


Multifidelity-CMA: a multifidelity approach for efficient personalisation of 3D cardiac electromechanical models

Roch Molléro¹  · Xavier Pennec¹ · Hervé Delingette¹ · Alan Garny² · Nicholas Ayache¹ · Maxime Sermesant¹

Received: 31 December 2016 / Accepted: 31 August 2017 / Published online: 11 September 2017
© Springer-Verlag GmbH Germany 2017

Abstract Personalised computational models of the heart are of increasing interest for clinical applications due to their discriminative and predictive abilities. However, the simulation of a single heartbeat with a 3D cardiac electromechanical model can be long and computationally expensive, which makes some practical applications, such as the estimation of model parameters from clinical data (the personalisation), very slow. Here we introduce an original multifidelity approach between a 3D cardiac model and a simplified “0D” version of this model, which enables to get reliable (and extremely fast) approximations of the global behaviour of the 3D model using 0D simulations. We then use this multifidelity approximation to speed-up an efficient parameter estimation algorithm, leading to a fast and computationally efficient personalisation method of the 3D model. In particular, we show results on a cohort of 121 different heart geometries and measurements. Finally, an exploitable code of the 0D model with scripts to perform parameter estimation will be released to the community.

Keywords Cardiac electromechanical modelling · Reduced model · Multifidelity modelling · Parameter estimation · Finite element mechanical modelling

1 Introduction

Electromechanical models of the heart simulate the physical behaviour of a patient’s heart, in order to perform advanced analysis of the cardiac function. They are of increasing interest to help clinicians in their daily practice (Kayvanpour et al. 2015; Baillargeon et al. 2014; Smith et al. 2011). In particular, recent works have been successful in predicting haemodynamic changes in cardiac resynchronisation therapy (Sermesant et al. 2012), ventricular tachycardia inducibility and dynamics (Chen et al. 2016), as well as in detecting and localising infarcts (Duchateau et al. 2016) using 3D personalised models.

After building the patient’s heart mesh geometry, the simulated heartbeat has to match clinical data, such as ejected blood volume and pressure measurements, or more detailed information about regional motion and abnormalities available from imaging modalities such as 3D echocardiography or cine MRI. This is done by finding adequate simulations settings (boundary conditions, loading constraints) and values of model parameters such as myocardial stiffness and contractility (Xi et al. 2011; Chabiniok et al. 2012). This phase of parameter estimation is usually referred to as the *personalisation* of the cardiac model (Marchesseau et al. 2013b) and results in a *personalised cardiac model* (Wang et al. 2012) made of a *patient-specific heart geometry* (Schaerer et al. 2006) and *patient-specific biomechanical parameters*.

A wide variety of 3D computer heart models exists in the literature, which describe the anatomy and physiology of the heart at various scales. For example, the 3D mesh describing the heart geometry can be made of very different numbers of nodes, and the cellular electromechanical phenomena underlying the build-up of myocardial forces can either be described with a large number of equations, or simplified equations. We refer to the two comprehensive reviews

✉ Roch Molléro
rochmollero@hotmail.com
Maxime Sermesant
maxime.sermesant@inria.fr

¹ Inria, Asclepios Research Project, Sophia Antipolis, France
² Auckland Bioengineering Institute, University of Auckland, Auckland, New Zealand

of Chabiniok et al. (2016) and Clayton et al. (2011) for a large discussion of various models of different scales, types and implementations. The scale and precision (also known as fidelity) of the model is chosen according to the study and the available data. In general, the time required to compute a simulation increases with its level of detail. The simulation of a 3D heartbeat using some of the most complex 3D models can take up to several hours of computation on computers with hundreds of cores (Panthee et al. 2016). This means that for applications where many simulations need to be repeatedly performed (e.g. parameter estimation), computational time becomes a real issue.

The joint use of low-fidelity models to approximate a high-fidelity model and lower the computational burden has been investigated by the *multifidelity modelling* community since (Kennedy and O'Hagan 2000). As described in Peherstorfer et al. (2016), a *model management method* usually handles and feeds the outputs of a *low-fidelity model* (e.g. a simplified model, a regression model or a projection-based model) to an underlying *application-specific method* (e.g. an optimisation algorithm) as surrogates to the high-fidelity model outputs. The method also optionally decides when to recompute simulations of the high-fidelity model to guarantee the accuracy of the low-fidelity approximation.

Here we present an original 0D/3D multifidelity approach for the personalisation of 3D cardiac models (Fig. 1). First, from our 3D cardiac model, we derived and implemented a simplified “0D” model which is faster by 4 orders of magnitude. This was performed as proposed in Caruel et al. (2014), by approximating the geometry of the ventricle as a sphere and assuming spherical symmetry and homogeneity of the electromechanical behaviour.

