



MOX-Report No. 35/2021

**Accelerating the convergence to a limit cycle in 3D
cardiac electromechanical simulations through a
data-driven 0D emulator**

Regazzoni, F.; Quarteroni, A.

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

mox-dmat@polimi.it

<http://mox.polimi.it>

Accelerating the convergence to a limit cycle in 3D cardiac electromechanical simulations through a data-driven 0D emulator

F. Regazzoni^{1,*} and A. Quarteroni^{1,2}

¹MOX - Dipartimento di Matematica, Politecnico di Milano,
P.zza Leonardo da Vinci 32, 20133 Milano, Italy

²Mathematics Institute, École Polytechnique Fédérale de Lausanne (EPFL),
Av. Piccard, CH-1015 Lausanne, Switzerland (*Professor Emeritus*)

**Corresponding author*

Abstract

The results of numerical simulations of 3D cardiac electromechanical models are typically characterized by a long transient before reaching a periodic solution known as limit cycle. Since the only clinically relevant output is the one associated with such limit cycle, a long transient translates into a serious computational overhead. To accelerate the convergence to the limit cycle, we propose a strategy based on a surrogate model, wherein the computationally demanding 3D components are replaced by a 0D emulator. This emulator is built through an automated data-driven algorithm on the basis of pressure-volume transients of as few as three heartbeats simulated through the 3D model. The 0D emulator, consisting of a time-dependent pressure-volume relationship, allows to accurately detect the location of the limit cycle in less than one minute on a standard laptop. Then, using as an initial guess for the 3D model the solution obtained with its 0D surrogate, it is possible to reach in just two heartbeats a solution that is as close to the limit cycle as the one obtained after more than 20 heartbeats with the full-order 3D model. In this manner, the proposed approach achieves an overall speedup in the simulation of about an order of magnitude.

In practical applications, an electromechanical model needs to be coupled with a model for the external circulation. The latter is typically represented by either a Windkessel-type preload-afterload model, emulating the boundary conditions, or by a closed-loop model of the entire circulatory network. The closed-loop model provides higher quality results in terms of physiological soundness; however, reaching a limit cycle is more challenging in this setting. It is in this context that our 0D emulator turns out to be particularly effective.

The 0D emulator is also recommended in many-query settings (e.g. when performing sensitivity analysis, parameter estimation and uncertainty quantification), that call for the repeated solution of the model for different values of the parameters. As a matter of fact, the emulator does not depend on the circulation model to which it is coupled, hence its construction does not have to be repeated when the parameters of the circulation model vary. Finally, should the parameters of the 3D electromechanical model vary as well, we propose a parametric emulator, obtained by interpolation of emulators constructed for given values of the parameters. In all these cases, our numerical results show that the emulator is able to provide the 3D model with an initial guess such that, after only two heartbeats, the solution is very close to the limit cycle. This paper is accompanied by a Python library implementing the proposed algorithm, open to the integration with existing cardiac solvers.

Keywords Mathematical modeling, Cardiac modeling, Cardiac electromechanics, Multiphysics models, Lumped-parameter models

1 Introduction

Computational models of the cardiac function represent a promising tool to quantitatively analyze clinical data. Through numerical simulations of cardiac electromechanics and hemodynamics, it is possible to reveal mechanistic links between the different components of this complex system, to elucidate the relationships between microscopic variations and macroscopic biomarkers, to perform *in silico* investigations on the effects of drugs or therapies [3, 6, 7, 9, 10, 13, 21, 27, 28, 30, 32]. Despite their impressive developments in recent years, a major obstacles to the clinical exploitation of computational models of cardiac electromechanics is their huge demand for computational resources. The numerical simulation of a single heartbeat for a biophysically detailed and anatomically accurate cardiac model, indeed, may require several hours (if not days) in terms of computational times on supercomputer platforms.

This huge computational cost is exacerbated by the fact that, to obtain meaningful results, simulating a single heartbeat might not even be sufficient. In fact, during the first heartbeats, the numerical model typically goes through a transient, and then it settles down to a periodic behavior. This periodic behavior represents the so-called *limit cycle* of the system of differential equations that makes up the mathematical model. In the literature, it is also referred to as *steady solution* or *periodic orbit*. A key point is that the only clinically meaningful quantities are those computed from such limit cycle, while the solution obtained during the transient has no physical meaning and it is only instrumental in reaching the periodic solution. However, unless all variables in the mathematical model – including tissue displacement, electrophysiological, subcellular and hemodynamical variables – are already along the periodic orbit at time $t = 0$, going through this transient is unavoidable. This represents a serious computational burden.

Additionally, cardiac electromechanics models need to be closed with suitable relationships linking the pressure inside the chambers with the blood flows through the valves. The two most widely used approaches in the literature involve, on the one hand, the use of preload-afterload models (e.g. Windkessel-type models) surrogating the boundary conditions [12, 16]; on the other hand, the coupling with closed-loop 0D (i.e. lumped-parameter) models of the external circulation [2, 11, 14, 24, 33]. The models belonging to the second class provide more meaningful results from a physiological perspective, because they take into account the mutual interactions between the cardiac and external circulation. However, reaching the limit cycle with closed-loop models is particularly critical. As a matter of fact, the variables of these 0D hemodynamics models must be suitably initialized as well. Since, in closed-loop model, any change in the state variables affects the vascular network downstream, thus creating circular dependencies, finding a periodic solution (i.e. a limit cycle) is particularly delicate in this case. For this reason, in this paper we devote a special care to 3D electromechanical models coupled with closed-loop circulation models.

Several expedients have been devised to find an initial guess for some of the variables of cardiac electromechanical models, in such a way they are (hopefully) close to the limit cycle. For instance, the subcellular variables of the ionic and activation models are typically initialized by running a large number of heartbeats in a single-cell simulation. The final state of this 0D model is then employed as initial state in each node of the 3D computational mesh. To initialize the tissue displacement, instead, the cardiac chambers are usually inflated by applying at the endocardium the end-diastolic pressure. The displacement obtained by solving this stationary elastic problem is then employed as initial state for the electromechanical simulation. However, the endocardial pressure needed to solve

the stationary problem is not typically known in advance, especially when the 3D electromechanical model is coupled with a 0D circulation model.

In recent years, a number of approaches have been proposed to reduce the computational cost associated with simulating the heartbeats required by electromechanical models to reach a limit cycle, especially when coupled with circulations models. Since high accuracy is not required during this transient phase, the simulation of these heartbeats can be conducted on a coarser computational mesh. Alternatively, in [2], 18 heartbeats are carried out without letting the Newton solver reach convergence for the mechanical subproblem (i.e. a single Newton iteration is performed for each time step), and then two heartbeats are simulated with a fully converging Newton solver. In [11], a purely mechanical simulation is done to find an initial condition for the circulation model that is close to the periodic orbit. However, reaching the limit cycle remains challenging, due to the large number of cycles that are typically required for convergence. As a matter of fact, depending on how far the initial guess is from the limit cycle, many heartbeats could be necessary to get close to a limit cycle. In practice, order of 20 beats are typically performed [2]. However, in [18] it is reported that order of 100 beats could be necessary in certain circumstances. This is particularly critical in many-query settings such as sensitivity analysis, parameter estimation and uncertainty quantification. In these contexts, which already in themselves require to simulate a high number of heartbeats, very often one renounces to achieve the limit cycle and simulates a single heartbeat, at the risk of obtaining a physiologically inaccurate solution. In [17] and [16], for instance, a single heartbeat is simulated for each choice of the parameters, even if the authors recognize that the validity of the results could be compromised by this choice.

The above reasons call for the development of techniques to achieve the limit cycle at a low computational cost, possibly without undergoing a full transient simulation with the 3D electromechanical model. In this paper we propose a strategy aimed at this purpose, based on a 0D emulator built on the basis of a few beats obtained with the 3D model. Once the model has been built, it is used - at a very low computational cost - to obtain a limit cycle by means of a 0D simulation, which provides an initial guess for the 3D model. This approach can be interpreted as a 3D-0D-3D V-cycle, reminiscent of the multigrid method [34]. Having replaced the 3D multiscale model with a 0D surrogate during the simulation of a number of heartbeats, the computational cost is drastically reduced. As a matter of fact, our approach allows to simulate with the full-order 3D model only a small fraction of the heartbeats required to reach the limit cycle, surrogating the large part of the heartbeats by means of the 0D emulator, at a negligible computational cost.

We propose an algorithm for the automated construction of such 0D emulator, starting from a pressure-volume transient generated with the full-order electromechanical model. A Python implementation is made publicly available in a GitHub repository¹ accompanying this paper, thus enabling for the integration with existing software.

This paper is structured as follows. In Sec. 2 we present the methods proposed in this paper. Then, in Sec. 3 we present a complete electromechanical model, including the external circulation. In Sec. 4 we show the results of several test cases, carried out with the above mentioned models, to demonstrate the effectiveness of the proposed methods. Finally, we draw our conclusions and final remarks in Sec. 5.

2 The 0D emulator

A mathematical model of cardiac electromechanics consists of a set of differential equations, describing the evolution of a set of variables within a 3D computational domain. These variables are

¹<https://github.com/FrancescoRegazzoni/cardioemulator>

associated with different physical processes pertaining to the cardiac function, including electrophysiology, cellular contraction and tissue mechanics. In what follows, we will employ the symbol \mathcal{M}_{3D} to refer to the 3D electromechanical model. The \mathcal{M}_{3D} model may include a single or multiple cardiac chambers. For simplicity, in this paper we will mainly focus on the single-chamber case, but generalization to the multiple-chamber case will be provided as well. In all the cases, the \mathcal{M}_{3D} must be supplemented with suitable closure relationships to link the fluxes of blood flowing through the cardiac valves with the blood pressure inside the cardiac chamber (or chambers). We will use the symbol \mathcal{C} to denote these closure relationships, whether they are in the form of afterload-preload models, or closed-loop circulation models. Coupling the electromechanical model \mathcal{M}_{3D} with the circulation model \mathcal{C} , we then obtain the model $\mathcal{M}_{3D}\text{-}\mathcal{C}$.

2.1 Building the 0D emulator

In this paper we propose a 0D emulator of the \mathcal{M}_{3D} model – that we will denote as \mathcal{M}_{0D} – derived from a few heartbeats simulated with the 3D model itself. Our 0D emulator is based on the assumption that, at each time t , the pressure p and volume V of a given cardiac chamber are linked through a pressure-volume (PV) relationship in the form of $p = \Psi(V, t)$. The main assumption under the 0D emulator is that the pressure-volume relationship can be factorized as follows

$$\Psi(V, t) = (1 - \varphi_{\text{act}}(t)) \Psi_{\text{ED}}(V) + \varphi_{\text{act}}(t) \Psi_{\text{ES}}(V), \quad (1)$$

where Ψ_{ES} represents the end-systolic pressure-volume relationship (ESPVR) and Ψ_{ED} represents the end-diastolic pressure-volume relationship (EDPVR). The function φ_{act} is a time-dependent function encoding the activation kinetics. Ideally, $\varphi_{\text{act}} = 0$ at the end of diastole (the tissue is fully relaxed), while $\varphi_{\text{act}} = 1$ at the end of systole (the tissue is fully contracted). Clearly, φ_{act} is periodic in t with period T_{HB} , that is the duration of a heartbeat. Note that inertia and damping effects are neglected within the 0D emulator.

