] A. M. Katz. *Physiology of the Heart*. Lippincott Williams & Wilkins, 2010.

[32] R. C. Kerckhoffs, M. L. Neal, Q. Gu, J. B. Bassingthwaighte, J. H. Omens, and A. D. Mc-Culloch. "Coupling of a 3D finite element model of cardiac ventricular mechanics to lumped systems models of the systemic and pulmonic circulation". In: *Annals of biomedical engineering* 35.1 (2007), pp. 1–18.

[33] S. Krishnamoorthi, M. Sarkar, and W. S. Klug. "Numerical quadrature and operator splitting in finite element methods for cardiac electrophysiology". In: *International Journal for Numerical Methods in Biomedical Engineering* 29 (2003), pp. 1243–1266.

[34] L. Marx, M. A. F. Gsell, A. Rund, and et al. "Personalization of electro-mechanical models of the pressure-overloaded left ventricle: fitting of Windkessel-type afterload models". In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 378 (2020), pp. 1015–1034.

[35] D. A. Nordsletten, S. A. Niederer, M. P. Nash, and et al. "Coupling multi-physics models to cardiac mechanics". In: *Progress in Biophysics and Molecular Biology* 104 (2011), pp. 77–88.

[36] A. S. Patelli, L. Dede', T. Lassila, and et al. "Isogeometric approximation of cardiac electrophysiology models on surfaces: an accuracy study with application to the human left atrium". In: *Computer Methods in Applied Mechanics and Engineering* 317 (2017), pp. 248–273.

[37] P. Pathmanathan, M. O. Bernabeu, S. A. Niederer, and et al. "Computational modelling of cardiac electrophysiology: explanation of the variability of results from different numerical solvers". In: *International Journal for Numerical Methods in Biomedical Engineering* 28 (2012), pp. 890–903.

[38] P. Pathmanathan, G. R. Mirams, J. Southern, and et al. "The significant effect of the choice of ionic current integration method in cardiac electro-physiological simulations". In: *International Journal for Numerical Methods in Biomedical Engineering* 27 (2011), pp. 1751–1770.

[39] P. Pathmanathan, S. Chapman, D. Gavaghan, and J. Whiteley. "Cardiac electromechanics: the effect of contraction model on the mathematical problem and accuracy of the numerical scheme". In: *The Quarterly Journal of Mechanics & Applied Mathematics* 63.3 (2010), pp. 375–399.

[40] L. Pegolotti, L. Dede', and A. Quarteroni. "Isogeometric Analysis of the electrophysiology in the human heart: Numerical simulation of the bidomain equations on the atria". In: *Computer Methods in Applied Mechanics and Engineering* 343 (2019), pp. 52–73.

[41] R. Piersanti, P. C. Africa, M. Fedele, C. Vergara, L. Dede', A. F. Corno, and A. Quarteroni. "Modeling cardiac muscle fibers in ventricular and atrial electrophysiology simulations". In: *Computer Methods in Applied Mechanics and Engineering* 373 (2021), p. 113468.

[42] A. Quarteroni, R. Sacco, and F. Saleri. *Numerical Mathematics*. Springer, 2010.

[43] A. Quarteroni, T. Lassila, S. Rossi, and et al. "Integrated heart - coupling multiscale and multiphysics models for the simulation of the cardiac function". In: *Computer Methods in Applied Mechanics and Engineering* 314 (2017), pp. 345–407.

[44] A. Quarteroni, L. Dede', A. Manzoni, and C. Vergara. *Mathematical modelling of the human cardiovascular system: data, numerical approximation, clinical applications*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2019.

[45] F. Regazzoni. "Mathematical modeling and Machine Learning for the numerical simulation of cardiac electromechanics". PhD thesis. Politecnico di Milano, 2020.

[46]   F. Regazzoni, L. Dede', and A. Quarteroni. "Active contraction of cardiac cells: a model for sarcomere dynamics with cooperative interactions". In: *Biomechanics and modeling in mechanobiology* 17 (2018), pp. 1663–1686.

[47]   F. Regazzoni, L. Dedè, and A. Quarteroni. "Machine learning for fast and reliable solution of time-dependent differential equations". In: *Journal of Computational Physics* 397 (2019), p. 108852.

[48]   F. Regazzoni, L. Dedè, and A. Quarteroni. "Machine learning of multiscale active force generation models for the efficient simulation of cardiac electromechanics". In: *Computer Methods in Applied Mechanics and Engineering* 370 (2020), p. 113268.

