Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/compbiomed

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

F. Regazzoni [a,*], A. Quarteroni [a,b]

[a] *MOX - Dipartimento di Matematica, Politecnico di Milano, P.zza Leonardo da Vinci 32, 20133, Milano, Italy*

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

ARTICLE INFO

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

ABSTRACT

The results of numerical simulations of cardiac electromechanics are typically characterized by a long transient before reaching a periodic solution known as limit cycle. This yields a serious computational overhead, as the only clinically relevant output is associated with such limit cycle. 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, built through an automated data-driven algorithm on the basis of pressure-volume transients of as few as three heartbeats simulated with the 3D model. The 0D emulator, consisting of a time-dependent pressure-volume relationship, can provide the 3D model with an initial guess, such that in just two heartbeats a solution is reached that is as close to the limit cycle as the one obtained after more than 20 heartbeats with the 3D model. 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. Indeed, the construction of the emulator does not have to be repeated when the parameters of the circulation model it is coupled with 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. This paper is accompanied by a Python library implementing the proposed algorithm, open to integration with existing cardiac solvers.

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 relatioships between microscopic variations and macroscopic biomarkers, to perform *in silico* investigations on the effects of drugs or therapies [1–11]. 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* [12–14]. 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 [15,16]; on the other hand, the coupling with closed-loop

[*] Corresponding author.
E-mail address: francesco.regazzoni@polimi.it (F. Regazzoni).

https://doi.org/10.1016/j.compbiomed.2021.104641
Received 2 June 2021; Received in revised form 6 July 2021; Accepted 6 July 2021
Available online 13 July 2021

0D (i.e. lumped-parameter) models of the external circulation [12–14, 17,18]. 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 [19]. 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 [18]. 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 [13,14,19]. 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 [14], 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 [13], 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 [14]. However, in [20] 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 [16,21], 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 heartbeats obtained with the 3D model. Once the emulator 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 [22]. 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 repository[1] 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 discuss the obtained results in Sec. 5 and we draw our conclusions and final remarks in Sec. 6.

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 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 $\mathscr{M}_{3D}$ to refer to the 3D electromechanical model. The $\mathscr{M}_{3D}$ 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 $\mathscr{M}_{3D}$ 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 $\mathscr{C}$ to denote these closure relationships, whether they are in the form of afterlead-preload models, or closed-loop circulation models. Coupling the electromechanical model $\mathscr{M}_{3D}$ with the circulation model $\mathscr{C}$, we then obtain the model $\mathscr{M}_{3D} - \mathscr{C}$.

2.1. Building the 0D emulator

In this paper we propose a 0D emulator of the $\mathscr{M}_{3D}$ model – that we will denote as $\mathscr{M}_{0D}$ – 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 $p = \mathscr{P}(V, t)$. The main assumption under the 0D emulator is that the pressure-volume relationship can be factorized as follows

$$\mathscr{P}(V,t) = (1 - \varphi_{\mathrm{act}}(t))\,\mathscr{P}_{\mathrm{ED}}(V) + \varphi_{\mathrm{act}}(t)\,\mathscr{P}_{\mathrm{ES}}(V), \tag{1}$$

where $\mathscr{P}_{\mathrm{ES}}$ represents the end-systolic pressure-volume relationship (ESPVR) and $\mathscr{P}_{\mathrm{ED}}$ represents the end-diastolic pressure-volume relationship (EDPVR). The function φ_{act} is a time-dependent function encoding the activation kinetics. Ideally, $\varphi_{\mathrm{act}} = 0$ at the end of diastole (the tissue is fully relaxed), while $\varphi_{\mathrm{act}} = 1$ 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. $\mathscr{P}_{\mathrm{ED}}$, $\mathscr{P}_{\mathrm{ES}}$ and φ_{act}) from the output of a simulation obtained with the $\mathscr{M}_{3D} - \mathscr{C}$ 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

[1] https://github.com/FrancescoRegazzoni/cardioemulator.

region of the pressure-volume plane. The algorithm to construct the 0D emulator is described in detail in App. 7. It envisages three steps, visually summarized in Fig. 1 and briefly described as follows.

(a) First, we construct the EDPVR function $\mathscr{P}_{\mathrm{ED}}(V)$, 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 $\mathscr{M}_{3\mathrm{D}}$ 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 [23] on the end-diastolic segments of the sample PV loops.
(b) Then, we obtain the ESPVR function $\mathscr{P}_{\mathrm{ES}}(V)$ by fitting the points corresponding to the systolic peaks with a straight line.
(c) Finally, we construct the activation kinetics function $\varphi_{\mathrm{act}}(t)$, inverting the relationship Eq. (1) with respect to $\varphi_{\mathrm{act}}(t)$ on the data corresponding to the last cycle, and periodically extending it outside the interval $[0, T_{\mathrm{HB}})$.

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

The $\mathscr{M}_{0\mathrm{D}}$ model surrogates the pressure-volume relationship of the $\mathscr{M}_{3\mathrm{D}}$ model. Hence, similarly to the $\mathscr{M}_{3\mathrm{D}}$ model, also the $\mathscr{M}_{0\mathrm{D}}$ model can be coupled with the closure relationships $\mathscr{C}$, 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 $\mathscr{M}_{0\mathrm{D}} - \mathscr{C}$, thus represents a surrogate of the $\mathscr{M}_{3\mathrm{D}} - \mathscr{C}$ model. We remark that, in case the $\mathscr{M}_{3\mathrm{D}}$ 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 $\mathscr{M}_{0\mathrm{D}} - \mathscr{C}$ model can be exploited to accelerate the search for a limit cycle of the $\mathscr{M}_{3\mathrm{D}} - \mathscr{C}$ 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 $\mathscr{M}_{3\mathrm{D}} - \mathscr{C}$ model.
(S2) Following Sec. 2.1, a 0D emulator $\mathscr{M}_{0\mathrm{D}}$ is built for each cardiac chamber included in the model, based on the sample PV loops. Then, a large number of heartbeats (say $n_{\mathrm{HB}}^{0\mathrm{D}}$) are simulated by solving the $\mathscr{M}_{0\mathrm{D}} - \mathscr{C}$ model, at a negligible computational cost, until a limit cycle is reached.
(S3) A second simulation is set up for the $\mathscr{M}_{3\mathrm{D}} - \mathscr{C}$ model, based on the limit cycle obtained at the previous step. Specifically, in case the $\mathscr{C}$ has some internal variables, these are set equal to their final state obtained in step 2.

Moreover, also the $\mathscr{M}_{3\mathrm{D}}$ 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 $\mathscr{M}_{0\mathrm{D}} - \mathscr{C}$ model simulation of step 2.
- **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 $\mathscr{M}_{0\mathrm{D}} - \mathscr{C}$ model simulation of step 2. In this case, the endocardial pressures are determined as Lagrange multipliers.

Both in 1 and 3, 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.

2.3. Using the emulator in many-query settings

We remark that the $\mathscr{M}_{0\mathrm{D}}$ emulator surrogates the $\mathscr{M}_{3\mathrm{D}}$ model, independently of the $\mathscr{C}$ model it is coupled with. This implies that, in case the parameters of the $\mathscr{C}$ model are changed, 1 does not need to be repeated. Therefore, the $\mathscr{M}_{0\mathrm{D}} - \mathscr{C}$ model can be employed as a surrogate of $\mathscr{M}_{3\mathrm{D}} - \mathscr{C}$ in many-query contexts that involve the repeated solution of the $\mathscr{M}_{3\mathrm{D}} - \mathscr{C}$ 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 $\mathscr{M}_{3\mathrm{D}}$ may require calibration. In case the parameters of the $\mathscr{C}$ and of the $\mathscr{M}_{3\mathrm{D}}$ models need to be simultaneously calibrated, we propose to build a parametric emulator of the $\mathscr{M}_{3\mathrm{D}}$ model, by interpolating between emulators obtained for fixed values of the parameters of the $\mathscr{M}_{3\mathrm{D}}$ model.

A typical case is when the active contractility (henceforth denoted by α) is to be determined. By the Young-Laplace law [24], 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

$$\mathscr{P}(V,t;\alpha) = \frac{\alpha-\alpha_A}{\alpha_B-\alpha_A}\mathscr{P}_B(V,t) + \frac{\alpha-\alpha_B}{\alpha_A-\alpha_B}\mathscr{P}_A(V,t), \tag{2}$$

where $\mathscr{P}_A$ and $\mathscr{P}_B$ 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 $\mathscr{P}_A$ and $\mathscr{P}_B$, Eq. (2) defines a parametric emulator where the contractility α can assume arbitrary values.

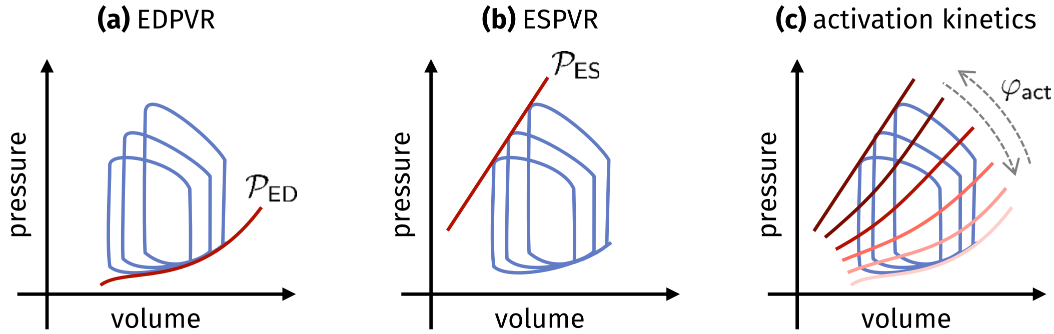

Fig. 1. The algorithm to construct the 0D emulator comprises three steps: (a) construction of the EDPVR (end-diastolic pressure-volume relationship) function $\mathscr{P}_{\mathrm{ED}}(V)$; (b) construction of the ESPVR (end-systolic pressure-volume relationship) function $\mathscr{P}_{\mathrm{ES}}(V)$;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