Then, we introduce a *multifidelity coupling* in order to approximate 3D model simulations from 0D model simulations. To this end, we build a parameter mapping which converts parameters of the 3D model into parameters of the 0D model, based on a few representative 3D simulations in the parameter space (called the *sigma-simulations*). Outputs of the 3D model are then approximated from 0D model simulations, thus enabling a reduction in the computational burden when a large number of 3D simulations outputs are required.

Finally, we present a *multifidelity personalisation method*, built by adapting an efficient optimisation algorithm called CMA-ES (Hansen 2006) to use approximations of the 3D simulations obtained through the *multifidelity coupling* instead of the real 3D simulations. This leads to a fast and computationally efficient personalisation method for the 3D model parameters.

A preliminary version of this work was described in Mollero et al. (2016). In this manuscript, we propose a significantly extended methodology for the *multifidelity coupling*. First, the *sigma-simulations* selection is performed so that additional computational gains are possible when some esti-

mated parameters have the same equations and values in both models. Then, a more robust, nonlinear, parameter mapping is used. An additional step is finally introduced to correct the possible errors arising during the estimation of 0D model parameters. We also present an improved methodology for the *multifidelity personalisation method* which enables the use of a single coupling for many iterations of CMA-ES. This is done while simultaneously ensuring that the approximation is accurate enough for the optimisation algorithm, resulting in an overall estimation which is 5 times faster than in Mollero et al. (2016) on average.

In terms of results, we present extended results and discussions for both the *multifidelity coupling* and the *multifidelity personalisation method*. The approximation accuracy of the coupling is compared to an hypersurface interpolation method, and the personalisation method is compared to BOBYQA (Powell 2009), a commonly used derivative-free optimisation algorithm. This leads to an extended discussion on the computational aspects of our method in a parallel environment. This work is illustrated on a personalisation problem involving 5 parameters and 3 outputs, and we demonstrate results on a database of 121 *different geometries and clinical values*, which we believe to be one of the largest cohort of personalised cardiac cases to date. This personalisation took around 2.5 days on our cluster.

Lastly, our 0D model equations are encoded in the CellML format (Cuellar et al. 2003) and made available for download from the Physiome Model Repository¹ (Yu et al. 2011). Python scripts to perform parameter estimation in the 0D model will be released within 1 month of publication, from the same location.

2 Multifidelity cardiac modelling and personalisation framework

In this work, we use both a 3D electromechanical model which can simulate the behaviour of complex patient-specific heart geometries and a reduced “0D” version of this model which can be summarised in a few equations. Both models rely on the same mechanical laws, but simplifying assumptions are made on the geometry of the 0D model to derive its equations. We also introduce the *personalisation framework* for the parameters of both models.

2.1 The 3D cardiac model

Our 3D cardiac electromechanical model is an implementation of the Bestel–Clement–Sorine (BCS) model (Chapelle et al. 2012) by Marchesseau et al. (2010, 2013a) in SOFA²,

¹ <https://models.physiomeproject.org/e/470>.

² <https://www.sofa-framework.org>.