[49]   F. Regazzoni, M. Salvador, P. C. Africa, M Fedele, L. Dede', and A. Quarteroni. "A cardiac electromechanics model coupled with a lumped parameters model for closed-loop blood circulation. Part I: model derivation". In: ().

[50]   F. Regazzoni and A. Quarteroni. "An oscillation-free fully partitioned scheme for the numerical modeling of cardiac active mechanics". In: *Computer Methods in Applied Mechanics and Engineering* (2020).

[51]   D. Romero, R. Sebastian, B. H. Bijnens, and et al. "Effects of the Purkinje system and cardiac geometry on biventricular pacing: a model study". In: *Annals of Biomedical Engineering* 38 (2010), pp. 1388–1398.

[52]   M. Salvador, L. Dedè, and A. Quarteroni. "An intergrid transfer operator using radial basis functions with application to cardiac electromechanics". In: *Computational Mechanics* 66 (2020), pp. 491–511.

[53]   W. J. Schroeder, B. Lorensen, and K. Martin. *The visualization toolkit: an object-oriented approach to 3D graphics.* Kitware, 2004.

[54]   N. Smith, D. Nickerson, E. Crampin, and P. Hunter. "Multiscale computational modelling of the heart". In: *Acta Numerica* 13 (2004), pp. 371–431.

[55]   K. H. ten Tusscher and A. V. Panfilov. "Alternans and spiral breakup in a human ventricular tissue model". In: *American Journal of Physiology. Heart and Circulatory Physiology* 291 (2006), pp. 1088–1100.

[56]   M Vázquez, R Arís, J Aguado-Sierra, G Houzeaux, A Santiago, M López, P Córdoba, M Rivero, and J. Cajas. "Alya red CCM: HPC-based cardiac computational modelling". In: *Selected topics of computational and experimental fluid mechanics.* Springer, 2015, pp. 189–207.

[57]   C. Vergara, M. Lange, S. Palamara, and et al. "A coupled 3D-1D numerical monodomain solver for cardiac electrical activation in the myocardium with detailed Purkinje network". In: *Journal of Computational Physics* 308 (2016), pp. 218–238.

[58]   T. Washio, J. Okada, A. Takahashi, K. Yoneda, Y. Kadooka, S. Sugiura, and T. Hisada. "Multiscale heart simulation with cooperative stochastic cross-bridge dynamics and cellular structures". In: *Multiscale Modeling & Simulation* 11.4 (2013), pp. 965–999.

[59]   C. Carton de Wiart, L. T. Diosady, A. Garai, N. K. Burgess, P. J. Blonigan, D. Ekelschot, and S. M. Murman. "Design of a modular monolithic implicit solver for multi-physics applications". In: *2018 AIAA Aerospace Sciences Meeting.* 2018, p. 1400.

MOX Technical Reports, last issues

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

77/2020   Parolini, N.; Ardenghi, G.;Dede', L.; Quarteroni, A.
A Mathematical Dashboard for the Analysis of Italian COVID-19 Epidemic Data

76/2020   Centofanti, F.; Lepore, A.; Menafoglio, A.; Palumbo, B.; Vantini, S.
Functional Regression Control Chart

75/2020   F. Dassi; A. Fumagalli; D. Losapio; S. Scialò; A. Scotti; G. Vacca
The mixed virtual element method for grids with curved interfaces

74/2020   Formaggia, L; Fumagalli, A.; Scotti, A.
A multi-layer reactive transport model for fractured porous media

73/2020   Bennati, L.; Vergara, C.; Domanin, M.; Trimarchi, S.; Malloggi, C.; Silani, V.; Parati, G.; Casa
A computational fluid structure interaction study for carotids with different atherosclerotic plaques

72/2020   Belli E.; Vantini S.
Measure Inducing Classification and Regression Trees for Functional Data

71/2020   Belli E; Vantini S.
Ridge regression with adaptive additive rectangles and other piecewise functional templates

70/2020   Belli E.
Smoothly Adaptively Centered Ridge Estimator

69/2020   Galvani, M.; Torti, A.; Menafoglio, A.; Vantini S.
FunCC: a new bi-clustering algorithm for functional data with misalignment

67/2020   Caramenti, L.; Menafoglio, A.; Sgobba, S.; Lanzano, G.
Multi-Source Geographically Weighted Regression for Regionalized Ground-Motion Models