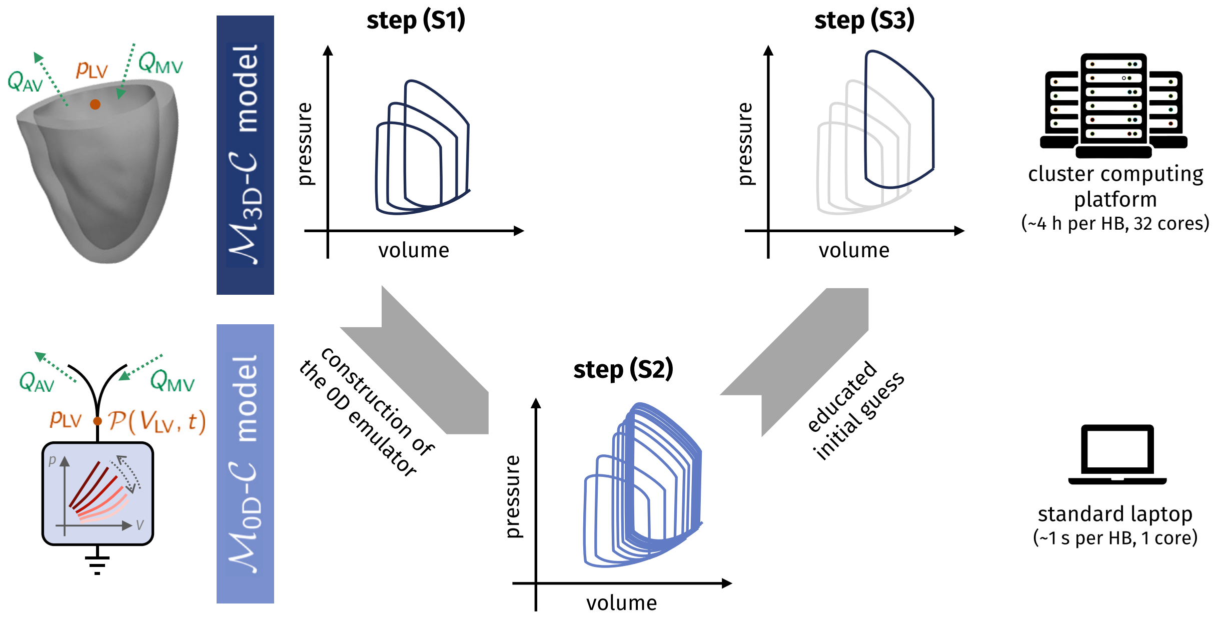

Fig. 2. Graphical display of the 3D-0D-3D V-cycle, in the case of a left ventricle (LV) electromechanical model. On the left, the high-fidelity 3D electromechanical model is compared with its 0D surrogate. At the terminals of both models we have the chamber pressure (p_{LV}) and the inflow and outflow blood flux (in the case of LV, the flows through the mitral and aortic valves, respectively denoted by Q_{MV} and Q_{AV}). In the right column we report, for illustrative purposes, the computational times required to simulate a heartbeat (HB) with the numerical models and on the computational platform used to produce the results of this paper (see Sec. 3.2).

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. Some use case scenarios enabled by our 0D emulator in many-query settings are illustrated in Sec. 5.

2.4. *Quantitative assessment of PV loop convergence*

To quantitatively assess how close a PV transient is 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

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

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. We remark that, should the solution be perfectly periodic, we would have $e_n = 0$.

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 $\mathscr{C}$ are represented by a closed-loop circulation model. Then, in Section 3.2, we present the $\mathscr{M}_{3D}$ 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 $\mathscr{M}_{3D}$ model. The state $\mathbf{y}$ may contain variables describing the electrophysiological activity, the subcellular dynamics and the tissue displacement. The vector $\mathbf{y}(t)$ may contain from a dozen to some tens of variables, depending on the models at hand [12,16,18]. 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 $\mathscr{C}$), 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 $\mathscr{M}_{3D}$ model. Examples are given in [17,18,25,26].

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

$$\begin{cases} \dfrac{\partial \mathbf{y}(t)}{\partial t} = \mathcal{L}(\mathbf{y}(t), p_{LV}(t), t) & \text{for } \ t \in (0,T], \\ \dfrac{d\mathbf{c}(t)}{dt} = \mathbf{f}(\mathbf{c}(t), p_{LV}(t), t) & \text{for } \ t \in (0,T], \\ V_{LV}^{\mathscr{M}}(\mathbf{y}(t)) = V_{LV}^{\mathscr{C}}(\mathbf{c}(t)) & \text{for } \ t \in (0,T], \\ \mathbf{y}(0) = \mathbf{y}_0, \\ \mathbf{c}(0) = \mathbf{c}_0. \end{cases} \tag{3}$$

The electromechanical and the circulation models are coupled through the volumetric consistency constraint $V_{LV}^{\mathscr{C}}(\mathbf{c}) = V_{LV}^{\mathscr{M}}(\mathbf{y})$, where the left-hand and right-hand sides represent the LV volume predicted, respectively, by the 0D circulation model ($\mathscr{C}$) and by the 3D electromechanical model ($\mathscr{M}_{3D}$). 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 $\mathscr{C}$

model and the $\mathscr{M}_{3D}$ model (in the latter, e.g., it plays a role in the boundary conditions at the endocardium). The $\mathscr{M}_{3D} - \mathscr{C}$ model is represented in Fig. 3 (top).

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

$$\begin{cases} \dfrac{d\mathbf{c}(t)}{dt} = \mathbf{f}(\mathbf{c}(t), p_{\mathrm{LV}}(t), t) & \text{for } t \in (0, T], \\ p_{\mathrm{LV}}(t) = \mathscr{P}(V_{\mathrm{LV}}^{\mathscr{C}}(\mathbf{c}(t)), t) & \text{for } t \in (0, T], \\ \mathbf{c}(0) = \mathbf{c}_0. \end{cases} \tag{4}$$

This $\mathscr{M}_{0D} - \mathscr{C}$ model represents a fully 0D surrogate of the $\mathscr{M}_{3D} - \mathscr{C}$ 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 $\mathscr{M}_{3D} - \mathscr{C}$ to $\mathscr{M}_{0D} - \mathscr{C}$ we only replaced the computationally intensive part (that is the 3D electromechanical model $\mathscr{M}_{3D}$) by the 0D surrogate $\mathscr{M}_{0D}$, while retaining the computationally lightweight part (that is the 0D circulation model $\mathscr{C}$) in its high-fidelity form. We will show that, in this way, the $\mathscr{M}_{0D} - \mathscr{C}$ model achieves a very favorable trade-off between accuracy and computational cost.

In case the $\mathscr{M}_{3D}$ 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 $\mathscr{M}_{0D} - \mathscr{C}$ model reads

$$\begin{cases} \dfrac{d\mathbf{c}(t)}{dt} = \mathbf{f}(\mathbf{c}(t), p_{\mathrm{LV}}(t), p_{\mathrm{RV}}(t), t) & \text{for } t \in (0, T], \\ p_{\mathrm{LV}}(t) = \mathscr{P}_{\mathrm{LV}}(V_{\mathrm{LV}}^{\mathscr{C}}(\mathbf{c}(t)), t) & \text{for } t \in (0, T], \\ p_{\mathrm{RV}}(t) = \mathscr{P}_{\mathrm{RV}}(V_{\mathrm{RV}}^{\mathscr{C}}(\mathbf{c}(t)), t) & \text{for } t \in (0, T], \\ \mathbf{c}(0) = \mathbf{c}_0, \end{cases} \tag{5}$$

where $\mathscr{P}_{\mathrm{LV}}$ and $\mathscr{P}_{\mathrm{RV}}$ 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 ($\mathscr{M}_{3D}$) and closed-loop circulation ($\mathscr{C}$) 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 [27], coupled with the *tenn Tusscher-Panfilov* model [28]. Active force generation is described through the biophysically detailed RDQ20-MF model [29]. To model the passive behavior of the cardiac tissue, we employ the quasi-incompressible exponential *Usyk* constitutive law [30]. 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 [31]. The geometry is taken from the Zygote Solid 3D heart model [32], for which the stress-free reference configuration is recovered by the algorithm proposed in [18]. The distribution of cardiac fibers is generated by means of the rule based Bayer-Blake-Plank-Trayanova algorithm [33, 34]. With the models at hand, the state vector $\mathbf{y}(t)$ comprises 42 variables (18 ionic variables, the transmembrane potential, 20 activation variables and 3 variables to describe the tissue displacement). For blood circulation (i.e. $\mathscr{C}$) we employ the closed-loop lumped-parameter model of [18], 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 [35]. Additionally, we adopt the stabilized-staggered scheme of [36] to couple the active stress model with the passive mechanics one.

In 1 and 3, 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 $SL = 2.2$ μm. Unlike the ionic model, the force generation model reaches the equilibrium at the end of each heartbeat. Therefore, in this case it is not necessary to find a limit cycle by running a high number of cycles; rather, it is sufficient to run a simulation lasting a few seconds so that an equilibrium 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.

To produce the numerical results of this paper, we employ a finite element approach featuring a computational mesh with 3.60×10^4 vertices for the mechanical variables and a finer mesh (2.60×10^5 vertices) for the electrophysiological variables. To advance the latter variables, we use a time step length of 0.1 ms, while for the former variables we use a five times larger timestep. Overall, the variables of the $\mathscr{M}_{3D} - \mathscr{C}$ electromechanical model amount to 5.7 millions for each time step. The simulation are obtained with the high-performance C++ library life$^{\mathrm{x}}$ (https://lifex.gitlab.io/lifex), developed within the iHEART project.[2] To run these large-scale numerical simulations we rely on the iHEART cluster (Lenovo SR950 192-Core Intel Xeon Platinum 8160, 2100 MHz, 1.7 TB RAM) at MOX, Dipartimento di Matematica, Politecnico di Milano, using 32 cores. On this computer platform, numerical simulations take approximately 4 h per heartbeat.

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 $\mathscr{M}_{3D} - \mathscr{C}$ model described in Sec. 3.2, we run a simulation of $N_{\mathrm{PV}} = 3$ 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. 7.1. We show in Fig. 4 the functions $\mathscr{P}_{\mathrm{ED}}(V)$, $\mathscr{P}_{\mathrm{ES}}(V)$ and $\varphi_{\mathrm{act}}(t)$ obtained with the two different approaches. We remark that the two approaches provide the same function $\mathscr{P}_{\mathrm{ES}}(V)$, while the function $\varphi_{\mathrm{act}}(t)$, being defined based on $\mathscr{P}_{\mathrm{ED}}(V)$, is slightly different in the two cases.