We propose an algorithm to construct, in a reproducible manner, the functions defining the emulator (i.e. Ψ_{ED} , Ψ_{ES} and φ_{act}) from the output of a simulation obtained with the $\mathcal{M}_{3D}\text{-}\mathcal{C}$ model. The results of this simulation are referred to as *sample PV loops*. We remark that the latter clearly do not need to be close to a limit cycle. On the contrary, the construction of the emulator is facilitated when the sample simulation starts far from the limit cycle, as the transient would cover a wider region of the pressure-volume plane. The algorithm to construct the 0D emulator is described in detail in App. A. It envisages three steps, visually summarized in Fig. 1 and briefly described as follows.

- (a) First, we construct the EDPVR function $\Psi_{\text{ED}}(V)$, following one of two alternative approaches (*simulated EDPVR* or *fitted EDPVR*). The simulated EDPVR approach consists in solving a sequence of elastostatic problems through the \mathcal{M}_{3D} model, for increasing values of pressure, and then interpolating the resulting pressure-volume pairs. Conversely, the fitted EDPVR is based on best-fit of the Klotz curve [15] on the end-diastolic segments of the sample PV loops.
- (b) Then, we obtain the ESPVR function $\Psi_{\text{ES}}(V)$ by fitting the points corresponding to the systolic peaks with a straight line.
- (c) Finally, we construct the activation kinetics function $\varphi_{\text{act}}(t)$, inverting the relationship Eq. (1) with respect to $\varphi_{\text{act}}(t)$ on the data corresponding to the last cycle, and periodically extending it outside the interval $[0, T_{\text{HB}})$.

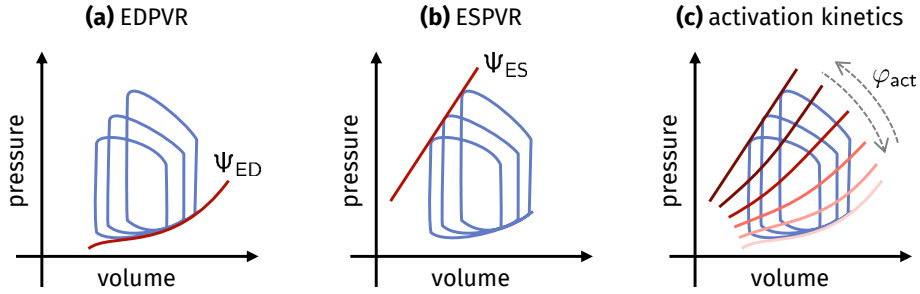


Figure 1: The algorithm to construct the 0D emulator comprises three steps: (a) construction of the EDPVR (end-diastolic pressure-volume relationship) function $\Psi_{ED}(V)$; (b) construction of the ESPVR (end-systolic pressure-volume relationship) function $\Psi_{ES}(V)$; (c) construction of the activation kinetics function $\varphi_{act}(t)$, accounting for the activation and relaxation times. The blue lines represent the sample PV loops used to construct the emulator.

2.2 3D-0D-3D V-cycle: reaching the limit cycle through the 0D emulator

The \mathcal{M}_{0D} model surrogates the pressure-volume relationship of the \mathcal{M}_{3D} model. Hence, similarly to the \mathcal{M}_{3D} model, also the \mathcal{M}_{0D} model can be coupled with the closure relationships \mathcal{C} , whether they are in the form of preload-afterload laws or in the form of a closed-loop circulation model. The resulting coupled model, that we denote by $\mathcal{M}_{0D-\mathcal{C}}$, thus represents a surrogate of the $\mathcal{M}_{3D-\mathcal{C}}$ model. We remark that, in case the \mathcal{M}_{3D} model includes more than one cardiac chamber, an independent 0D emulator should be built for each chamber, according to the algorithm of Sec. 2.1.

In fact, the $\mathcal{M}_{0D-\mathcal{C}}$ model can be exploited to accelerate the search for a limit cycle of the $\mathcal{M}_{3D-\mathcal{C}}$ model itself. Our proposed procedure, named *3D-0D-3D V-cycle*, is made of three steps, summarized in Fig. 2.

- (S1) First, a few sample PV loops are simulated with the $\mathcal{M}_{3D-\mathcal{C}}$ model.
- (S2) Following Sec. 2.1, a 0D emulator \mathcal{M}_{0D} is built for each cardiac chamber included in the model, based on the sample PV loops. Then, a large number of heartbeats (say n_{HB}^{0D}) are simulated by solving the $\mathcal{M}_{0D-\mathcal{C}}$ model, at a negligible computational cost, until a limit cycle is reached.
- (S3) A second simulation is set up for the $\mathcal{M}_{3D-\mathcal{C}}$ model, based on the limit cycle obtained at the previous step. Specifically, in case the \mathcal{C} has some internal variables, these are set equal to their final state obtained in step (S2). Moreover, also the \mathcal{M}_{3D} model is initialized according to one of the following alternative strategies:
 - **pressure control:** the tissue displacement is initialized by applying at the endocardial surfaces the final pressures obtained with the $\mathcal{M}_{0D-\mathcal{C}}$ model simulation of step (S2).
 - **volume control:** the tissue displacement is initialized by finding the equilibrium configuration under the constraint that the volume of the chambers equal the final volumes obtained in the $\mathcal{M}_{0D-\mathcal{C}}$ model simulation of step (S2). In this case, the endocardial pressures are determined as Lagrange multipliers.

Both in (S1) and (S3), the state variables associated with the electrophysiology and active force generation models can be initialized by means of standard strategies, according to the models at hand. In Sec. 3.2 we will provide an example, related to the models we used to produce the results presented in this paper.

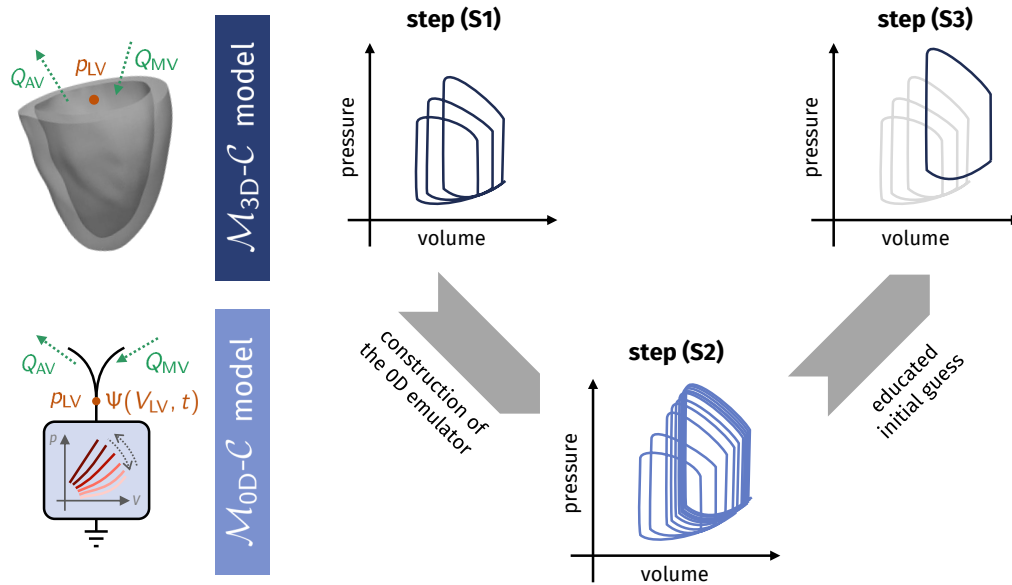


Figure 2: Graphical display of the 3D-0D-3D V-cycle, in the case of a left ventricle (LV) electromechanical model. On the left, the high-fidelity 3D electromechanical model is compared with its 0D surrogate. At the terminals of both models we have the chamber pressure (p_{LV}) and the inflow and outflow blood flux (in the case of LV, the flows through the mitral and aortic valves, respectively denoted by Q_{MV} and Q_{AV}).

2.3 Using the emulator in many-query settings

We remark that the \mathcal{M}_{0D} emulator surrogates the \mathcal{M}_{3D} model, independently of the \mathcal{C} model it is coupled with. This implies that, in case the parameters of the \mathcal{C} model are changed, (S1) does not need to be repeated. Therefore, the \mathcal{M}_{0D} - \mathcal{C} model can be employed as a surrogate of \mathcal{M}_{3D} - \mathcal{C} in many-query contexts that involve the repeated solution of the \mathcal{M}_{3D} - \mathcal{C} model for different choices of the circulation parameters. It can be used, e.g. to accelerate sensitivity analysis or the calibration the parameters of the circulation model.

In realistic scenarios, however, besides the hemodynamical parameters, also the ones of the electromechanical model \mathcal{M}_{3D} may require calibration. In case the parameters of the \mathcal{C} and of the \mathcal{M}_{3D} models need to be simultaneously calibrated, we propose to build a parametric emulator of the \mathcal{M}_{3D} model, by interpolating between emulators obtained for fixed values of the parameters of the \mathcal{M}_{3D} model.

A typical case is when the active contractility (henceforth denoted by α) is to be determined. By the Young-Laplace law [19], the pressure within a cavity is proportional to the wall tension, which is given by the sum of a passive and an active part. Since the active part is in turn proportional to the contractility, we expect an affine dependence of the PV relationship on α . That is, we expect that the PV relationship for a generic contractility α is well approximated by

$$\Psi(V, t; \alpha) = \frac{\alpha - \alpha_A}{\alpha_B - \alpha_A} \Psi_B(V, t) + \frac{\alpha - \alpha_B}{\alpha_A - \alpha_B} \Psi_A(V, t), \quad (2)$$

where Ψ_A and Ψ_B represent the PV relationships for $\alpha = \alpha_A$ and $\alpha = \alpha_B$, respectively. Hence, let us suppose to generate two sample simulations for two contractility values, namely α_A and α_B . Based on each sample simulation, we construct a different emulator, according to the algorithm of Sec. 2.1. Denoting the two emulators by Ψ_A and Ψ_B , Eq. (2) defines a parametric emulator where the contractility α can assume arbitrary values.

In case one needs to parameterize an emulator with respect to parameters whose dependency lacks a physical intuition analogous to the case discussed above, one can still phenomenologically apply formula (2). Alternatively, an interpolation formula with more than two points can be considered.

For illustrative purposes, let us consider a use case of the 0D emulator for parameter calibration purposes in a LV electromechanical model. In a first stage (step (S1)), a simulation is performed with the \mathcal{M}_{3D} - \mathcal{C} model, for some choice of parameters of the \mathcal{C} model. In case the \mathcal{M}_{3D} model is also to be calibrated, then an additional simulation is performed, with a different value of α (to enable a parametric emulator). Based on these sample PV loops, an emulator (or a parametric emulator) is built. Now we pass to step (S2), in which the parameters of the model are varied (either manually or through some automated procedure) until the distance between simulated and measured pressures and volumes are minimized. This step is entirely performed resorting to the \mathcal{M}_{0D} - \mathcal{C} only. Once the calibration has been successfully completed, we move on to step (S3), in which we run a simulation with the \mathcal{M}_{3D} - \mathcal{C} model, directly with the calibrated parameters and with the initial state obtained in step (S2). In this way, the computational cost of the entire calibration procedure is drastically reduced, as the calibration procedure is performed in fully 0D manner.