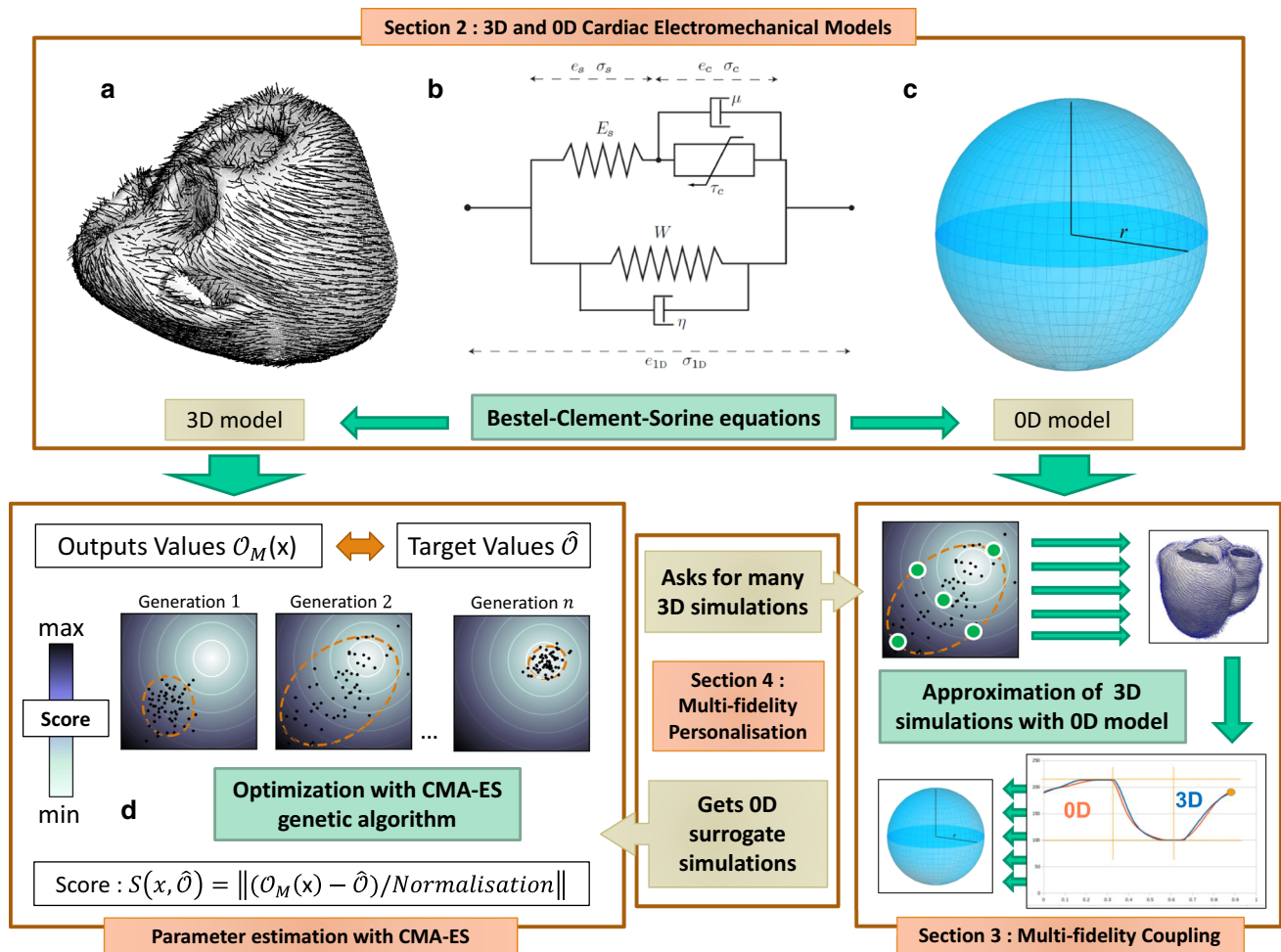


Fig. 1 3D and 0D cardiac models (a–c). Our *multifidelity personalisation method* performs *parameter estimation* in the 3D model using CMA-ES (d), based on 0D simulations obtained through the *multifidelity coupling* between the models

an open-source simulation software. The model uses the following items as an input:

- A 3D tetrahedral biventricular mesh, either synthetically created or derived from segmented MRI images.
- A set of myocardial fibres directions, defined at each node of the mesh. Here we use synthetic fibres from the rule based on [Streeter \(1979\)](#).
- A set of depolarisation and repolarisation times at each node of the set computed from an electrophysiology model. Here we use the Eikonal model as described in [Sermesant et al. \(2012\)](#).

Myocardial forces are then computed at each node and at each time step from the equations of the BCS model. Then, the myocardial motion (mesh nodes velocities) as well as ventricular volumes and pressures is computed at each time step of the cardiac cycle from these forces. See “Appendix A” for a description of the mechanical model equations and parameters. With myocardial meshes made of around 15 000

nodes and a time step of 5 ms, a single beat of 0.9s takes 15 minutes to compute on average on a single-core (Intel(R) Core(TM) i7-4600U [2.10GHz]).

2.2 The 0D cardiac model

As described in [Caruel et al. \(2014\)](#), it is possible to derive the equations of a fast *0D model* of the heart, which relies on the same BCS equations. This is done by making the following simplifying assumptions on the geometry, the electrical activation and the properties of the material:

1. The ventricle has a spherical shape.
2. The material is incompressible.
3. The electrical activity is synchronous and homogeneous over the sphere.

With these assumptions of spherical symmetry, myocardial forces and motion are also spherically symmetric and can be entirely described by the inner radius r of the ventri-

cle. Deformation and stress tensors can also be reduced to a simple form (see Caruel et al. (2014), which leads to a system of a dozen equations (see “Appendix B”).