The triplet $(\mathscr{P}_{\mathrm{ED}}, \mathscr{P}_{\mathrm{ES}}, \varphi_{\mathrm{act}})$ 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 $\varphi_{\mathrm{act}}'(t) > 0$) are shown. The different shades of red represent

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

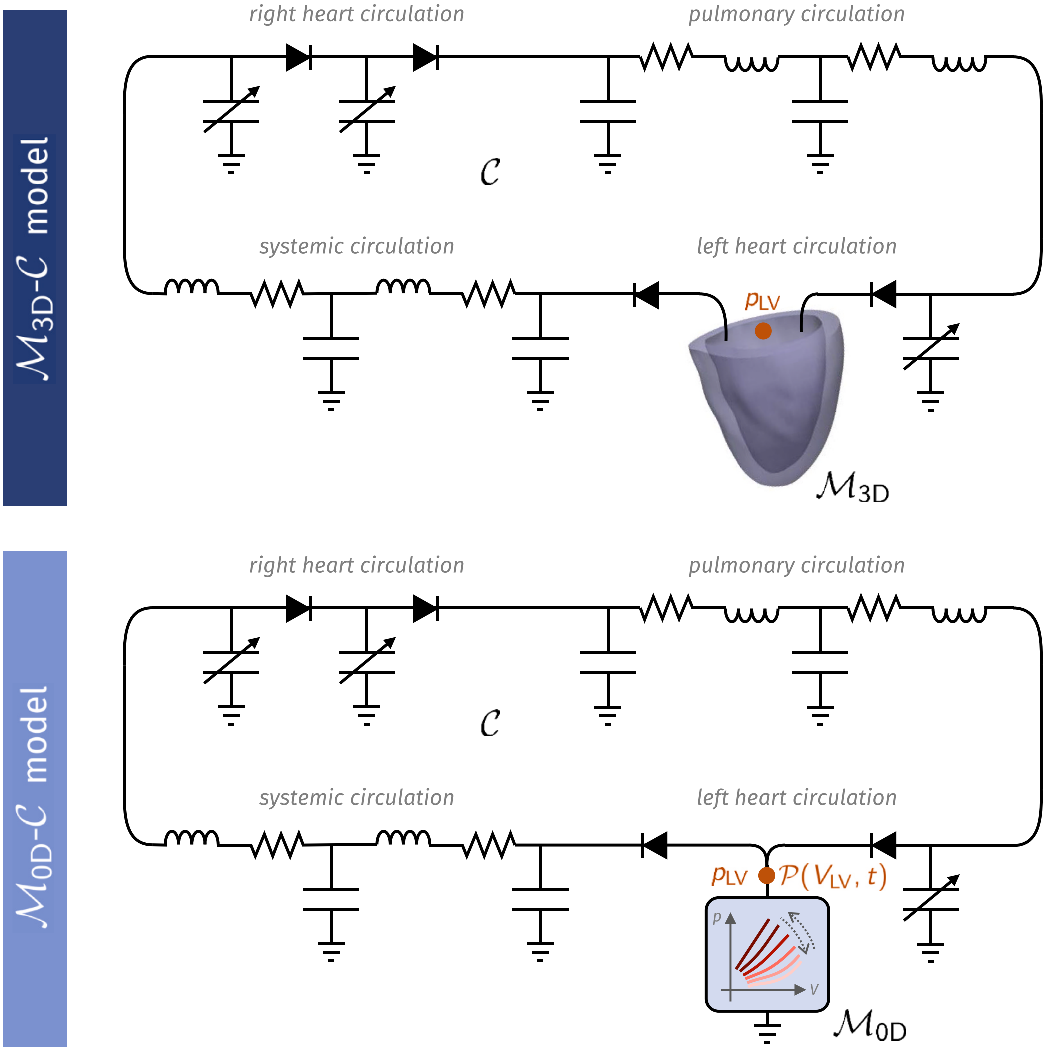

Fig. 3. Comparison of the $\mathscr{M}_{3D} - \mathscr{C}$ (top) and the $\mathscr{M}_{0D} - \mathscr{C}$ (bottom) models. In the former, a 3D electromechanical model (denoted by $\mathscr{M}_{3D}$) of one or multiple cardiac chambers is coupled with a lumped-parameter model of blood circulation (denoted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

different instants of the heartbeat. Similarly, the lower left region shows - with different shades of blue - the curves associated with diastole (when $\varphi_{act}{}'(t) < 0$). These curves are denser in the lower part, where φ_{act} varies more slowly. An alternative representation is shown in Fig. 6, where $\mathscr{P}$ 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 relationship can be factorized as in Eq. (1). If this assumption was exact, than superimposing the function

$$\widetilde{\varphi}_{act}(t) = \frac{\mathscr{P}(V(t), t) - \mathscr{P}_{ED}(V(t))}{\mathscr{P}_{ES}(V(t)) - \mathscr{P}_{ED}(V(t))}. \tag{6}$$

for different heartbeats, all the curves would coincide. That is to say, the function $\widetilde{\varphi}_{act}$ would be periodic with period T_{HB}. In Fig. 7, we show the function $\widetilde{\varphi}_{act}(t)$ 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 $\mathscr{M}_{0D}$ emulator to faithfully replicate the behavior of the $\mathscr{M}_{3D}$ model when coupled with the $\mathscr{C}$ model, we solve the $\mathscr{M}_{0D} - \mathscr{C}$ model starting from the same initial conditions used for the $\mathscr{M}_{3D} - \mathscr{C}$ model to produce the sample PV loops. In Fig. 8, we compare the PV loops obtained with the $\mathscr{M}_{3D} - \mathscr{C}$ model and with its fully 0D surrogate, the $\mathscr{M}_{0D} - \mathscr{C}$ 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. In Table 1 we report the relative mean square errors in the pressure and volume transients obtained with the $\mathscr{M}_{0D} - \mathscr{C}$ model instead of the $\mathscr{M}_{3D} - \mathscr{C}$ model. 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. Concerning computational times, the construction of the 0D emulator takes less than 1 s on a standard laptop and the numerical simulations performed with the $\mathscr{M}_{0D} - \mathscr{C}$ model take nearly 40 s for 100 heartbeats. This brings a great computational gain with respect to the full-order $\mathscr{M}_{3D} - \mathscr{C}$ model, that requires nearly 4 h per heartbeat on a 32 cores cluster computing platform (see Sec. 3.2).

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

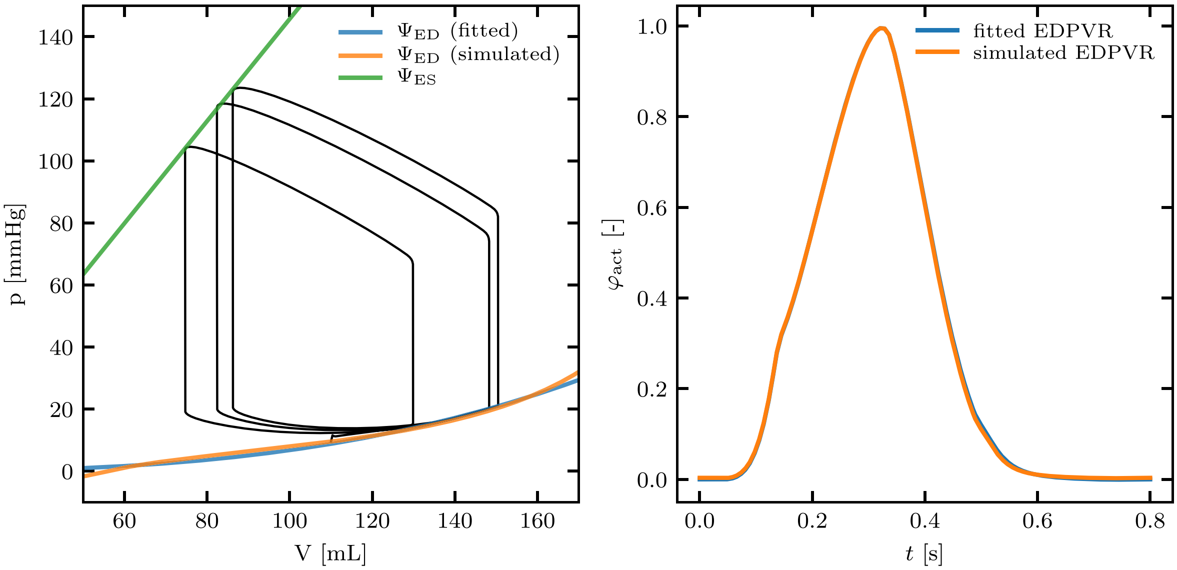

Fig. 4. Representation of the 0D emulators obtained with the *fitted EDPVR* and with the *simulated EDPVR* approaches. Left: end-diastolic and end-systolic PV relationships. The black curve represents the sample PV loops used to build the emulator. Rights: activation kinetics function $\varphi_{act}(t)$,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

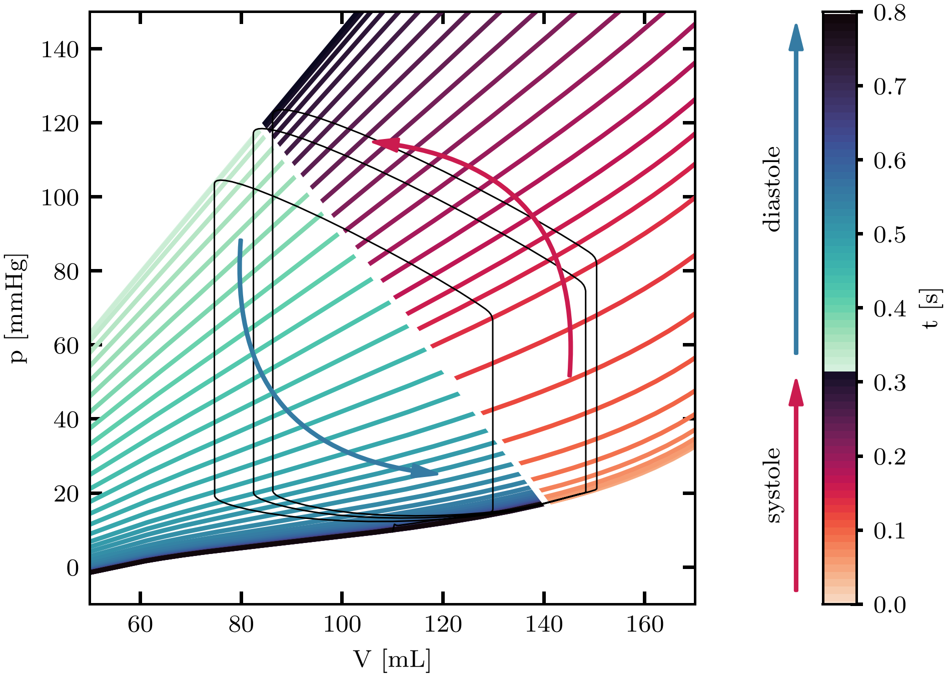

Fig. 5. Evolution of the PV relationships in the different phases of the heartbeat. See text for more details. The figure refers to the emulator obtained with the simulated EDPVR approach.

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 Table 2.