3 A cardiac electromechanical model including external circulation

In this section we show how the methods presented in Sec. 2 translate to a concrete use case. Specifically, in Sec. 3.1 we consider the case where the closure relations \mathcal{C} are represented by a

closed-loop circulation model. Then, in Section 3.2, we present the \mathcal{M}_{3D} model used to produce the numerical results presented in this paper.

3.1 Coupling the 0D emulator with closed-loop circulation models

Let us denote by $\mathbf{y}(t)$ a vector collecting the state variables of the \mathcal{M}_{3D} model. The state \mathbf{y} may contain variables describing the electrophysiological activity, the subcellular dynamics and the tissue displacement. For the sake of generality, we introduce a (nonlinear) differential operator \mathcal{L} , that collectively encodes the mathematical models describing the different physical processes. These are typically written in the form of Partial Differential Equations (PDEs), with suitable boundary and initial conditions.

We consider then a lumped-parameter circulation model (denoted by \mathcal{C}), whose state $\mathbf{c}(t) \in \mathbb{R}^N$ is composed of N variables, describing pressures, volumes and blood fluxed in several compartments of the cardiovascular network. Notice that the latter may possibly include also cardiac chambers that are not represented in the \mathcal{M}_{3D} model. Examples are given in [1, 5, 14, 24].

We first consider the case in which the model \mathcal{M}_{3D} only includes the left ventricle (LV). In this case, the coupled electromechanics-circulation model $\mathcal{M}_{3D}\text{-}\mathcal{C}$ reads as follows

$$\left\{ \begin{array}{l} \frac{\partial \mathbf{y}(t)}{\partial t} = \mathcal{L}(\mathbf{y}(t), p_{LV}(t), t) \quad \text{for } t \in (0, T], \\ \frac{d\mathbf{c}(t)}{dt} = \mathbf{f}(\mathbf{c}(t), p_{LV}(t), t) \quad \text{for } t \in (0, T], \\ V_{LV}^{\mathcal{M}}(\mathbf{y}(t)) = V_{LV}^{\mathcal{C}}(\mathbf{c}(t)) \quad \text{for } t \in (0, T], \\ \mathbf{y}(0) = \mathbf{y}_0, \\ \mathbf{c}(0) = \mathbf{c}_0. \end{array} \right. \quad (3)$$

The electromechanical and the circulation models are coupled through the volumetric consistency constraint $V_{LV}^{\mathcal{C}}(\mathbf{c}) = V_{LV}^{\mathcal{M}}(\mathbf{y})$, where the left-hand and right-hand sides represent the LV volume predicted, respectively, by the 0D circulation model (\mathcal{C}) and by the 3D electromechanical model (\mathcal{M}_{3D}). This coupling is enforced via a Lagrange multiplier, namely p_{LV} , which represents the blood pressure inside the LV. Note that the variable p_{LV} appears at right-hand side of both the \mathcal{C} model and the \mathcal{M}_{3D} model (in the latter, e.g., it plays a role in the boundary conditions at the endocardium). The $\mathcal{M}_{3D}\text{-}\mathcal{C}$ model is represented in Fig. 3 (top).

In this setting, the 0D emulator \mathcal{M}_{0D} can be coupled with the 0D circulation model \mathcal{C} , leading to the following $\mathcal{M}_{0D}\text{-}\mathcal{C}$ model

$$\left\{ \begin{array}{l} \frac{d\mathbf{c}(t)}{dt} = \mathbf{f}(\mathbf{c}(t), p_{LV}(t), t) \quad \text{for } t \in (0, T], \\ p_{LV}(t) = \Psi(V_{LV}^{\mathcal{C}}(\mathbf{c}(t)), t) \quad \text{for } t \in (0, T], \\ \mathbf{c}(0) = \mathbf{c}_0. \end{array} \right. \quad (4)$$

This $\mathcal{M}_{0D}\text{-}\mathcal{C}$ model represents a fully 0D surrogate of the $\mathcal{M}_{3D}\text{-}\mathcal{C}$ model (see Fig. 3). In the former, the LV pressure p_{LV} is not a Lagrange multiplier enforcing the volumetric consistency – as in the latter – rather, it can be directly obtained from the 0D emulator. We remark that in passing from $\mathcal{M}_{3D}\text{-}\mathcal{C}$ to $\mathcal{M}_{0D}\text{-}\mathcal{C}$ we only replaced the computationally intensive part (that is the 3D electromechanical model \mathcal{M}_{3D}) by the 0D surrogate \mathcal{M}_{0D} , while retaining the computationally lightweight part (that is the 0D circulation model \mathcal{C}) in its high-fidelity form. We will show that, in this way, the $\mathcal{M}_{0D}\text{-}\mathcal{C}$ model achieves a very favorable trade-off between accuracy and computational cost.

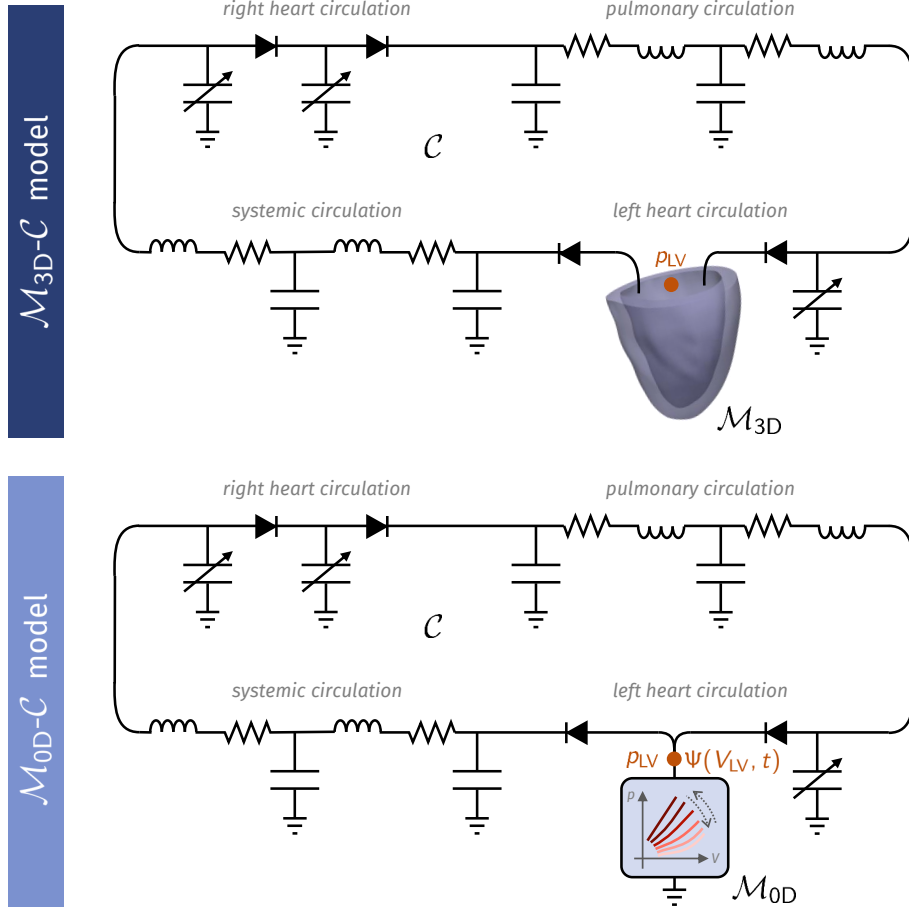


Figure 3: Comparison of the $\mathcal{M}_{3D}\text{-}\mathcal{C}$ (top) and the $\mathcal{M}_{0D}\text{-}\mathcal{C}$ (bottom) models. In the former, a 3D electromechanical model (denoted by \mathcal{M}_{3D}) of one or multiple cardiac chambers is coupled with a lumped-parameter model of blood circulation (denoted by \mathcal{C}). In the latter, instead, the \mathcal{M}_{3D} is replaced by a 0D emulator, denoted by \mathcal{M}_{0D} . The figure considers the case when only the LV is included in the \mathcal{M}_{3D} model.

In case the \mathcal{M}_{3D} model includes other chambers besides the LV, an independent 0D emulator should be built for each chamber. Then, considering for instance the biventricular case, the $\mathcal{M}_{0D-\mathcal{C}}$ model reads

$$\begin{cases} \frac{d\mathbf{c}(t)}{dt} = \mathbf{f}(\mathbf{c}(t), p_{LV}(t), p_{RV}(t), t) & \text{for } t \in (0, T], \\ p_{LV}(t) = \Psi_{LV}(V_{LV}^{\mathcal{C}}(\mathbf{c}(t)), t) & \text{for } t \in (0, T], \\ p_{RV}(t) = \Psi_{RV}(V_{RV}^{\mathcal{C}}(\mathbf{c}(t)), t) & \text{for } t \in (0, T], \\ \mathbf{c}(0) = \mathbf{c}_0, \end{cases} \quad (5)$$

where Ψ_{LV} and Ψ_{RV} denote the 0D emulator of the left and right ventricles, respectively.

3.2 The electromechanical model

Before presenting the results obtained with the methods proposed in this paper, we briefly introduce the electromechanics (\mathcal{M}_{3D}) and closed-loop circulation (\mathcal{C}) models employed to produce these results. Nonetheless, we emphasize that the above methods can be applied to different models too.

We model cardiac electrophysiology through the Monodomain equation [8], coupled with the *ten Tusscher-Panfilov* model [29]. Active force generation is described through the biophysically detailed RDQ20-MF model [22]. To model the passive behavior of the cardiac tissue, we employ the quasi-incompressible exponential *Usyk* constitutive law [31]. At the pericardium we impose spring-damper boundary conditions in both normal and tangent direction, while at the ventricular base we impose energy-consistent boundary conditions [23]. The geometry is taken from the Zygote Solid 3D heart model [35], for which the stress-free reference configuration is recovered by the algorithm proposed in [24]. The distribution of cardiac fibers is generated by means of the rule based Bayer-Blake-Plank-Trayanova algorithm [4, 20]. For blood circulation (i.e. \mathcal{C}) we employ the closed-loop lumped-parameter model of [24], that describes the systemic and pulmonary systemic and venous networks by means of RLC (resistance, inductance, capacitance) circuits, cardiac chambers as time-varying elastance elements and cardiac valves as diodes. Concerning the numerical approximation, we use bilinear Finite Elements on a tetrahedral mesh for the space discretization, while for the time discretization we employ the staggered scheme presented in [25]. Additionally, we adopt the stabilized-staggered scheme of [26] to couple the active stress model with the passive mechanics one. The simulation are obtained with the high-performance C++ library `lifex` (<https://lifex.gitlab.io/lifex>), developed within the iHEART project².

In (S1) and (S3), for the initialization of the state variables we proceed as follows.