We implemented the equations into C code and solve the system of equations using an explicit *Forward Euler* method with a temporal discretisation of 0.01 ms. This leads to the simulation of around 15 beats per second. We also encoded the 0D model in the CellML format (Cuellar et al. 2003), which is an open standard based on the XML markup language to store and exchange computer-based mathematical models. This model can be downloaded from the Physiome Model Repository³ and easily exploited through the software OpenCOR (Garny and Hunter 2015).

2.3 Parameter estimation framework for cardiac models

After building the model, parameter estimation is usually the first step to analyse clinical data with a model. It consists in finding parameter values for which the simulation with the model reproduces available values and quantities in the data, such as pressure or volume measurements. In particular, when the geometry is patient-specific, this phase is called cardiac model *personalisation* (Marchesseau et al. 2013b; Kayvanpour et al. 2015).

Formally, we consider a cardiac model M , a set of simulated quantities called the *outputs* \mathcal{O} and a subset \mathcal{P}_M of varying parameters of the model (while the other parameters are supposed fixed). Given a vector of these parameters $x \in \Omega_M$, we note $\mathcal{O}_M(x)$ the values of the outputs \mathcal{O} in the simulation of M with parameter values x . The goal of personalisation is to find parameter values $x^* \in \Omega_M$ for which the outputs values $\mathcal{O}_M(x^*)$ best match some target values $\widehat{\mathcal{O}}$.

This is an inverse problem, which can be tackled by different methods (see the review of Chabiniok et al. (2016)). We propose here a parameter estimation framework (Fig. 1d) through derivative-free optimisation, using an efficient genetic algorithm called CMA-ES (Hansen 2006).

2.3.1 Robust optimisation with the genetic algorithm

CMA-ES

We define the *score* $S(x, \widehat{\mathcal{O}})$ of some parameter values x as the L_2 distance between $\mathcal{O}_M(x)$ and $\widehat{\mathcal{O}}$, normalised by the Hadamard (coordinate-by-coordinate) division \oslash with a vector \mathcal{N} , in order to compare outputs with different units:

$$S(x, \widehat{\mathcal{O}}) = \|(\mathcal{O}_M(x) - \widehat{\mathcal{O}}) \oslash \mathcal{N}\|.$$

We then perform a derivative-free optimisation with the genetic algorithm *CMA-ES*, which aims at minimising this score S . The algorithm (which stands for covariance matrix

adaptation evolution strategy) asks at each iteration n for the scores of m points $x_i \in \Omega_M$ (a *generation*), drawn from a multivariate distribution with covariance I_n^c and mean I_n^m . Then, it combines Bayesian principles of *maximum likelihood* with *natural gradient descent* on the ranks of the points scores in the generation to update both I_n^c and I_n^m .

The CMA-ES algorithm has many advantages in this context. First, it can explore a large and unbounded parameter space while still performing a local search at each iteration and has shown very good results on problems involving hundreds of parameters to optimise (Geijtenbeek et al. 2013). Second, because the updates of the I_n^c and I_n^m only depend on the score ranks, it is very robust to outliers in the generation, in particular to parameter values for which the simulation diverges (in which case we give an arbitrary high score to these parameters).

Also, since each score comes from an independent simulation, this algorithm is well suited to parallel environments. We can either decide to set a very high population size m and do many parallel simulations (in this case the algorithm can converge in a few iterations), or a lower population size and rely instead on many iterations of the algorithm for convergence.

2.3.2 Application to the 0D model

Because the 0D model is extremely fast (15 beats per seconds), parameter estimation is also very fast with the 0D model. For example, with a population size of 50 points per generation it takes less than 50 generations and 3 minutes on a 4-core computer (with parallel computation of the simulations within each generation) to make most of the problems with sets of up to 10 outputs and parameters converge.

In our current implementation, 29 outputs can be extracted from the pressure, volume and flow curves and 25 parameters of the 0D model can be estimated. Python scripts to automatically perform the parameter estimation will be released within 1 month after the publication, available for download from the Physiome Model Repository.⁴

2.3.3 Application to the 3D model

It is possible to apply directly this framework to the 3D model, but the computational burden can become an issue because of the time required to compute the 3D simulations. Indeed, either we set a small population size, but we need many iterations of CMA-ES (of around 15 minutes each). Or we set a high population size but is the number of parallel CPUs used at the same time which may become prohibitive. In Sect. 4, our *multifidelity personalisation method* lowers this computational burden by replacing the outputs values

³ <https://models.physiomeproject.org/e/470>.

⁴ <https://models.physiomeproject.org/e/470>.