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, 1 b). The results are shown in Figs. 9 and 10Figs. 9 and 10, respectively. In the figures, the $n_{HB}^{0D} = 100$ heartbeats performed in a fully 0D manner (that is, through the $\mathscr{M}_{0D} - \mathscr{C}$ model) are shown in black. The last cycle, corresponding to the limit cycle of the $\mathscr{M}_{0D} - \mathscr{C}$ model, is highlighted in orange. The $\mathscr{M}_{3D} - \mathscr{C}$ 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 $\mathscr{M}_{3D} - \mathscr{C}$ model initialized in this way is very close to a periodic regime: indeed, no significant drift in the PV loops obtained with the $\mathscr{M}_{3D} - \mathscr{C}$ model is visible. In Sec. 4.3 we will provide a quantitative analysis of this aspect. We observe that the emulator built

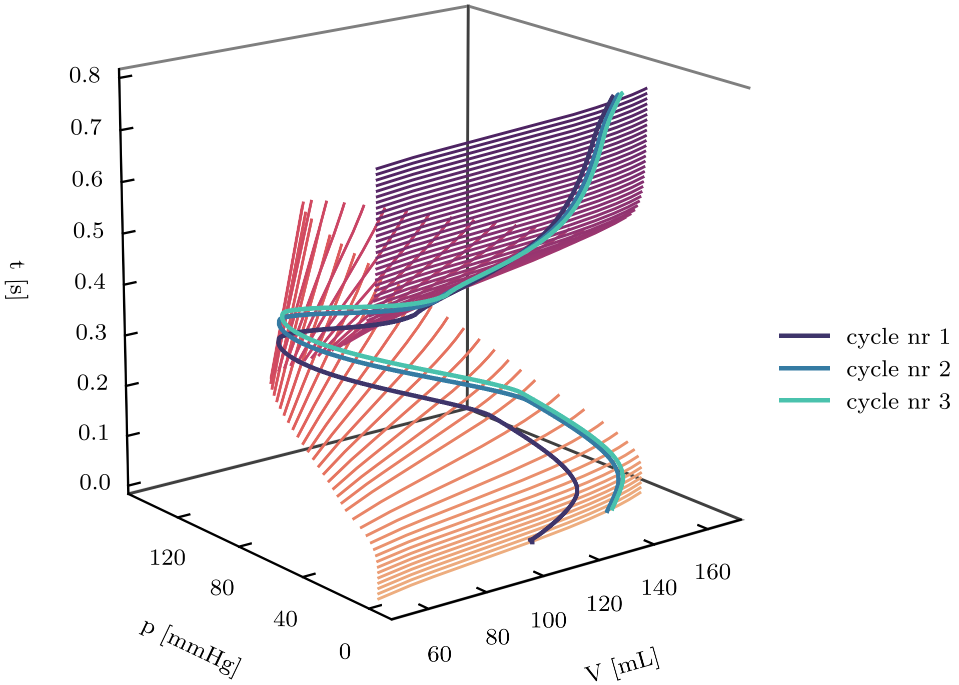

Fig. 6. Function $\mathscr{P}$ represented in the (t, V, p) plane. The figure refers to the emulator obtained with the simulated EDPVR approach.

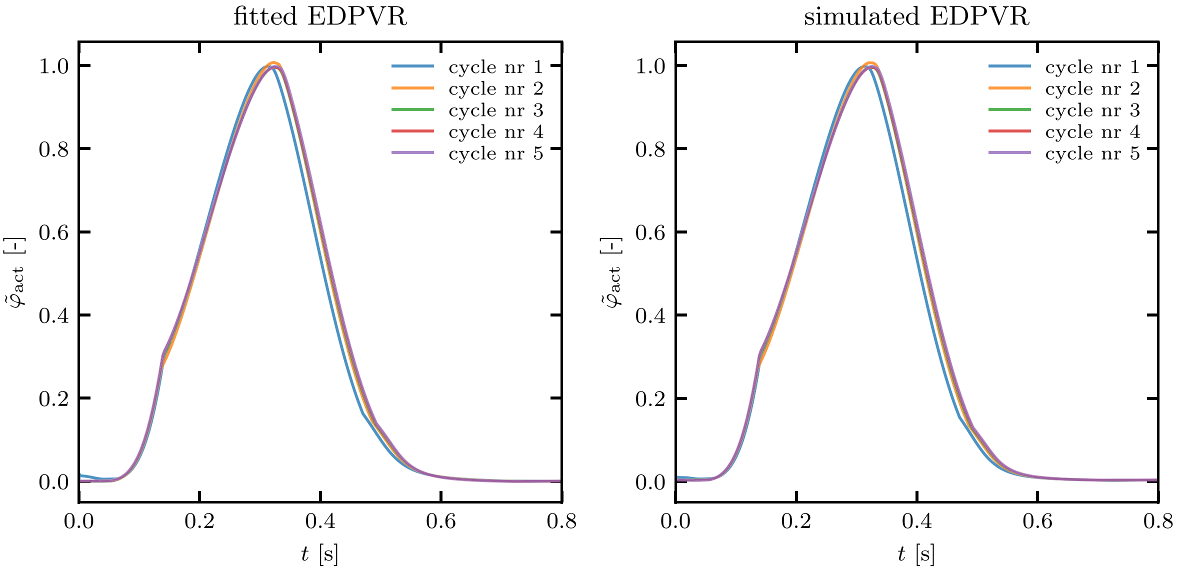

Fig. 7. Function $\tilde{\varphi}_{\text{act}}$, as defined in (6), obtained for 5 consecutive heartbeats with the emulators built with the *fitted EDPVR* (left) and the *simulated EDPVR* approaches (right).

by the simulated EDPVR approach (Test 1 b) better predicts the end-diastolic part of the PV loops than that build 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 $\mathscr{M}_{\text{0D}}$ emulator to effectively surrogate the $\mathscr{M}_{\text{3D}}$ 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 1 b. By using this setting, we perform steps 2 and 3 of the 3D-0D-3D V-cycle, with both emulators (fitted and simulated EDPVR), leading to Tests 2a and 2 b, respectively. The results, shown in Figs. 11 and 12, reveal that the $\mathscr{M}_{\text{0D}}$ emulator can effectively detect the position of the limit cycle associated with a setting even different from the one used to

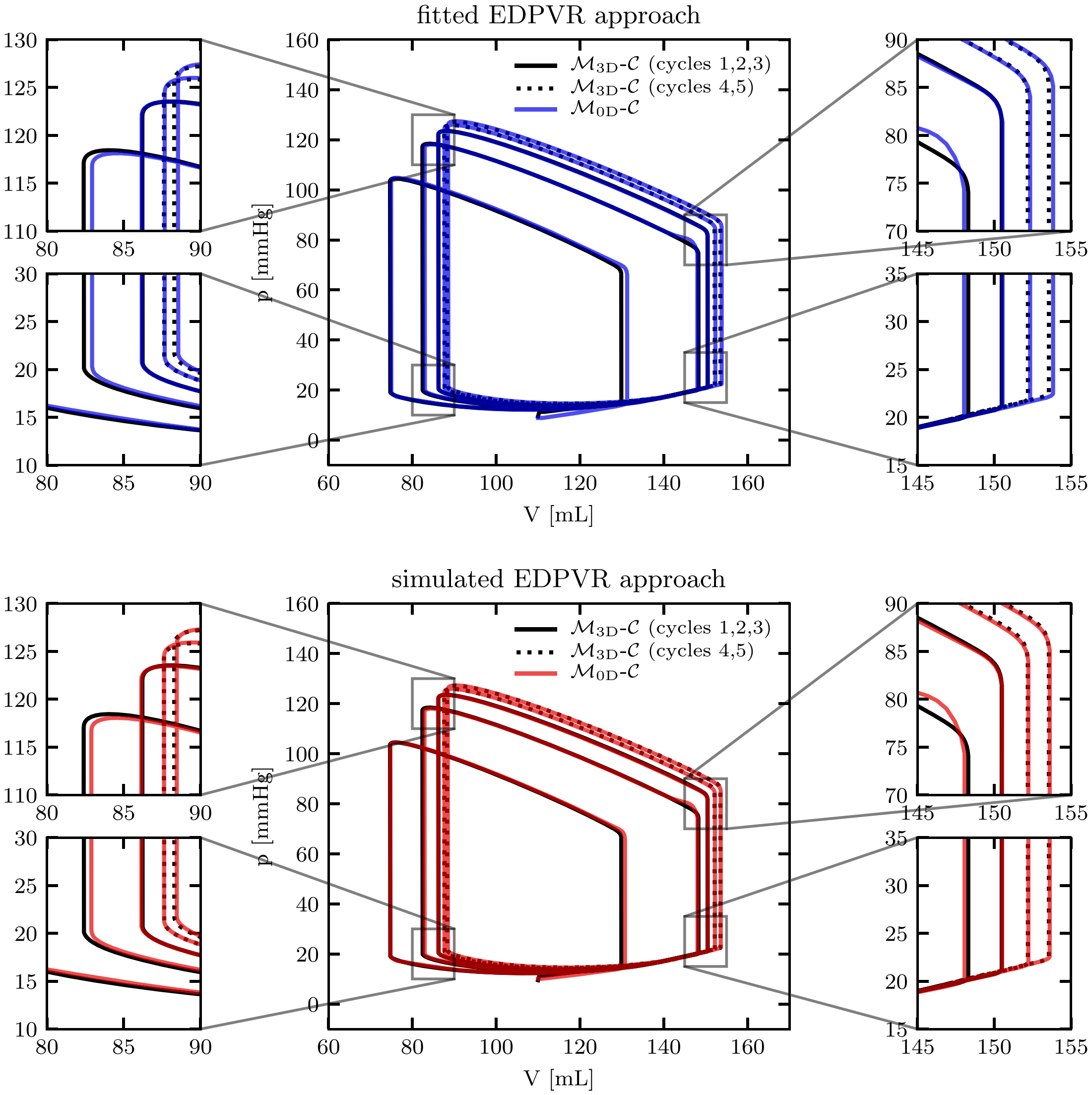

Fig. 8. Comparison of the PV loops obtained with the $\mathscr{M}_{3D} - \mathscr{C}$ model and with the $\mathscr{M}_{0D} - \mathscr{C}$ surrogates. The two rows refer to the emulator built with the *fitted EDPVR* (top) and the *simulated EDPVR* approaches (bottom). The black lines are obtained with the $\mathscr{M}_{3D} - \mathscr{C}$ 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 $\mathscr{M}_{0D} - \mathscr{C}$ 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.

Table 1
Relative mean square errors in the pressure and volume transients obtained by replacing the $\mathscr{M}_{3D} - \mathscr{C}$ model with $\mathscr{M}_{0D} - \mathscr{C}$ surrogates, built with the *fitted EDPVR* and the *simulated EDPVR* approaches. The three columns refer to cycles 1 to 3 (used to construct to emulators), cycles 4 to 5 (not "seen" by the emulator) and finally to all cycles.

	relative error on pressure			**relative error on volume**		
EDPVR	cycles # 1–3	cycles # 4–5	cycles # 1–5	cycles # 1–3	cycles # 4–5	cycles # 1–5
fitted	2.82×10^{-2}	6.86×10^{-3}	2.14×10^{-2}	8.41×10^{-3}	2.93×10^{-3}	6.65×10^{-3}
simulated	2.73×10^{-2}	7.28×10^{-3}	2.08×10^{-2}	8.40×10^{-3}	3.43×10^{-3}	6.74×10^{-3}

Table 2
List of the test cases considered in this paper. The third column indicates whether the test case is performed with the same $\mathscr{C}$ model parameters used to produce the sample PV loops; similarly, the fourth column indicates whether the same $\mathscr{M}_{3D}$ model parameters (specifically, contractility) are used in the sample PV loops and in the test case.