- First, we run a single-cell simulation with the ionic model for 1000 heartbeats, to reach a limit cycle. The ionic variables and the transmembrane potential obtained at the end of this 0D simulation are then employed to initialize the corresponding variables in each node of the computational mesh.
- Then, we perform a single-cell simulation with the force generation model, considering a constant calcium input equal to the final calcium concentration obtained in the 0D ionic simulation (end-diastolic concentration) and a reference sarcomere length $SL = 2.2 \mu\text{m}$. The simulation is stopped when a steady-state is reached.
- Finally, we initialize the tissue displacement, with the volume control approach, as described in Sec. 2.2. In this step, we set a uniform active stress, corresponding to the end-diastolic steady-state obtained at the previous step.

²iHEART - An Integrated Heart Model for the simulation of the cardiac function, European Research Council (ERC) grant agreement No 740132, P.I. Prof. A. Quarteroni

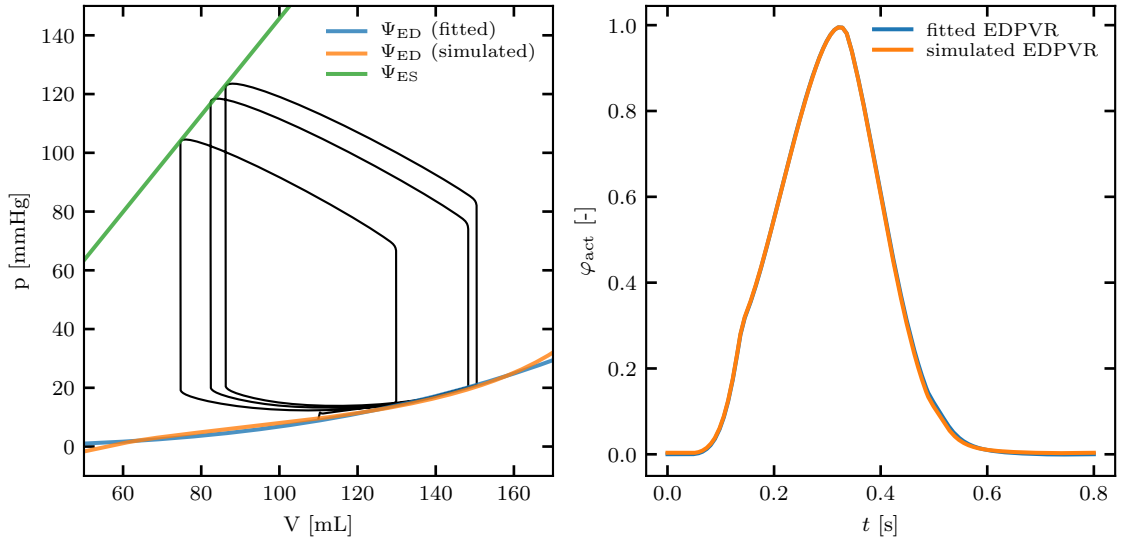


Figure 4: Representation of the 0D emulators obtained with the *fitted EDPVR* and with the *simulated EDPVR* approaches. Left: end-diastolic and end-systolic PV relationships. The black curve represents the sample PV loops used to build the emulator. Right: activation kinetics function $\varphi_{\text{act}}(t)$, obtained with the two approaches.

4 Results

This section is devoted to present some numerical results obtained by means of the methods proposed in this paper.

4.1 Construction of the 0D emulator

By employing the \mathcal{M}_{3D-C} model described in Sec. 3.2, we run a simulation of $N_{PV} = 3$ heartbeats, thus obtaining the PV loops displayed in Fig. 4 (left, black line). Based on these sample PV loops, we build two emulators, employing the *fitted EDPVR* and the *simulated EDPVR* approaches, as described in App. A.1. We show in Fig. 4 the functions $\Psi_{ED}(V)$, $\Psi_{ES}(V)$ and $\varphi_{\text{act}}(t)$ obtained with the two different approaches. We remark that the two approaches provide the same function $\Psi_{ES}(V)$, while the function $\varphi_{\text{act}}(t)$, being defined based on $\Psi_{ED}(V)$, is slightly different in the two cases.

The triplet $(\Psi_{ED}, \Psi_{ES}, \varphi_{\text{act}})$ unambiguously identifies the emulator. However, a more effective representation is provided by Fig. 5, where we show how the emulated PV relationship evolves in the different phases of the heartbeat. The figure is divided into two parts. In the upper-right region, the PV curves corresponding to systole (i.e. the times t such that $\varphi'_{\text{act}}(t) > 0$) are shown. The different shades of red represent different instants of the heartbeat. Similarly, the lower left region shows - with different shades of blue - the curves associated with diastole (when $\varphi'_{\text{act}}(t) < 0$). These curves are denser in the lower part, where φ_{act} varies more slowly. An alternative representation is shown in Fig. 6, where Ψ is displayed in the (t, V, p) plane. The same graph also shows the three sample PV loops used to build the emulator.

The construction of the 0D emulator is based on the assumption that the time-dependent PV

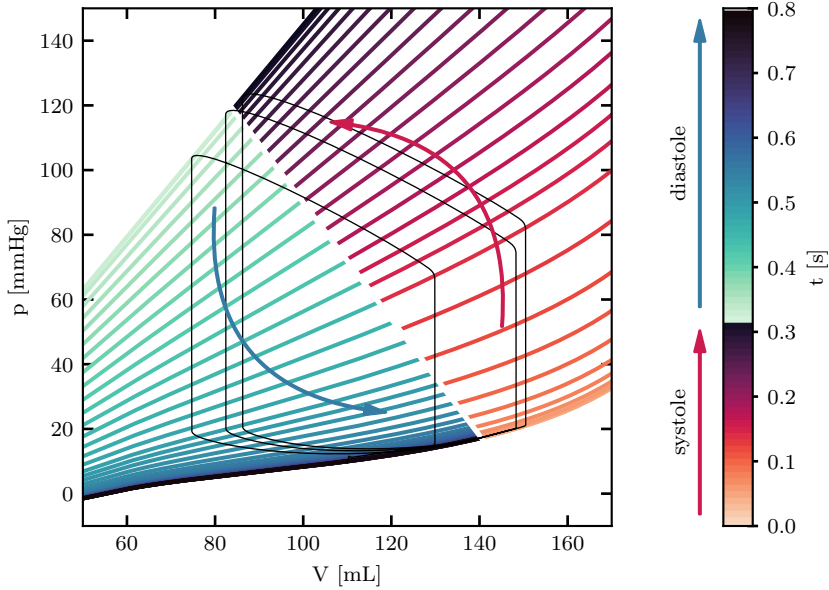


Figure 5: Evolution of the PV relationships in the different phases of the heartbeat. See text for more details. The figure refers to the emulator obtained with the simulated EDPVR approach.

relationship can be factorized as in Eq. (1). If this assumption was exact, than superimposing the function

$$\tilde{\varphi}_{\text{act}}(t) = \frac{\Psi(V(t), t) - \Psi_{\text{ED}}(V(t))}{\Psi_{\text{ES}}(V(t)) - \Psi_{\text{ED}}(V(t))}. \quad (6)$$

for different heartbeats, all the curves would coincide. That is to say, the function $\tilde{\varphi}_{\text{act}}$ would be periodic with period T_{HB} . In Fig. 7, we show the function $\tilde{\varphi}_{\text{act}}(t)$ obtained, with the two emulators, for the three sample heartbeats (used for the construction of the emulator themselves) and two additional heartbeats, for a total of five heartbeats. As shown in the figure, the overlap along the different beats is very good. This supports the validity of assumption (1), and thus that of the emulator itself in surrogating the behavior of the high-fidelity electromechanical model.

To test the ability of the $\mathcal{M}_{0\text{D}}$ emulator to faithfully replicate the behavior of the $\mathcal{M}_{3\text{D}}$ model when coupled with the \mathcal{C} model, we solve the $\mathcal{M}_{0\text{D}}\text{-}\mathcal{C}$ model starting from the same initial conditions used for the $\mathcal{M}_{3\text{D}}\text{-}\mathcal{C}$ model to produce the sample PV loops. In Fig. 8, we compare the PV loops obtained with the $\mathcal{M}_{3\text{D}}\text{-}\mathcal{C}$ model and with its fully 0D surrogate, the $\mathcal{M}_{0\text{D}}\text{-}\mathcal{C}$ model. In addition to the three heartbeats used in the construction of the emulators, we consider two further beats, in order to test the ability of the emulators to be predictive beyond the time horizon of the data that forms their basis. The figures show how the curves obtained with the emulator trace with excellent fidelity those of reference. Remarkably, the accuracy increases for the last beats, when the model is approaching the limit cycle. This is a promising remark towards the use of the emulators in 3D-0D-3D V-cycles as described in Sec. 2.2.

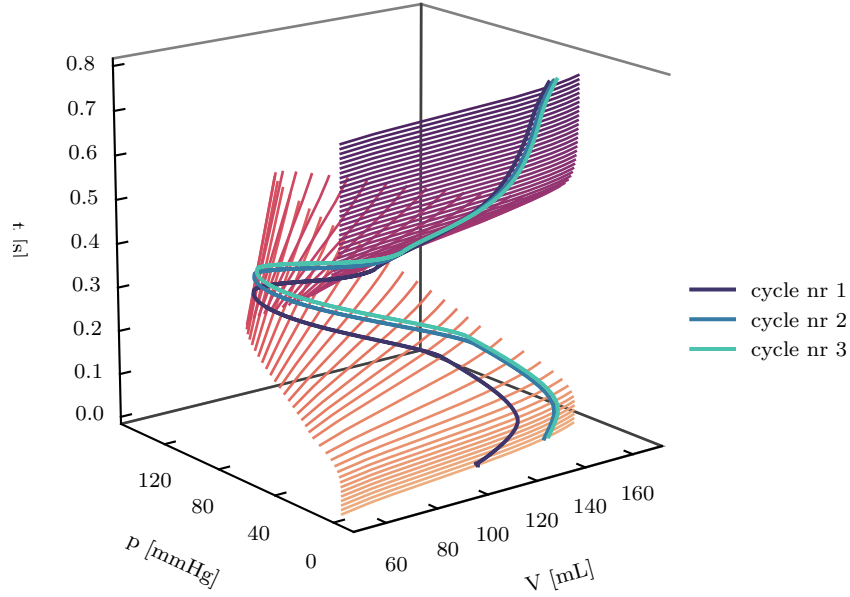


Figure 6: Function Ψ represented in the (t, V, p) plane. The figure refers to the emulator obtained with the simulated EDPVR approach.

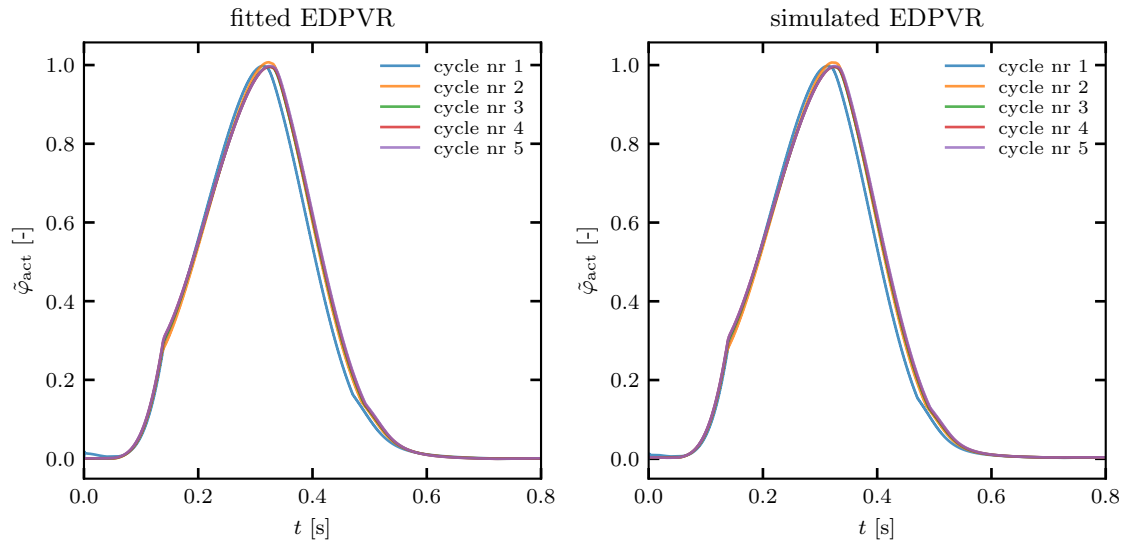


Figure 7: Function $\tilde{\varphi}_{\text{act}}$, as defined in (6), obtained for 5 consecutive heartbeats with the emulators built with the *fitted EDPVR* (left) and the *simulated EDPVR* approaches (right).

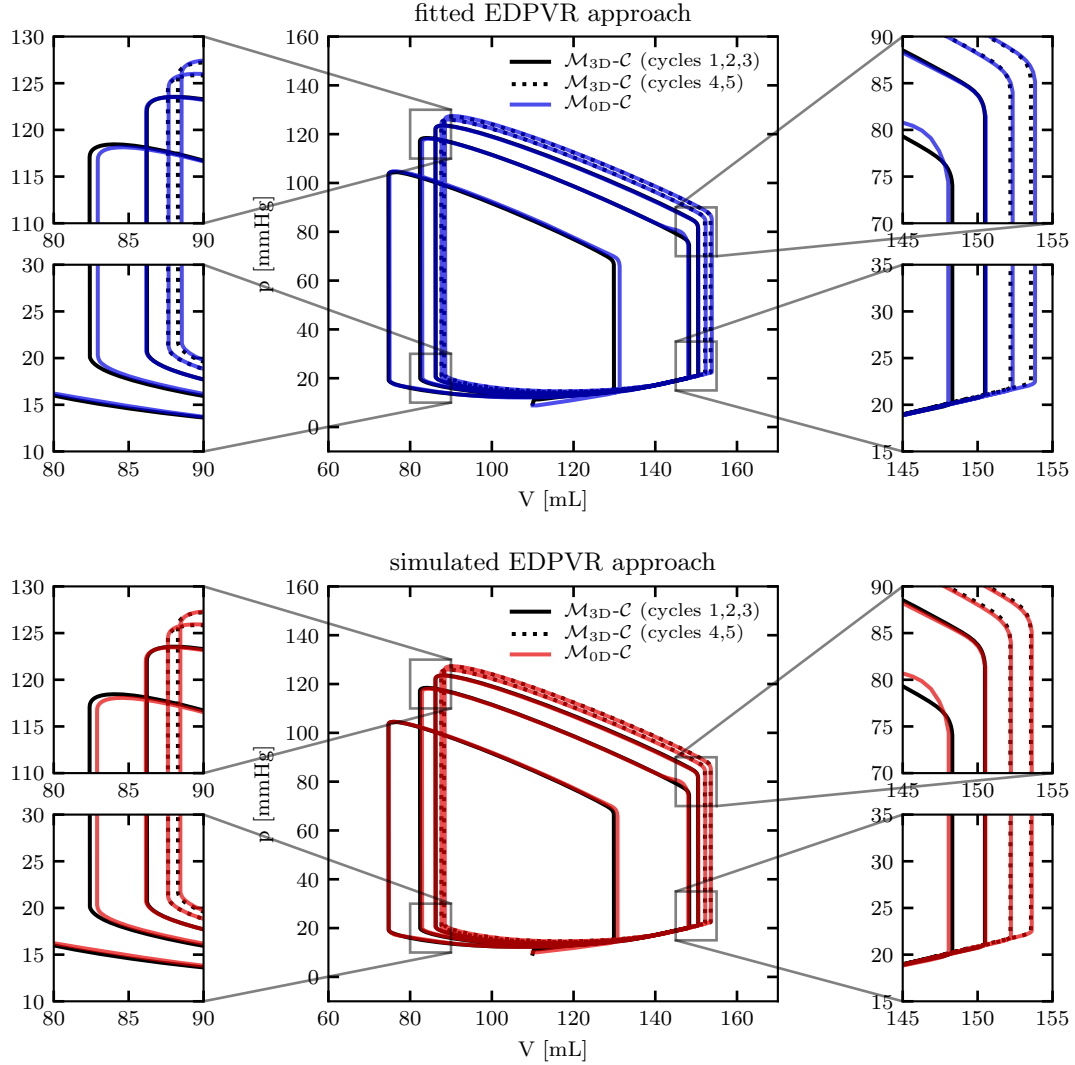


Figure 8: Comparison of the PV loops obtained with the \mathcal{M}_{3D-C} model and with the \mathcal{M}_{0D-C} surrogates. The two rows refer to the emulator built with the *fitted EDPVR* (top) and the *simulated EDPVR* approaches (bottom). The black lines are obtained with the \mathcal{M}_{3D-C} model (solid lines for the PV loops used to build the emulators, dashed lines for the two additional loops), while the coloured lines are obtained with the \mathcal{M}_{0D-C} model. The enlargements show in greater detail the instants of valve opening and closing, in which the transitions between isovolumetric and filling/ejecting phases take place.

	EDPV	Same parameters as sample PV loops	
		<i>Circulation model</i>	<i>Contractility</i>
Test 1a	fitted	✓	✓
Test 1b	simulated	✓	✓
Test 2a	fitted	✗	✓
Test 2b	simulated	✗	✓
Test 3a	fitted	✓	✗
Test 4a	fitted	✗	✗

Table 1: List of the test cases considered in this paper. The third column indicates whether the test case is performed with the same \mathcal{C} model parameters used to produce the sample PV loops; similarly, the fourth column indicates whether the same \mathcal{M}_{3D} model parameters (specifically, contractility) are used in the sample PV loops and in the test case.

4.2 3D-0D-3D V-cycles

In this section we consider the use of the emulators constructed in Sec. 4.1 to speed up the convergence to limit cycles in cardiac electromechanics simulations, as described in Sec. 2.2. We consider several test cases, in which we employ emulators alternatively built using the fitted EDPVR and simulated EDPVR approaches. The test cases are summarized in Tab. 1.

First, we consider the case in which we look for the limit cycle associated with the same parameter setting used to produce the sample PV loops (Tests 1a, 1b). The results are shown in Figs. 9 and 10, respectively. In the figures, the $n_{HB}^{0D} = 100$ heartbeats performed in a fully 0D manner (that is, through the $\mathcal{M}_{0D}\text{-}\mathcal{C}$ model) are shown in black. The last cycle, corresponding to the limit cycle of the $\mathcal{M}_{0D}\text{-}\mathcal{C}$ model, is highlighted in orange. The $\mathcal{M}_{3D}\text{-}\mathcal{C}$ model is then initialized according to the above results, and a five-heartbeat simulation is performed. The obtained PV loops are reported in green. The figures show that the pressure and volume values obtained in this way follow very closely those predicted by the emulators. As a matter of fact, the error is always below 2 mmHg for the pressure and 2 mL for the volume. More importantly, the $\mathcal{M}_{3D}\text{-}\mathcal{C}$ model initialized in this way is very close to a periodic regime: indeed, no significant drift in the PV loops obtained with the $\mathcal{M}_{3D}\text{-}\mathcal{C}$ model is visible. In Sec. 4.3 we will provide a quantitative analysis of this aspect. We observe that the emulator built by the simulated EDPVR approach (Test 1b) better predicts the end-diastolic part of the PV loops than that built by the fitted EDPVR approach (Test 1a). This is not surprising, as, while the former is built exploiting physical knowledge, the latter is only based on an extrapolation of the sample PV loops.

We then evaluate the ability of the \mathcal{M}_{0D} emulator to effectively surrogate the \mathcal{M}_{3D} model still in settings other than the one used to generate the PV loops samples. To this end, we increase the active elastances of the atria, increase the systemic resistances, and decrease the total amount of circulating blood. These changes are designed so that the regime PV loops significantly differ from the ones obtained in Tests 1a and 1b. By using this setting, we perform steps (S2) and (S3) of the 3D-0D-3D V-cycle, with both emulators (fitted and simulated EDPVR), leading to Tests 2a and 2b, respectively. The results, shown in Figs. 11 and 12, reveal that the \mathcal{M}_{0D} emulator can effectively detect the position of the limit cycle associated with a setting even different from the one used to build the emulator itself.

Finally, we consider the case where even the parameters of the electromechanical model are varied between sample PV loop generation and limit cycle search. Specifically, we consider a parametric emulator, as described in Sec. 2.3, build with respect to the contractility α . In the electromechanical

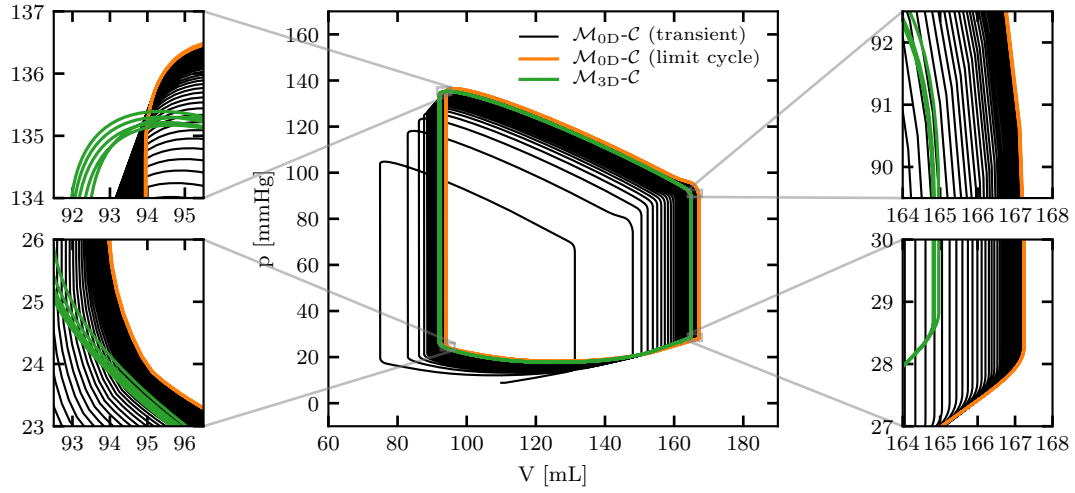


Figure 9: Test 1a: results of the 3D-0D-3D V-cycle (fitted EDPVR approach).