	EDPVR	Same parameters as sample PV loops	
		Circulation model	*Contractility*
Test 1a	fitted	✓	✓
Test 1b	simulated	✓	✓
Test 2a	fitted	✗	✓
Test 2b	simulated	✗	✓
Test 3a	fitted	✓	✗
Test 4a	fitted	✗	✗

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 model at hand, contractility is tuned through a parameter – denoted as a_{XB} in the original paper [29] – that collectively

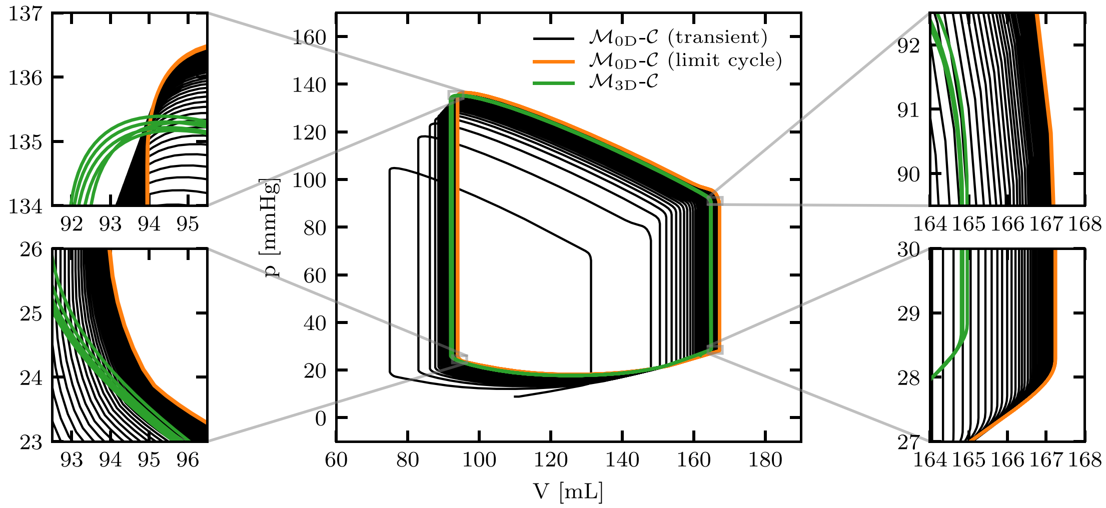

Fig. 9. Test 1a: results of the 3D-0D-3D V-cycle (fitted EDPVR approach).

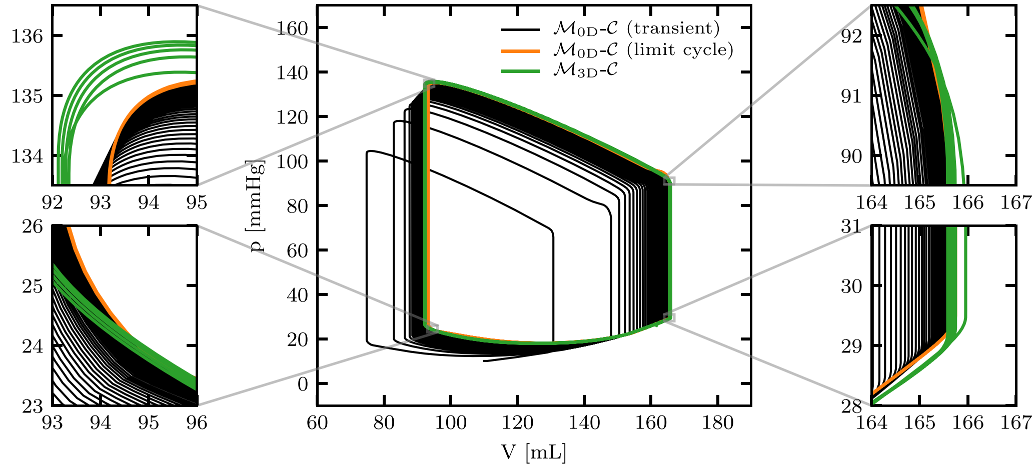

Fig. 10. Test 1 b: results of the 3D-0D-3D V-cycle (simulated EDPVR approach).

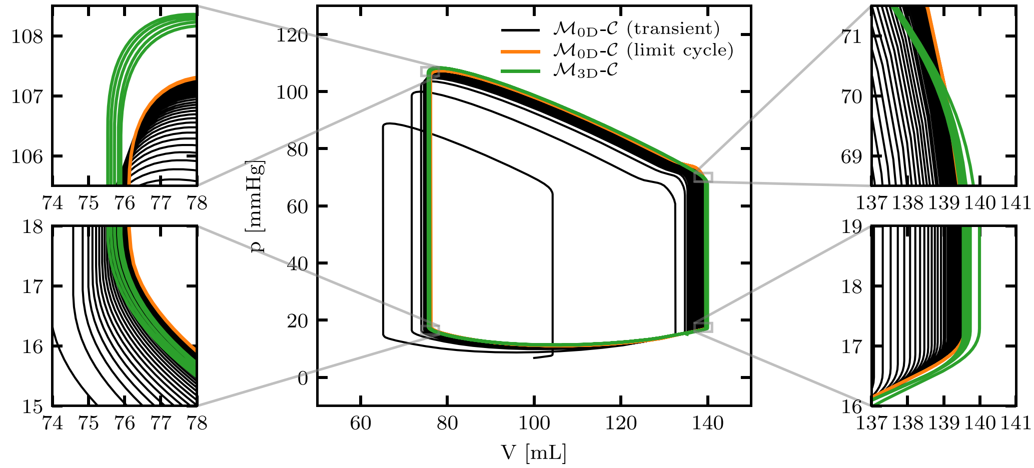

Fig. 11. Test 2a: results of the 3D-0D-3D V-cycle (fitted EDPVR approach).

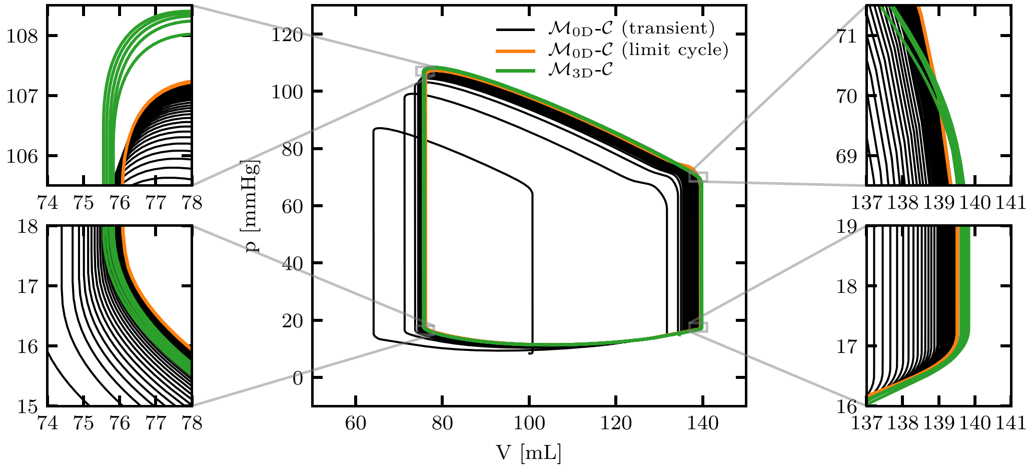

Fig. 12. Test 2 b: results of the 3D-0D-3D V-cycle (simulated EDPVR approach).

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 $\mathscr{M}_{3D} - \mathscr{C}$ model, we obtain three sample PV loops for each of the two settings (step 1). Then, following the algorithm of Sec. 2.1, we build two emulators (one for each contractility value), respectively denoted by $\mathscr{P}_A$ and $\mathscr{P}_B$ (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 $n_{\mathrm{HB}}^{\mathrm{0D}} = 100$ heartbeats through the $\mathscr{M}_{0D} - \mathscr{C}$ model (step 2). Finally, we perform step 3 (by running simulations with the $\mathscr{M}_{3D} - \mathscr{C}$ model, initialized according to the results of the $\mathscr{M}_{0D} - \mathscr{C}$ model). We consider two settings: first, we employ the same $\mathscr{C}$ model parameters as in the sample PV loops (Test 3a); then, we employ different $\mathscr{C}$ model parameters (Test 3 b). The results show that the parametric emulator is effective in predicting the location of the limit cycle of the $\mathscr{M}_{3D} - \mathscr{C}$ 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. For a quantitative assesment of the closeness of these solutions to a limit cycle, we employ the metric e_n defined in Sec. 2.4, that is the normalized difference between consecutive cycles. 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 1, e_n rapidly decreases (orange line). During step 2 (blue line), e_n first decreases following the values of step 1 (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 3. As shown in the figure, thanks to the intermediate simulation performed through the $\mathscr{M}_{0D} - \mathscr{C}$ model, the second of the two simulations obtained with the $\mathscr{M}_{3D} - \mathscr{C}$ model (i.e. step 3) starts much closer to a limit cycle than the first one (step 1). Already the second of the five beats performed in step 3 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 3 is obtained by virtue of three heartbeats simulated during step 1, the heartbeats in step 3 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 3. 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

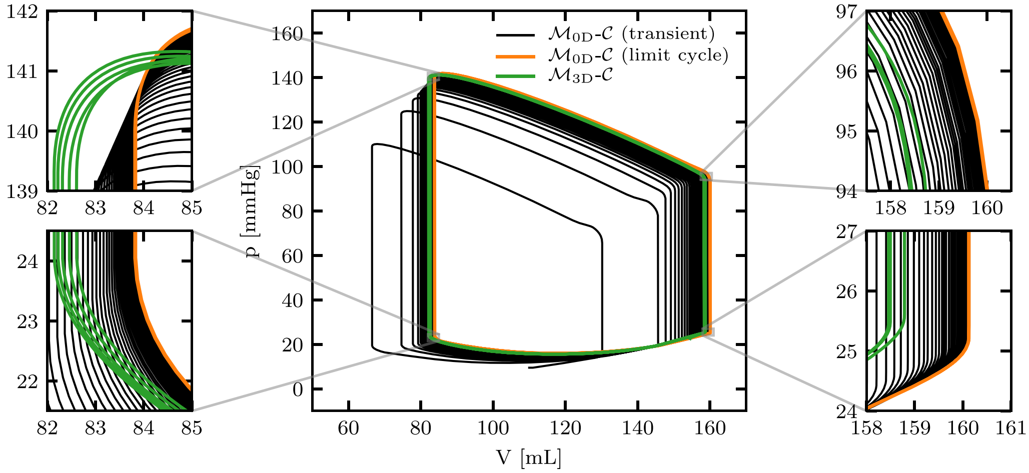

Fig. 13. Test 3a: results of the 3D-0D-3D V-cycle (fitted EDPVR approach).

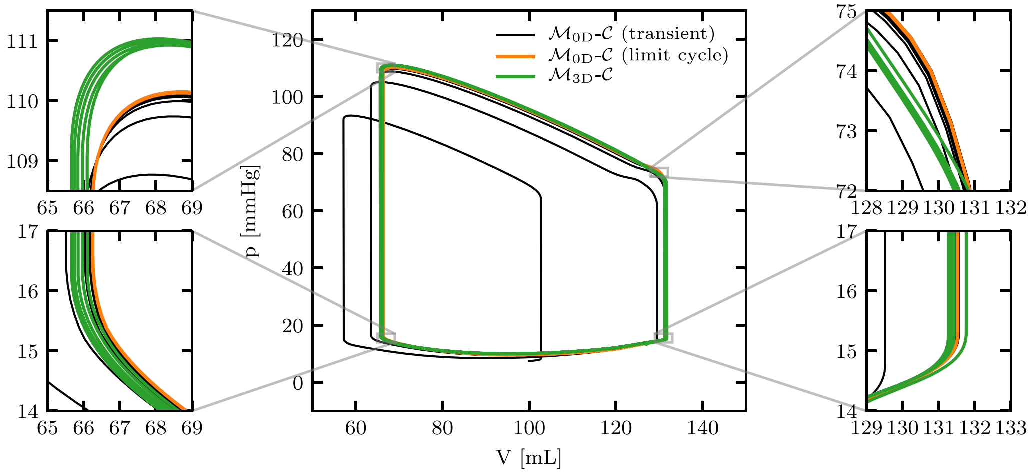

Fig. 14. Test 4a: results of the 3D-0D-3D V-cycle (fitted EDPVR approach).

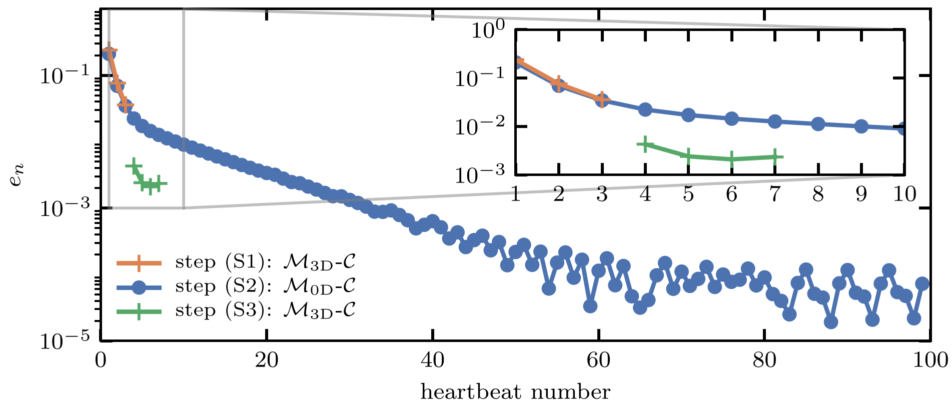

Fig. 15. Trend of the metric e_n in the three steps of the 3D-0D-3V V-cycle in Test 1a (semilogarithmic scale).

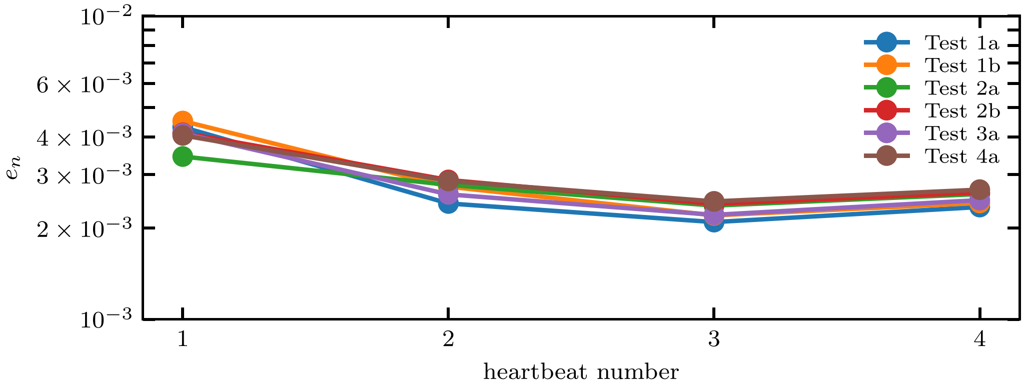

Fig. 16. Trend of the metric e_n in step 3 of the 3D-0D-3V V-cycle in the different test cases (semilogarithmic scale).

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 $2 \cdot 10^{-3} \div 3 \cdot 10^{-3}$.

5. Discussion

In this work, we propose a method to considerably reduce the computational cost required to converge to a limit cycle in cardiac electromechanics simulations. Our method is based on a data-driven 0D emulator, built in a data-driven way from a PV transient obtained with the 3D electromechanical model, by which it is possible to simulate the full transient up to the limit cycle at a very low computational cost. This provides an initial condition for the 3D full-order model, which thus converges to the limit cycle in as few as two cycles.

We remark that other 0D models describing the time-dependent PV relationship of a cardiac chamber have been previously proposed in the literature. This is the case of time-varying elastance models, which assume a PV relationship of the type of $\mathscr{P}(V,t) = E(t)(V - V_0)$, or variations of this, where $E(t)$ is the time-dependent elastance (see e.g. [17,18,26]). The peculiarity of our model is that, while these time-varying elastance models are based on prescribed functions, our 0D emulator is obtained in a data-driven way, i.e. interpolating the outputs of a more complex 3D model. This makes it particularly accurate in replicating the results, thus achieving an excellent fidelity, as demonstrated by our numerical results.

An alternative approach to the one proposed in this paper is replacing the high-fidelity 3D electromechanical model with a physics-based reduced model, built e.g. through projection methods [37–39]. Such a reduced model could allow to simulate the transient up to the limit cycle thus providing an initial condition to the 3D high-fidelity model, in a similar way as done in this paper. The 0D emulator proposed in this paper has however several advantages. First, it is non intrusive in the 3D model, that is it does not require accessing the full-order state of the 3D model; conversely, it only requires pressure and volume recordings, thus allowing it to be easily employed with virtually

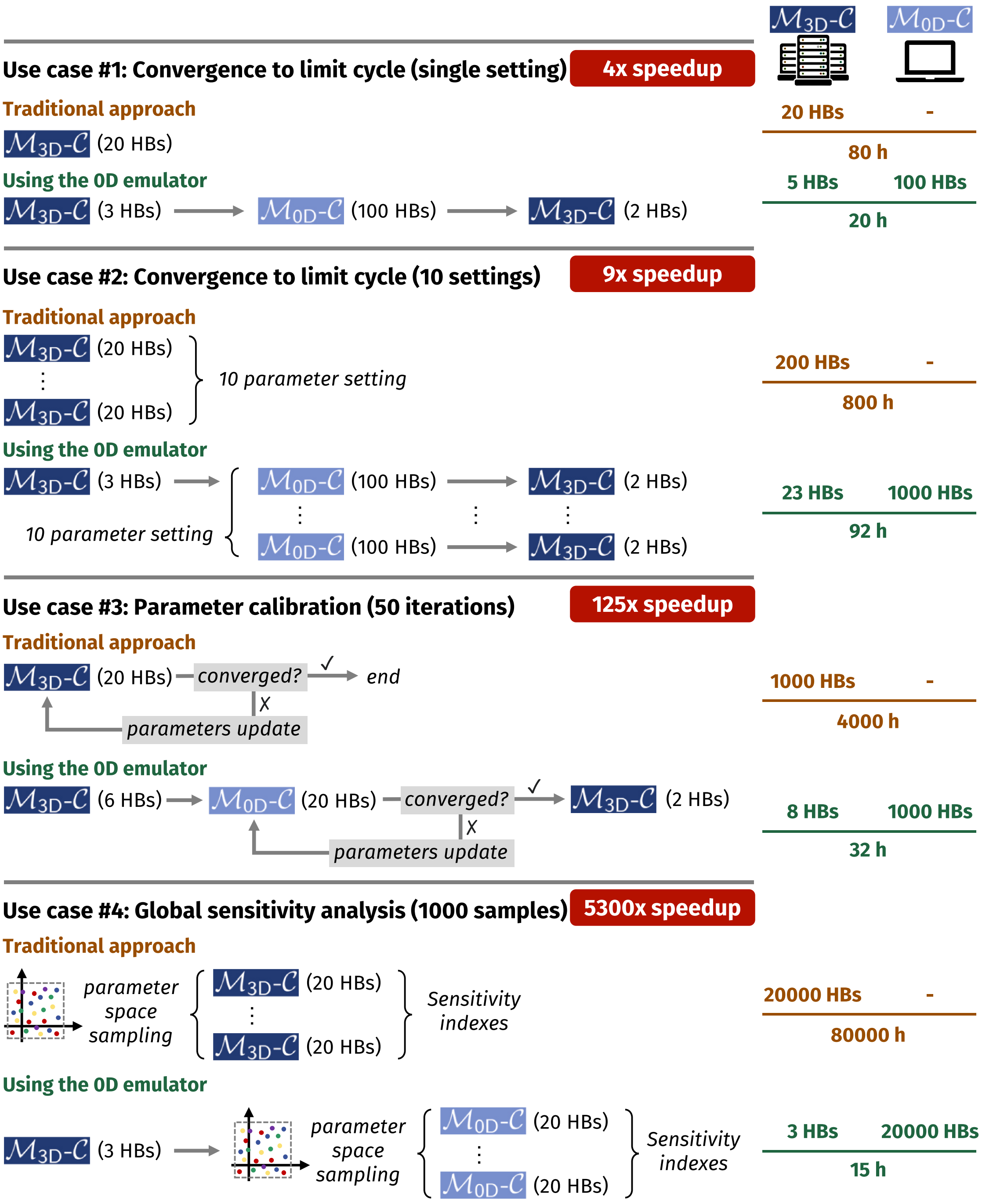
$\mathcal{M}_{3D}$-$\mathcal{C}$
$\mathcal{M}_{0D}$-$\mathcal{C}$
Use case #1: Convergence to limit cycle (single setting)
4x speedup
Traditional approach
$\mathcal{M}_{3D}$-$\mathcal{C}$ (20 HBs)
20 HBs
-
80 h
Using the 0D emulator
$\mathcal{M}_{3D}$-$\mathcal{C}$ (3 HBs)
$\mathcal{M}_{0D}$-$\mathcal{C}$ (100 HBs)
$\mathcal{M}_{3D}$-$\mathcal{C}$ (2 HBs)
5 HBs
100 HBs
20 h
Use case #2: Convergence to limit cycle (10 settings)
9x speedup
Traditional approach
$\mathcal{M}_{3D}$-$\mathcal{C}$ (20 HBs)
⋮
$\mathcal{M}_{3D}$-$\mathcal{C}$ (20 HBs)
10 parameter setting
200 HBs
-
800 h
Using the 0D emulator
$\mathcal{M}_{3D}$-$\mathcal{C}$ (3 HBs)
10 parameter setting
$\mathcal{M}_{0D}$-$\mathcal{C}$ (100 HBs)
⋮
$\mathcal{M}_{0D}$-$\mathcal{C}$ (100 HBs)
$\mathcal{M}_{3D}$-$\mathcal{C}$ (2 HBs)
⋮
$\mathcal{M}_{3D}$-$\mathcal{C}$ (2 HBs)
23 HBs
1000 HBs
92 h
Use case #3: Parameter calibration (50 iterations)
125x speedup
Traditional approach
$\mathcal{M}_{3D}$-$\mathcal{C}$ (20 HBs)
converged?
✓
end
X
parameters update
1000 HBs
-
4000 h
Using the 0D emulator
$\mathcal{M}_{3D}$-$\mathcal{C}$ (6 HBs)
$\mathcal{M}_{0D}$-$\mathcal{C}$ (20 HBs)
converged?
✓
$\mathcal{M}_{3D}$-$\mathcal{C}$ (2 HBs)
X
parameters update
8 HBs
1000 HBs
32 h
Use case #4: Global sensitivity analysis (1000 samples)
5300x speedup
Traditional approach
parameter space sampling
$\mathcal{M}_{3D}$-$\mathcal{C}$ (20 HBs)
⋮
$\mathcal{M}_{3D}$-$\mathcal{C}$ (20 HBs)
Sensitivity indexes
20000 HBs
-
80000 h
Using the 0D emulator
$\mathcal{M}_{3D}$-$\mathcal{C}$ (3 HBs)
parameter space sampling
$\mathcal{M}_{0D}$-$\mathcal{C}$ (20 HBs)
⋮
$\mathcal{M}_{0D}$-$\mathcal{C}$ (20 HBs)
Sensitivity indexes
3 HBs
20000 HBs
15 h