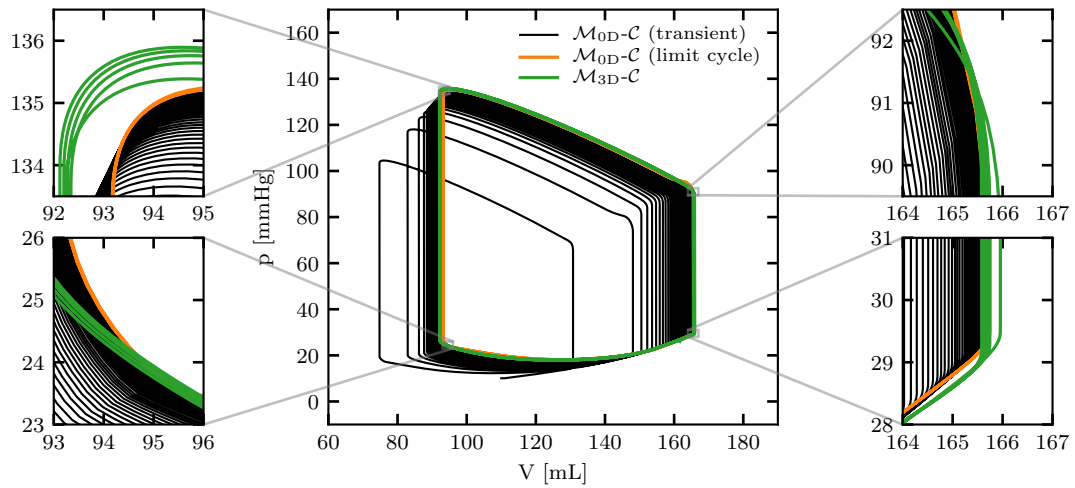


Figure 10: Test 1b: results of the 3D-0D-3D V-cycle (simulated EDPVR approach).

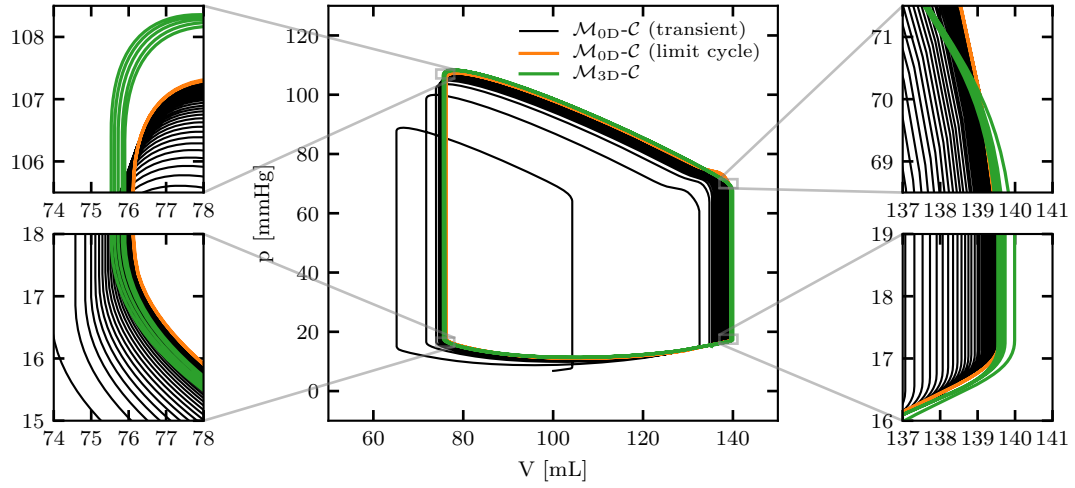


Figure 11: Test 2a: results of the 3D-0D-3D V-cycle (fitted EDPVR approach).

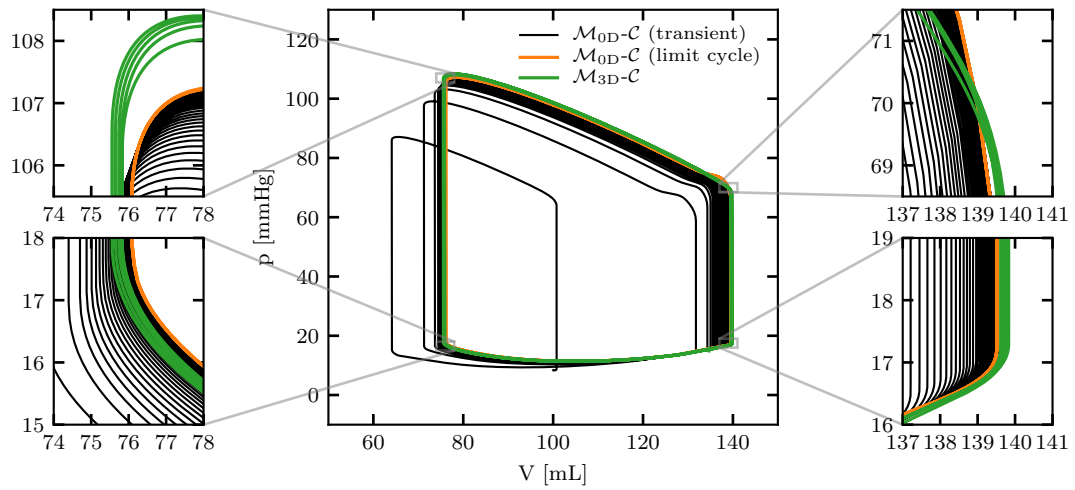


Figure 12: Test 2b: results of the 3D-0D-3D V-cycle (simulated EDPVR approach).

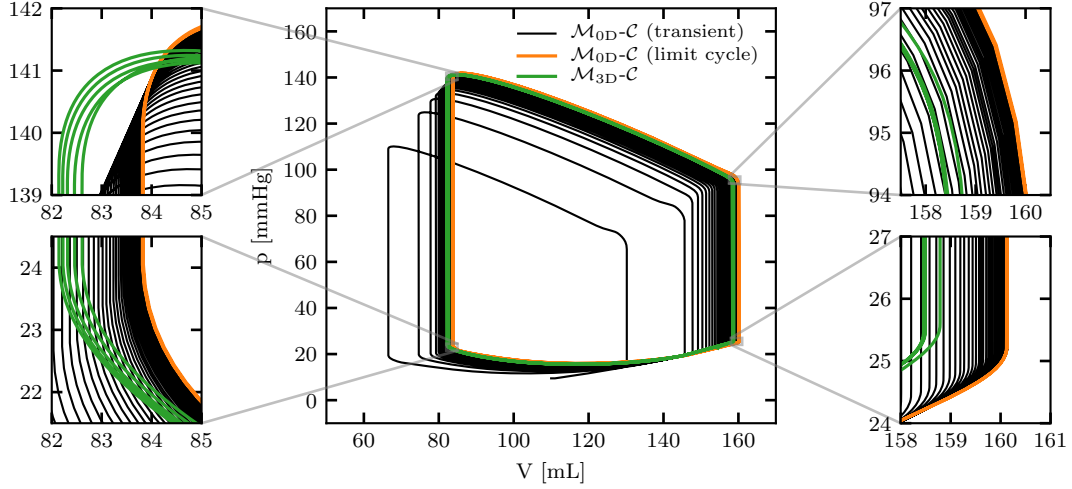


Figure 13: Test 3a: results of the 3D-0D-3D V-cycle (fitted EDPVR approach).

model at hand, contractility is tuned through a parameter – denoted as α_{XB} in the original paper [22] – that collectively accounts for the stiffness of crossbridges and for the surface density of myofilaments in the cardiac tissue. More precisely, we select two contractility values (namely $\alpha_A = 160$ MPa and $\alpha_A = 240$ MPa) and, by means of the \mathcal{M}_{3D-C} model, we obtain three sample PV loops for each of the two settings (step (S1)). Then, following the algorithm of Sec. 2.1, we build two emulators (one for each contractility value), respectively denoted by Ψ_A and Ψ_B (for simplicity we only consider the fitted EDPVR case). In this manner, we build a parametric emulator, according to Eq. (2). To test the parametric emulator, we select an intermediate contractility value (namely $\alpha = 190$ MPa), and we simulate $n_{HB}^{OD} = 100$ heartbeats through the \mathcal{M}_{0D-C} model (step (S2)). Finally, we perform step (S3) (by running simulations with the \mathcal{M}_{3D-C} model, initialized according to the results of the \mathcal{M}_{0D-C} model). We consider two settings: first, we employ the same \mathcal{C} model parameters as in the sample PV loops (Test 3a); then, we employ different \mathcal{C} model parameters (Test 3b). The results show that the parametric emulator is effective in predicting the location of the limit cycle of the \mathcal{M}_{3D-C} model even for a contractility value different than the ones used to build the parametric emulator (see Figs. 13 and 14).

4.3 Quantitative assessment of PV loop convergence

In the previous section, we showed numerical solutions obtained by applying the 3D-0D-3D V-cycle proposed in this paper. To quantitatively assess how close these solutions are to a limit cycle, we adopt as a metric the difference between two consecutive PV loops. Specifically, we denote the pressure and volume curves associated with the n -th cycle as

$$\begin{aligned} p_{LV}^n(t) &= p_{LV}(t + (n-1)T_{HB}), & t \in (0, T), \\ V_{LV}^n(t) &= V_{LV}(t + (n-1)T_{HB}), & t \in (0, T). \end{aligned}$$

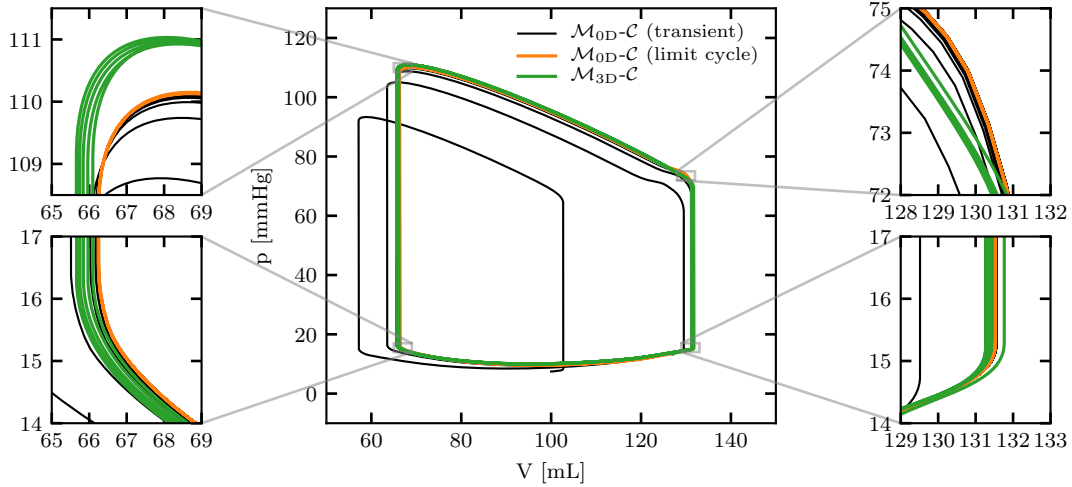


Figure 14: Test 4a: results of the 3D-0D-3D V-cycle (fitted EDPVR approach).