Fig. 17. Summary of four use case scenarios of the proposed 0D emulator. For each use case, the pipeline enabled by the 0D emulator is compared with the traditional one. In the right column, we report the total number of heartbeats (HBs) that need to be simulated either with the $\mathscr{M}_{3D} - \mathscr{C}$ model or with the $\mathscr{M}_{0D} - \mathscr{C}$ one. The total computational times refer to the numerical discretization and the computer platform as described in Sec. 3.2.

any electromechanical model. Moreover, thanks to its lightweight formulation, both the construction of the 0D emulator and its use only require a few seconds of computational time on a standard laptop, whereas the construction of projection-based reduced models is typically more computationally demanding and the provided speedup is also typically lower [39,40]. Clearly, the higher computational cost of projection-based reduced models is repaid by a higher richness of the output produced (using reduced models, it is indeed possible to approximate the entire simulation output, including tissue displacement). Still, for the purpose considered in this work (i.e. convergence to the limit cycle), our 0D emulator provides an optimal trade-off between computational resources and completeness of output.

In recent times, some lightweight emulators of cardiac electromechanical models have been proposed in the literature [41–44]. These are typically based on Machine Learning algorithms (such as Gaussian Process emulators and Artificial Neural Networks), used to approximate the map from a set of parameters to a set of scalar outputs (quantities of interest, QoIs), such as maximum and minimum pressures and volumes. These emulators are however static maps and thus they are not suitable to simulate the transient phase to a limit cycle. As a matter of fact, while these emulators are in fact functions fitting the parameters-to-QoI maps, the emulator we propose in this paper allows to actually *simulate* – albeit in a surrogate form if compared to the 3D electromechanical model – the cardiac cycles.

We remark that our 0D emulator can have other applications besides the one for which it is proposed in this work. In fact, in contexts where the only outputs of interest in the simulation are pressures and volumes, the 0D emulator can entirely replace the 3D electromechanical model. Moreover, the 0D emulator could be built directly from clinical measurements of pressure-volume transients, thus providing a patient-specific data-driven model of the electromechanical function. In these scenarios, thanks to its very low computational cost, the 0D emulator could be employed to simulate the transient phase following changes in the circulatory network (such as those induced by hypertensive or inotropic drugs, or by physical exercise), which would not be feasible with a computationally demanding large-scale 3D electromechanical model. Indeed the 0D emulator, as demonstrated by our numerical results, is able, besides to detect the limit cycle, also to faithfully simulate the transient phase.

For illustrative purposes, we mention four use case scenarios of our 0D emulator. These use cases are summarized in Fig. 17. For each use case we report the computational speedup enabled by the use of the 0D emulator. Clearly, computational times may depend on the $\mathscr{M}_{3D} - \mathscr{C}$ model at hand, on its numerical discretization and on the computer platform used. The figures mentioned below are referred to the models used to produce the numerical results of this paper (see Sec. 3.2).

- Use case #1 corresponds to the 3D-0D-3D V-cycle as presented in Sec. 2.2. The total computational cost amounts to 3 heartbeats performed with the $\mathscr{M}_{3D} - \mathscr{C}$ model, in order to construct the emulator, 100 heartbeats with the $\mathscr{M}_{0D} - \mathscr{C}$ model and finally two additional heartbeats with the $\mathscr{M}_{3D} - \mathscr{C}$ model, for a total of 20 h of computational time. This amount of time should be compared with the computational time needed to reach the limit cycle with a full-order simulation performed with the $\mathscr{M}_{3D} - \mathscr{C}$ model. Since 20 heartbeats are typically needed for this purpose [14], the total computational time, without 0D emulator, would be 80 h. Thus, in use case #1, the use of the 0D emulator provides a 4x speedup.
- In use case #2, we assume that we are interested in simulating several clinical scenarios, corresponding to changes in the hemodynamic conditions of the external circulatory network (e.g. induced by vasodilator drugs or hypertensive pathologies). In this case, since the variations involve only the $\mathscr{C}$ model, it is possible to build a unique emulator, valid for all settings. In case, e.g., 10 different settings need to be simulated, the use of the emulator allows to decrease the computational cost from 800 h to 92 h (9x speedup).
- In use case #3 we consider a parameter calibration scenario for a LV electromechanical model. Specifically, let us assume that we need to calibrate the parameters of the $\mathscr{C}$ model and the active contractility of the $\mathscr{M}_{3D}$ model, in order to personalize the model from clinical measurements of a specific patient's biomarkers. This task can be approached through an optimization problem, in which one iteratively solve the $\mathscr{M}_{3D} - \mathscr{C}$ problem up to the limit cycle, evaluates the mismatch between the measured and predicted biomarkers, and finally updates the parameters according to a suitable algorithm. This cycle is repeated until a convergence criterion in satisfied. The 0D emulator can significantly speedup this process. As a matter of fact, the computationally demanding iterative optimization loop can be performed by fully relying on the $\mathscr{M}_{0D} - \mathscr{C}$ model. The $\mathscr{M}_{3D} - \mathscr{C}$ model is only used to build the emulator (in this case two simulations of three heartbeats duration are required since we need a parametric emulator, according to Sec. 2.3) and to perform two heartbeats with the final parameters setting. This use case can be interpreted as a 3D-0D-3D V-cycle as well, where the parameter calibration iterative loop is performed at step 2, in a fully 0D manner. Assuming that the optimization algorithm requires 50 iterations for convergence, then using the 0D emulator as described above would provide a 125x speedup.
- Finally, as use case #4, we consider a global sensitivity analysis on the influence of external circulation parameters on the LV function. Global sensitivity methods envisage sampling the parameter space with a large number of instances and extracting from the associated $\mathscr{M}_{3D} - \mathscr{C}$ model solutions suitable quantities of interest (i.e. biomarkers). Finally, how much each parameter contributes to a determined quantity of interest is assessed by computing suitable indices, such as Sobol indices [45], Borgonovo indices [46] or Morris elementary effects [47]. In case the quantities of interest only involve pressures and volumes, there is actually no need to perform step 3 of the 3D-0D-3D V-cycle, at the price of a little approximation. In other terms, the 0D emulator, once build, can fully replace the full-order 3D model in this use case. Since global sensitivity analysis may require order of 1000 samples for statistical convergence [45,48], the 0D emulator would allow to carry out a global sensitivity analysis, even if approximate, that would have been otherwise unaffordable. Indeed, on the computer platform considered, the computational time relying on the full order $\mathscr{M}_{3D} - \mathscr{C}$ would be greater than 9 years, while our 0D emulator enables to carry out global sensitivity analysis (including the construction of the emulator) in just 15 h. In this setting, the speedup provided by the 0D emulator would be 5300x.

6. 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 $\mathscr{M}_{3D}$), while keeping the 0D circulation model $\mathscr{C}$, 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 $\mathscr{M}_{3D} - \mathscr{C}$ model that is very close to a periodic solution. In fact, in all the tests considered, by virtue of the simulation performed with the $\mathscr{M}_{0D} - \mathscr{C}$ surrogate, after only two heartbeats the $\mathscr{M}_{3D} - \mathscr{C}$ 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 $\mathscr{M}_{3D} - \mathscr{C}$ 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 1 s to build the emulator, less than 1 min to run 100 heartbeats with the $\mathscr{M}_{0D} - \mathscr{C}$ 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 $\mathscr{M}_{3D} - \mathscr{C}$ 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 $\mathscr{M}_{0D} - \mathscr{C}$ model can entirely replace the $\mathscr{M}_{3D} - \mathscr{C}$ model in such many-query settings. Therefore, we propose our $\mathscr{M}_{0D} - \mathscr{C}$ model as a surrogate for the full-order $\mathscr{M}_{3D} - \mathscr{C}$ model in sensitivity analysis and parameter calibration scenarios.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests.

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 $\mathscr{P}_{ED}$ 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 ($p_{ED,i}$, for $i = 1, \ldots, N_{ED}$), and recording the corresponding volumes ($V_{ED,i}$). 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 $p_{ED,i}$ with the solution obtained with the previous value $p_{ED,i-1}$. 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. ($p_{ED,i}$, $V_{ED,i}$) for $i = 1, \ldots, N_{ED}$). Finally, we define the function $p = \mathscr{P}_{ED}(V)$ as a piecewise linear interpolation of these points. Outside the interval $[V_{ED,1}, V_{ED,N_{ED}}]$, we define $\mathscr{P}_{ED}$ 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 [23]. 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 ($p_{ED,i}$, $V_{ED,i}$) for $i = 1, \ldots, N_{ED}$. These points are then fitted with the Klotz curve

$$\mathscr{P}_{ED}(V) = A_n \left(\frac{V - V_0^{ED}}{V_{30}^{ED} - V_0^{ED}} \right)^{B_n}, \tag{7}$$

where, according to [23], we set the universal constants $A_n = 28.2$ 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 ($p_{ED,i}$, $V_{ED,i}$), via the Levenberg-Marquardt optimization method.

A.2 End-systolic pressure-volume relationship

For the ESPVR we assume the linear relationship

$$\mathscr{P}_{ES}(V) = E^{ES}(V - V_0^{ES}), \tag{8}$$

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

n-th cycle. For each heartbeat n, we select 10 points as

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

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

$$\{(p_{\mathrm{ES},i}, V_{\mathrm{ES},i}) : 1 \le i \le N_{\mathrm{ES}}\} := \left\{ \left(p(t_j^n), V(t_j^n) \right) : 1 \le n \le n_{\mathrm{HB}}, 1 \le j \le 10 \right\}$$

A.3 Activation kinetics function

From Eq. (1) it follows

$$\varphi_{\mathrm{act}}(t) = \frac{\mathscr{P}(V,t) - \mathscr{P}_{\mathrm{ED}}(V)}{\mathscr{P}_{\mathrm{ES}}(V) - \mathscr{P}_{\mathrm{ED}}(V)}. \tag{9}$$

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

$$\widetilde{\varphi}_{\mathrm{act}}(t_i) = \frac{p_i - \mathscr{P}_{\mathrm{ED}}(V_i)}{\mathscr{P}_{\mathrm{ES}}(V_i) - \mathscr{P}_{\mathrm{ED}}(V_i)}, \qquad \text{for} \quad i = 1, \dots, N_{\mathrm{PV}}, \tag{10}$$

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

$$\varphi_{\mathrm{act}}(t) = \widetilde{\varphi}_{\mathrm{act}}((n_{\mathrm{HB}} - 1)T_{\mathrm{HB}} + \bmod(t, T_{\mathrm{HB}})) \tag{11}$$

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

References

[1] J.M. Guccione, K.D. Costa, A.D. McCulloch, Finite element stress analysis of left ventricular mechanics in the beating dog heart, J. Biomech. 28 (10) (1995) 1167–1177.

[2] N. Smith, D. Nickerson, E. Crampin, P. Hunter, Multiscale computational modelling of the heart, Acta Numer. 13 (2004) 371–431.

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

[4] E.J. Vigmond, C. Clements, D.M. McQueen, C.S. Peskin, Effect of bundle branch block on cardiac output: a whole heart simulation study, Prog. Biophys. Mol. Biol. 97 (2–3) (2008) 520–542.

[5] M. Boulakia, S. Cazeau, M.A. Fernández, J.-F. Gerbeau, N. Zemzemi, Mathematical modeling of electrocardiograms: a numerical study, Ann. Biomed. Eng. 38 (3) (2010) 1071–1097.

[6] N.A. Trayanova, Whole-heart modeling applications to cardiac electrophysiology and electromechanics, Circ. Res. 108 (2011) 113–128.

[7] M. Fink, S. Niederer, E. Cherry, F. Fenton, J. Koivumäki, G. Seemann, R. Thul, H. Zhang, F. Sachse, D. Beard, E. Crampin, N. Smith, Cardiac cell modelling: observations from the heart of the cardiac physiome project, Prog. Biophys. Mol. Biol. 104 (1) (2011) 2–21.

[8] S. Sugiura, T. Washio, A. Hatano, J. Okada, H. Watanabe, T. Hisada, Multi-scale simulations of cardiac electrophysiology and mechanics using the university of tokyo heart simulator, Prog. Biophys. Mol. Biol. 110 (2) (2012) 380–389.

[9] B. Baillargeon, N. Rebelo, D.D. Fox, R.L. Taylor, E. Kuhl, The living heart project: a robust and integrative simulator for human heart function, Eur. J. Mech. Solid. 48 (2014) 38–47.

[10] R. Chabiniok, V. Wang, M. Hadjicharalambous, L. Asner, J. Lee, M. Sermesant, E. Kuhl, A. Young, P. Moireau, M. Nash, D. Chapelle, D. Nordsletten, Multiphysics and multiscale modelling, data–model fusion and integration of organ physiology in the clinic: ventricular cardiac mechanics, Interface Focus 6 (2) (2016) 20150083.

[11] A. Quarteroni, L. Dede', A. Manzoni, 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.

[12] T. Washio, K. Yoneda, J. Okada, T. Kariya, S. Sugiura, T. Hisada, Ventricular fiber optimization utilizing the branching structure, International Journal for Numerical Methods in Biomedical Engineering (2015), e02753.

[13] T. Gerach, S. Schuler, J. Fröhlich, L. Lindner, E. Kovacheva, R. Moss, E.M. Wülfers, G. Seemann, C. Wieners, A. Loewe, Electro-mechanical whole-heart digital twins: a fully coupled multi-physics approach, Mathematics 9 (11) (2021).

[14] C.M. Augustin, M.A. Gsell, E. Karabelas, G. Plank, Physiologically Valid 3d-0d Closed Loop Model of the Heart and Circulation–Modeling the Acute Response to Altered Loading and Contractility, 2020, 08802 arXiv preprint arXiv:2009.

[15] A. Gerbi, L. Dedè, A. Quarteroni, A monolithic algorithm for the simulation of cardiac electromechanics in the human left ventricle, Mathematics in Engineering 1 (1) (2018) 1–37.

[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, Comput. Methods Appl. Mech. Eng. 361 (2020) 112762.

[17] M. Hirschvogel, M. Bassilious, L. Jagschies, 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, International Journal for Numerical Methods in Biomedical Engineering 33 (8) (2017), e2842.

[18] F. Regazzoni, M. Salvador, P.C. Africa, M. Fedele, L. Dede', A. Quarteroni, A Cardiac Electromechanics Model Coupled with a Lumped Parameters Model for Closed-Loop Blood Circulation. Part I: Model Derivation, 2020, p. 15040, arXiv preprint arXiv:2011.

[19] J. Tomek, A. Bueno-Orovio, E. Passini, X. Zhou, A. Minchole, O. Britton, C. Bartolucci, S. Severi, A. Shrier, L. Virag, et al., Development, calibration, and validation of a novel human ventricular myocyte model in health, disease, and drug block, Elife 8 (2019), e48890.

[20] J. Lumens, T. Delhaas, B. Kirn, T. Arts, Three-wall segment (triseg) model describing mechanics and hemodynamics of ventricular interaction, Ann. Biomed. Eng. 37 (11) (2009) 2234–2255.

[21] E. Lluch, O. Camara, R. Doste, B. Bijnens, M. De Craene, M. Sermesant, V.Y. Wang, M.P. Nash, H.G. Morales, Calibration of a fully coupled electromechanical meshless computational model of the heart with experimental data, Comput. Methods Appl. Mech. Eng. 364 (2020) 112869.

[22] J. Xu, Theory of Multilevel Methods, 8924558, Cornell University Ithaca, NY, 1989.

[23] S. Klotz, I. Hay, M.L. Dickstein, G.-H. Yi, J. Wang, M.S. Maurer, D.A. Kass, D. Burkhoff, Single-beat estimation of end-diastolic pressure-volume relationship: a novel method with potential for noninvasive application, Am. J. Physiol. Heart Circ. Physiol. 291 (1) (2006). H403–H412.

[24] I. Mirsky, Left ventricular stresses in the intact human heart, Biophys. J. 9 (2) (1969) 189–208.

[25] T. Arts, T. Delhaas, P. Bovendeerd, X. Verbeek, F.W. Prinzen, Adaptation to mechanical load determines shape and properties of heart and circulation: the circadapt model, Am. J. Physiol. Heart Circ. Physiol. 288 (4) (2005) H1943–H1954.

[26] P.J. Blanco, R.A. Feijóo, A 3d-1d-0d computational model for the entire cardiovascular system, Comput. Mech. 24 (2010) 5887–5911.

[27] P. Colli Franzone, L. Pavarino, S. Scacchi, Mathematical Cardiac Electrophysiology, 13, Springer, 2014.

[28] K.H. Ten Tusscher, A.V. Panfilov, Alternans and spiral breakup in a human ventricular tissue model, Am. J. Physiol. Heart Circ. Physiol. 291 (3) (2006) H1088–H1100.

[29] F. Regazzoni, L. Dedè, A. Quarteroni, Biophysically detailed mathematical models of multiscale cardiac active mechanics, PLoS Comput. Biol. 16 (10) (2020), e1008294.

[30] T.P. Usyk, I.J. LeGrice, A.D. McCulloch, Computational model of three-dimensional cardiac electromechanics, Comput. Visual Sci. 4 (4) (2002) 249–257.

[31] F. Regazzoni, L. Dedè, A. Quarteroni, Machine learning of multiscale active force generation models for the efficient simulation of cardiac electromechanics, Comput. Methods Appl. Mech. Eng. 370 (2020) 113268.
[32] Zygote 3d models. https://www.zygote.com/, 2019.
[33] J.D. Bayer, R.C. Blake, G. Plank, N.A. Trayanova, A novel rule-based algorithm for assigning myocardial fiber orientation to computational heart models, Ann. Biomed. Eng. 40 (10) (2012) 2243–2254.
[34] R. Piersanti, P.C. Africa, M. Fedele, C. Vergara, L. Dede', A.F. Corno, A. Quarteroni, Modeling cardiac muscle fibers in ventricular and atrial electrophysiology simulations, Comput. Methods Appl. Mech. Eng. 373 (2021) 113468.
[35] F. Regazzoni, M. Salvador, P.C. Africa, M. Fedele, L. Dede', A. Quarteroni, A Cardiac Electromechanics Model Coupled with a Lumped Parameters Model for Closed-Loop Blood Circulation. Part II: Numerical Approximation, 2020, p. 15051, arXiv preprint arXiv:2011.
[36] F. Regazzoni, A. Quarteroni, An oscillation-free fully partitioned scheme for the numerical modeling of cardiac active mechanics, Comput. Methods Appl. Mech. Eng. 373 (2021) 113506.
[37] A. Quarteroni, A. Manzoni, F. Negri, Reduced Basis Methods for Partial Differential Equations. An Introduction, 92, Springer, 2016.
[38] J. Hesthaven, G. Rozza, B. Stamm, Certified Reduced Basis Methods for Parametrized Partial Differential Equations, Springer, 2016.
[39] S. Pagani, A. Manzoni, Enabling forward uncertainty quantification and sensitivity analysis in cardiac electrophysiology by reduced order modeling and machine learning, International Journal for Numerical Methods in Biomedical Engineering (2021), e3450.
[40] D. Bonomi, A. Manzoni, A. Quarteroni, A matrix deim technique for model reduction of nonlinear parametrized problems in cardiac mechanics, Comput. Methods Appl. Mech. Eng. 324 (2017) 300–326.
[41] P. Di Achille, A. Harouni, S. Khamzin, O. Solovyova, J.J. Rice, V. Gurev, Gaussian process regressions for inverse problems and parameter searches in models of ventricular mechanics, Front. Physiol. 9 (2018) 1002.
[42] Y. Dabiri, A. Van der Velden, K.L. Sack, J.S. Choy, G.S. Kassab, J.M. Guccione, Prediction of left ventricular mechanics using machine learning, Frontiers in physics 7 (2019) 117.
[43] S. Longobardi, A. Lewalle, S. Coveney, I. Sjaastad, E.K.S. Espe, W.E. Louch, C. J. Musante, A. Sher, S.A. Niederer, Predicting left ventricular contractile function via Gaussian process emulation in aortic-banded rats, Phil. Trans. Math. Phys. Eng. Sci. 378 (2173) (2020) 20190334.
[44] L. Cai, L. Ren, Y. Wang, W. Xie, G. Zhu, H. Gao, Surrogate models based on machine learning methods for parameter estimation of left ventricular myocardium, Royal Society Open Science 8 (1) (2021) 201121.
[45] I.M. Sobol', On sensitivity estimation for nonlinear mathematical models, Matematicheskoe modelirovanie 2 (1) (1990) 112–118.
[46] E. Plischke, E. Borgonovo, C.L. Smith, Global sensitivity measures from given data, Eur. J. Oper. Res. 226 (3) (2013) 536–550.
[47] M.D. Morris, Factorial sampling plans for preliminary computational experiments, Technometrics 33 (2) (1991) 161–174.
[48] T. Homma, A. Saltelli, Importance measures in global sensitivity analysis of nonlinear models, Reliab. Eng. Syst. Saf. 52 (1) (1996) 1–17.