Then, we denote by e_n the overall normalized difference in volume and pressure between the n -th and the $(n + 1)$ -th heartbeats:

$$e_n = \frac{\|p_{LV}^n - p_{LV}^{n+1}\|_{L^2(0,T)}}{\|p_{LV}^{n+1}\|_{L^2(0,T)}} + \frac{\|V_{LV}^n - V_{LV}^{n+1}\|_{L^2(0,T)}}{\|V_{LV}^{n+1}\|_{L^2(0,T)}}.$$

The quantity e_n provides a metric for how close the solution is to the limit cycle. In fact, as we approach the periodic solution, e_n progressively decreases. If the solution were perfectly periodic, we would have $e_n = 0$.

In Fig. 15 we show the trend of e_n in the three phases of the 3D-0D-3D V-cycle for Test 1a. During step (S1), e_n rapidly decreases (orange line). During step (S2) (blue line), e_n first decreases following the values of step (S1) (as a further validation of the 0D surrogate), and then slows down the decrease, settling along an exponential trend. In the final phase of the curve, e_n shows some oscillations (due to the rounding errors becoming predominant). On the other hand, in this phase, the very small values of e_n testify that the model is virtually moving along a periodic orbit. Finally, the green line shows the quantity e_n obtained during step (S3). As shown in the figure, thanks to the intermediate simulation performed through the \mathcal{M}_{0D-C} model, the second of the two simulations obtained with the \mathcal{M}_{3D-C} model (i.e. step (S3)) starts much closer to a limit cycle than the first one (step (S1)). Already the second of the five beats performed in step (S3) has a value of e_n that in the simulation without educated guess would be obtained only after 23 beats. To emphasize the fact that step (S3) is obtained by virtue of three heartbeats simulated during step (S1), the heartbeats in step (S3) are represented in the figure starting with index four.

For the sake of brevity, for the other test cases besides Test 1a we only report the values of e_n obtained in step (S3). As shown in Fig. 16, the 3D-0D-3D V-cycle based strategy is able to bring the model very close to a periodic solution in all the considered cases. In particular, using different parameters for the circulation model and/or for the electromechanical model itself (thus using a parametric emulator) does not compromise the ability of the 3D-0D-3D V-cycle approach to accurately detect the limit-cycle. In all cases, in fact, the value of e_n obtained after only two heartbeats is as little as $2 \cdot 10^{-3} \div 3 \cdot 10^{-3}$.

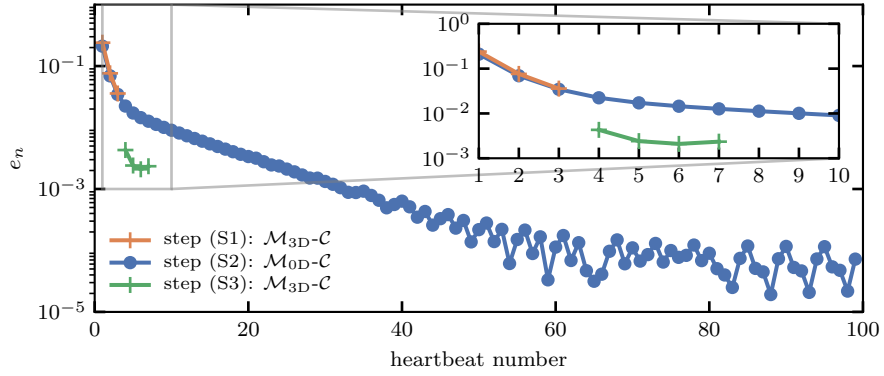


Figure 15: Trend of the metric e_n in the three steps of the 3D-0D-3V V-cycle in Test 1a (semilogarithmic scale).

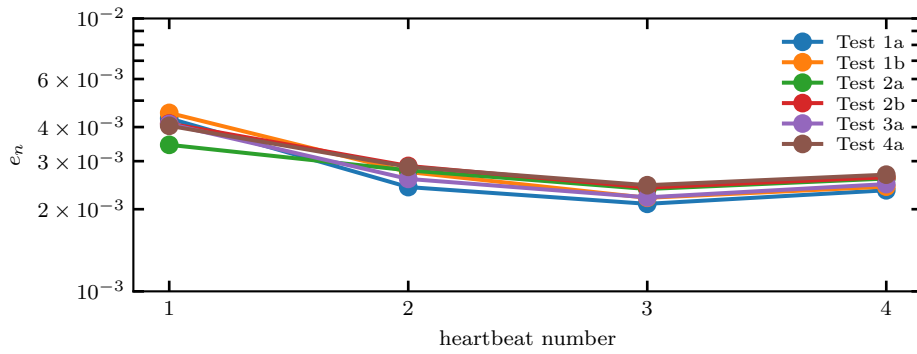


Figure 16: Trend of the metric e_n in step (S3) of the 3D-0D-3V V-cycle in the different test cases (semilogarithmic scale).

5 Conclusions

In this paper we proposed a 0D emulator of 3D cardiac electromechanical models, built following a data-driven approach, i.e. based on some sample PV loops generated with the 3D model to be surrogated. We then proposed a strategy, based on a 3D-0D-3D V-cycle, that allows to significantly accelerate the convergence to a limit cycle in cardiac electromechanics simulations. Our approach consists in surrogating the computationally demanding part (i.e. the 3D electromechanical model \mathcal{M}_{3D}), while keeping the 0D circulation model \mathcal{C} , characterized by a much lower computational cost, in its high-fidelity form. Remarkably, the 0D emulator can also be used in conjunction with a circulation model calibrated differently from the one used to generate the sample PV loops. Furthermore, by interpolating between different emulators, it is possible to approximate the electromechanical model even for wide sets of values of its parameters.

The results showed that our strategy allows to obtain an initial guess for the \mathcal{M}_{3D} - \mathcal{C} model that is very close to a periodic solution. In fact, in all the tests considered, by virtue of the simulation performed with the \mathcal{M}_{0D} - \mathcal{C} surrogate, after only two heartbeats the \mathcal{M}_{3D} - \mathcal{C} model presents a normalized increment between consecutive heartbeats lower than $3 \cdot 10^{-3}$. We remark that, without such educated guess, the same degree of convergence to the limit cycle would be reached in the same setting after more than 20 cycles. Conversely, thanks to our approach, we only need to simulate with the full-order \mathcal{M}_{3D} - \mathcal{C} model a total of 5 heartbeats (namely three to build the emulator and two starting from the educated guess). On the other hand, the cost associated with the 0D step of the 3D-0D-3D V-cycle is negligible (less than one second to build the emulator, less than one minute to run 100 heartbeats with the \mathcal{M}_{0D} - \mathcal{C} model). Moreover, this phase can be fully automated, as testified by the Python scripts publicly available with this publication.

We remark that the initialization strategy proposed in this paper is even more convenient in many-query settings, such as sensitivity analysis, parameter estimation or uncertainty quantification. As a matter of fact, in these settings the emulator can be built offline, i.e. once for all, and in the online phase only the final two heartbeats need to be simulated with the full-order \mathcal{M}_{3D} - \mathcal{C} model for each parameter setting to evaluate. Furthermore, if the quantities of interest involve only variables that can be derived from pressures and volumes, then the \mathcal{M}_{0D} - \mathcal{C} model can entirely replace the \mathcal{M}_{3D} - \mathcal{C} model in such many-query settings. Therefore, we propose our \mathcal{M}_{0D} - \mathcal{C} model as a surrogate for the full-order \mathcal{M}_{3D} - \mathcal{C} model in sensitivity analysis and parameter calibration scenarios.

Acknowledgements

This project has been supported by the Italian research project MIUR PRIN17 2017AXL54F “Modeling the heart across the scales: from cardiac cells to the whole organ”.

Appendix A Algorithm for the automated construction of the 0D emulator

In this appendix we present the three-step algorithm for the construction of the 0D emulator.

A.1 End-diastolic pressure-volume relationship

To build the EDPVR function Ψ_{ED} we propose two alternative approaches (simulated EDPVR or fitted EDPVR), depending on the available data.

Simulated EDPVR. When possible, an accurate estimation of the EDPVR can be obtained by means of the 3D mechanical model, by numerically finding the equilibrium configurations when the chamber is loaded with a sequence of pressures ($p_{\text{ED},i}$, for $i = 1, \dots, N_{\text{ED}}$), and recording the corresponding volumes ($V_{\text{ED},i}$). The active stress can be neglected, or – in case the force generation model at hand features a residual active stress in end-diastolic conditions – it should be set equal to this end-diastolic active stress magnitude (obtained through a 0D simulation of the cardiomyocyte model) uniformly in the whole domain Ω_0 . We remark that the computational cost of this phase can be significantly reduced by adopting a continuation approach, that is initializing the Newton solver for a given $p_{\text{ED},i}$ with the solution obtained with the previous value $p_{\text{ED},i-1}$. In this manner, the numerical cost associated with this phase is, in our experience, negligible compared with that required to obtain the sample PV loops.

We thus obtain a sequence of pressure and volume pairs (i.e. $(p_{\text{ED},i}, V_{\text{ED},i})$ for $i = 1, \dots, N_{\text{ED}}$). Finally, we define the function $p = \Psi_{\text{ED}}(V)$ as a piecewise linear interpolation of these points. Outside the interval $[V_{\text{ED},1}, V_{\text{ED},N_{\text{ED}}}]$, we define Ψ_{ED} through a linear extrapolation of the first and last segments.

Fitted EDPVR. Alternatively, we propose to estimate EDPVR directly from the sample PV loops, by fitting the points associated with the final parts of diastole with the Klotz curve [15]. These points can be selected either manually or automatically, by monitoring some activation indicator. In this paper, we employ the points associated with the last 0.1 s of each sample PV loop. Alternatively, one could select the time instants in which the average active tension is, say, in the lowest 1% of the range spanned during the whole sample simulation. The associated PV pairs are denoted as $(p_{\text{ED},i}, V_{\text{ED},i})$ for $i = 1, \dots, N_{\text{ED}}$. These points are then fitted with the Klotz curve

$$\Psi_{\text{ED}}(V) = A_n \left(\frac{V - V_0^{\text{ED}}}{V_{30}^{\text{ED}} - V_0^{\text{ED}}} \right)^{B_n}, \quad (7)$$

where, according to [15], we set the universal constants $A_n = 28.2 \text{ mmHg}$ and $B_n = 2.79$, while the patient-dependent constants V_0^{ED} and V_{30}^{ED} are obtained by a least-squares fitting of the points $(p_{\text{ED},i}, V_{\text{ED},i})$, via the Levenberg-Marquardt optimization method.

A.2 End-systolic pressure-volume relationship

For the ESPVR we assume the linear relationship

$$\Psi_{\text{ES}}(V) = E^{\text{ES}}(V - V_0^{\text{ES}}), \quad (8)$$

where the end-systolic elastance E^{ES} and resting volume V_0^{ES} are obtained via a least-squares fitting of a collection of points (denoted by $(p_{\text{ES},i}, V_{\text{ES},i})$ for $i = 1, \dots, N_{\text{ES}}$), corresponding to the systolic peaks of the sample PV loops. These points are selected as the top-left corners of the sample PV loops.

To identify the pairs $(p_{\text{ES},i}, V_{\text{ES},i})$, we propose the following algorithm. Ideally, PV loops are located below the curve $p = E^{\text{ES}}(V - V_0^{\text{ES}})$, intersecting such curve at the systolic peak. The corresponding time instant can thus be identified as the time t that maximizes the quantity $p(t) - E^{\text{ES}}V(t)$. As the end-systolic elastance E^{ES} is not known a priori, we select a sequence of 10 elastance values $E_1 < E_2 < \dots < E_{10}$, uniformly sampling the interval $[2, 3] \text{ mmHg mL}^{-1}$. Then, we separately consider each sample heartbeat and we denote by $I_n = [(n-1)T_{\text{HB}}, nT_{\text{HB}}]$ the time interval corresponding to the n -th cycle. For each heartbeat n , we select 10 points as

$$t_j^n = \arg \max_{t \in I_n} \{p(t) - E_j V(t)\}.$$

Finally, the end-systolic pressure-volume pairs are obtained by collecting the values in the time instants obtained above:

$$\{(p_{\text{ES},i}, V_{\text{ES},i}) : 1 \leq i \leq N_{\text{ES}}\} := \{(p(t_j^n), V(t_j^n)) : 1 \leq n \leq n_{\text{HB}}, 1 \leq j \leq 10\}.$$

A.3 Activation kinetics function

From Eq. (1) it follows

$$\varphi_{\text{act}}(t) = \frac{\Psi(V, t) - \Psi_{\text{ED}}(V)}{\Psi_{\text{ES}}(V) - \Psi_{\text{ED}}(V)}. \quad (9)$$

Therefore, denoting by (t_i, V_i, p_i) , for $i = 1, \dots, N_{\text{PV}}$, the times, volumes and pressures composing the sample PV loops dataset, we can define

$$\tilde{\varphi}_{\text{act}}(t_i) = \frac{p_i - \Psi_{\text{ED}}(V_i)}{\Psi_{\text{ES}}(V_i) - \Psi_{\text{ED}}(V_i)}, \quad \text{for } i = 1, \dots, N_{\text{PV}}, \quad (10)$$

having defined the functions Ψ_{ED} and Ψ_{ES} as described above. For t in the intervals (t_i, t_{i+1}) we employ a linear interpolation between $\tilde{\varphi}_{\text{act}}(t_i)$ and $\tilde{\varphi}_{\text{act}}(t_{i+1})$. Finally, we select the last heartbeat of the sample simulation – that is $(n_{\text{HB}} - 1)T_{\text{HB}} \leq t < n_{\text{HB}}T_{\text{HB}}$, where n_{HB} is the number of heartbeats in the sample simulation – and we periodically extend it outside this interval:

$$\varphi_{\text{act}}(t) = \tilde{\varphi}_{\text{act}}((n_{\text{HB}} - 1)T_{\text{HB}} + \text{mod}(t, T_{\text{HB}})). \quad (11)$$

This concludes our procedure to automatically build the 0D emulator $\mathcal{M}_{0\text{D}}$.

References

- [1] T. Arts, T. Delhaas, P. Bovendeerd, X. Verbeek, and F. W. Prinzen. “Adaptation to mechanical load determines shape and properties of heart and circulation: the CircAdapt model”. In: *American Journal of Physiology-Heart and Circulatory Physiology* 288.4 (2005), H1943–H1954.
- [2] C. M. Augustin, M. A. Gsell, E. Karabelas, and G. Plank. “Physiologically valid 3D-0D closed loop model of the heart and circulation—Modeling the acute response to altered loading and contractility”. In: *arXiv preprint arXiv:2009.08802* (2020).
- [3] B. Baillargeon, N. Rebelo, D. D. Fox, R. L. Taylor, and E. Kuhl. “The living heart project: a robust and integrative simulator for human heart function”. In: *European Journal of Mechanics-A/Solids* 48 (2014), pp. 38–47.
- [4] J. D. Bayer, R. C. Blake, G. Plank, and N. A. Trayanova. “A novel rule-based algorithm for assigning myocardial fiber orientation to computational heart models”. In: *Annals of Biomedical Engineering* 40.10 (2012), pp. 2243–2254.
- [5] 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.
- [6] M. Boulakia, S. Cazeau, M. A. Fernández, J.-F. Gerbeau, and N. Zemzemi. “Mathematical modeling of electrocardiograms: a numerical study”. In: *Annals of biomedical engineering* 38.3 (2010), pp. 1071–1097.
- [7] 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.

- [8] P. Colli Franzone, L. Pavarino, and S. Scacchi. *Mathematical cardiac electrophysiology*. Vol. 13. Springer, 2014.
- [9] 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.
- [10] 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.
- [11] T. Gerach, S. Schuler, J. Fröhlich, L. Lindner, E. Kovacheva, R. Moss, E. M. Wülfers, G. Seemann, C. Wieners, and A. Loewe. “Electro-Mechanical Whole-Heart Digital Twins: A Fully Coupled Multi-Physics Approach”. In: *Mathematics* 9.11 (2021).
- [12] A. Gerbi, L. Dedè, and A. Quarteroni. “A monolithic algorithm for the simulation of cardiac electromechanics in the human left ventricle”. In: *Mathematics in Engineering* 1.1 (2018), pp. 1–37.
- [13] J. M. Guccione, K. D. Costa, and A. D. McCulloch. “Finite element stress analysis of left ventricular mechanics in the beating dog heart”. In: *Journal of Biomechanics* 28.10 (1995), pp. 1167–1177.
- [14] 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.
- [15] S. Klotz, I. Hay, M. L. Dickstein, G.-H. Yi, J. Wang, M. S. Maurer, D. A. Kass, and D. Burkhoff. “Single-beat estimation of end-diastolic pressure-volume relationship: a novel method with potential for noninvasive application”. In: *American Journal of Physiology-Heart and Circulatory Physiology* 291.1 (2006), H403–H412.
- [16] F Levrero-Florencio, F Margara, E Zacur, A Bueno-Orovio, Z. Wang, A Santiago, J Aguado-Sierra, G Houzeaux, V Grau, D Kay, et al. “Sensitivity analysis of a strongly-coupled human-based electromechanical cardiac model: Effect of mechanical parameters on physiologically relevant biomarkers”. In: *Computer methods in applied mechanics and engineering* 361 (2020), p. 112762.
- [17] E. Lluch, O. Camara, R. Doste, B. Bijmens, M. De Craene, M. Sermesant, V. Y. Wang, M. P. Nash, and H. G. Morales. “Calibration of a fully coupled electromechanical meshless computational model of the heart with experimental data”. In: *Computer Methods in Applied Mechanics and Engineering* 364 (2020), p. 112869.
- [18] J. Lumens, T. Delhaas, B. Kirn, and T. Arts. “Three-wall segment (TriSeg) model describing mechanics and hemodynamics of ventricular interaction”. In: *Annals of biomedical engineering* 37.11 (2009), pp. 2234–2255.
- [19] I Mirsky. “Left ventricular stresses in the intact human heart”. In: *Biophysical journal* 9.2 (1969), pp. 189–208.
- [20] 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.

- [21] 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.
- [22] F. Regazzoni, L. Dedè, and A. Quarteroni. “Biophysically detailed mathematical models of multiscale cardiac active mechanics”. In: *PLOS Computational Biology* 16.10 (2020), e1008294.
- [23] 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.
- [24] 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: *arXiv preprint arXiv:2011.15040* (2020).
- [25] 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 II: numerical approximation”. In: *arXiv preprint arXiv:2011.15051* (2020).
- [26] 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* 373 (2021), p. 113506.
- [27] N. Smith, D. Nickerson, E. Crampin, and P. Hunter. “Multiscale computational modelling of the heart”. In: *Acta Numerica* 13 (2004), pp. 371–431.
- [28] S. Sugiura, T. Washio, A. Hatano, J. Okada, H. Watanabe, and T. Hisada. “Multi-scale simulations of cardiac electrophysiology and mechanics using the University of Tokyo heart simulator”. In: *Progress in Biophysics and Molecular Biology* 110.2 (2012), pp. 380–389.
- [29] 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* 291.3 (2006), H1088–H1100.
- [30] N. A. Trayanova. “Whole-heart modeling applications to cardiac electrophysiology and electromechanics”. In: *Circulation Research* 108 (2011), pp. 113–128.
- [31] T. P. Usyk, I. J. LeGrice, and A. D. McCulloch. “Computational model of three-dimensional cardiac electromechanics”. In: *Computing and Visualization in Science* 4.4 (2002), pp. 249–257.
- [32] E. J. Vigmond, C. Clements, D. M. McQueen, and C. S. Peskin. “Effect of bundle branch block on cardiac output: a whole heart simulation study”. In: *Progress in biophysics and molecular biology* 97.2-3 (2008), pp. 520–542.
- [33] T. Washio, K. Yoneda, J. Okada, T. Kariya, S. Sugiura, and T. Hisada. “Ventricular fiber optimization utilizing the branching structure”. In: *International Journal for Numerical Methods in Biomedical Engineering* (2015), e02753.
- [34] J. Xu. *Theory of multilevel methods*. Vol. 8924558. Cornell University Ithaca, NY, 1989.
- [35] *Zygote 3D models*. 2019. URL: <https://www.zygote.com/>.

MOX Technical Reports, last issues

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

- 34/2021** Bonaventura, L.; Gatti F.; Menafoglio A.; Rossi D.; Brambilla D.; Papini M.; Longoni L.
An efficient and robust soil erosion model at the basin scale
- 33/2021** Lupo Pasini, M.; Gabbi, V.; Yin, J.; Perotto, S.; Laanait, N.
Scalable balanced training of conditional generative adversarial neural networks on image data
- 32/2021** Sangalli, L.M.
Spatial regression with partial differential equation regularization
- 31/2021** Ferraccioli, F.; Arnone, E.; Finos, L.; Ramsay, J.O.; Sangalli, L.M.
Nonparametric density estimation over complicated domains
- 30/2021** Fumagalli, I.
A reduced 3D-0D FSI model of the aortic valve including leaflets curvature
- 29/2021** Fumagalli, I.; Vitullo, P.; Scrofani, R.; Vergara, C.
Image-based computational hemodynamics analysis of systolic obstruction in hypertrophic cardiomyopathy
- 28/2021** Ferro, N.; Perotto, S.; Bianchi, D.; Ferrante, R.; Mannisi, M.
Design of cellular materials for multiscale topology optimization: application to patient-specific orthopedic devices
- 26/2021** Vigano, L.; Sollini, M.; Ieva, F.; Fiz, F.; Torzilli, G.
Chemotherapy-Associated Liver Injuries: Unmet Needs and New Insights for Surgical Oncologists
- 27/2021** Scimone, R.; Menafoglio, A.; Sangalli, L.M.; Secchi, P.
A look at the spatio-temporal mortality patterns in Italy during the COVID-19 pandemic through the lens of mortality densities
- 24/2021** Regazzoni, F.; Chapelle, D.; Moireau, P.
Combining Data Assimilation and Machine Learning to build data-driven models for unknown long time dynamics - Applications in cardiovascular modeling