Table 1 Sets of 3D model parameters and global outputs in the example

Global outputs \mathcal{O}
Stroke volume SV
Mean aortic pressure MP
Diastolic aortic pressure DP
Parameters of the 3D model \mathcal{P}
Contractility σ
Stiffness c_1
Peripheral resistance R_p
Aortic compliance C
Central venous pressure P_{ve}

of 3D simulations with approximations computed from 0D simulations through a *multifidelity coupling* between the two models, as explained in Sect. 3.

3 Multifidelity coupling: approximating global outputs values of the 3D model

We present here a *multifidelity coupling* between the 3D and the 0D model. We will call *global outputs* of the models quantities which can be computed from the simulations of both models, such as the total ejected blood volume (stroke volume) or the minimal (diastolic) aortic pressure.

We consider a set of N_1 parameters of the 3D model \mathcal{P}_{3D} , a set of global outputs \mathcal{O} , and a set of parameter values $x_i \in \Omega_{3D}$ of the parameters \mathcal{P}_{3D} . The goal is to get approximations of the values $\mathcal{O}_{3D}(x_i)$ by performing 0D simulations and only a few 3D simulations.

We will illustrate the method on the following problem: a set of 5 parameters \mathcal{P}_{3D} of the 3D model, and a set of 3 outputs \mathcal{O} listed in Table 1. We want to approximate the output values for $m = 30$ simulations with parameters x_i , drawn from a multivariate distribution (as in a CMA-ES iteration).

3.1 Global strategy: building a mapping between 3D and 0D mechanical parameters

Because they rely on the same equations, both models display many similar trends in their *global outputs* values when some parameters vary. For example, if a 3D simulation and a 0D simulation have the same stroke volume, the stroke volume variations with changes in the haemodynamic parameters (such as the peripheral resistance) are very similar in both models.

However, some parameters do not behave exactly the same, and are not always even in the same range of values. This is especially the case of **mechanical parameters** such

as the 3D and 0D dampings which rely on different equations. But even for parameters from the same equations in both models (such as σ and c_1), the values might be very different in 0D and 3D simulations with similar outputs, due to the different assumptions.

Formally, there is no trivial function which can convert the $x_i \in \Omega_{3D}$ into values $y \in \Omega_{0D}$ of 0D model parameters \mathcal{P}_{0D} , for which the global outputs values $\mathcal{O}_{0D}(y)$ and $\mathcal{O}_{3D}(x)$ are the same (or at least close). The idea of the *multifidelity coupling* is to find 0D model simulations which are similar to a few selected 3D simulations and then **build a parameter mapping ϕ between the parameters of both models**. We use the following strategy:

1. First we perform a few representative 3D simulations within the domain of interest (called the *sigma-simulations* with parameters $X_i \in \Omega_{3D}$).
2. Then, for each 3D sigma-simulation with parameter values $X_i \in \Omega_{3D}$, we estimate parameter values $Y_i \in \Omega_{0D}$ of a *coupled 0D simulation* which approximates the outputs \mathcal{O} of the 3D sigma-simulation.
3. From those 3D model parameters X_i and 0D model parameters Y_i , we derive a *parameter mapping* ϕ which converts 3D parameters into 0D parameters.
4. Finally, we approximate the global outputs values $\mathcal{O}_{3D}(x)$ of all the 3D simulations $x_i \in \Omega_{3D}$, from the 0D simulations with parameters $\phi(x_i) \in \Omega_{0D}$. This is done by adding a correction term ψ which is learnt to avoid numerical errors in the previous steps.

The overall process is illustrated in Fig. 2. In the sequel, we first discuss the selection of representative *sigma-simulations* (Sect. 3.2), then the computation of *coupled 0D simulations* (Sect. 3.3), then the *parameter mapping* ϕ (Sect. 3.4) and the *correction term* ψ (Sect. 3.5). Finally, we give numerical results of the multifidelity approximation in Sect. 3.6.

3.2 Sigma-simulations: performing representative 3D simulations within the domain of interest

We consider a subset $\mathcal{P}'_{3D} \subset \mathcal{P}_{3D}$ of $N_2 < N_1$ parameters which cannot be converted directly into 0D model parameters. In order to assess the global outputs variations to these parameters in the set of $x_i \in \Omega_{3D}$, we perform a few selected simulations in the domain Ω_{3D} .

To this end, we perform PCA on the set of $x_i \in \Omega_{3D}$, which gives N_1 eigenvectors of the set. Then, we extract the N_2 eigenvectors \mathbf{d}_k which display the maximal variations of the parameters \mathcal{P}'_{3D} . This is done by sorting the eigenvectors by the norm of their projection of the subspace made by the coordinates corresponding to the parameters \mathcal{P}'_{3D} , and selecting the N_2 largest.

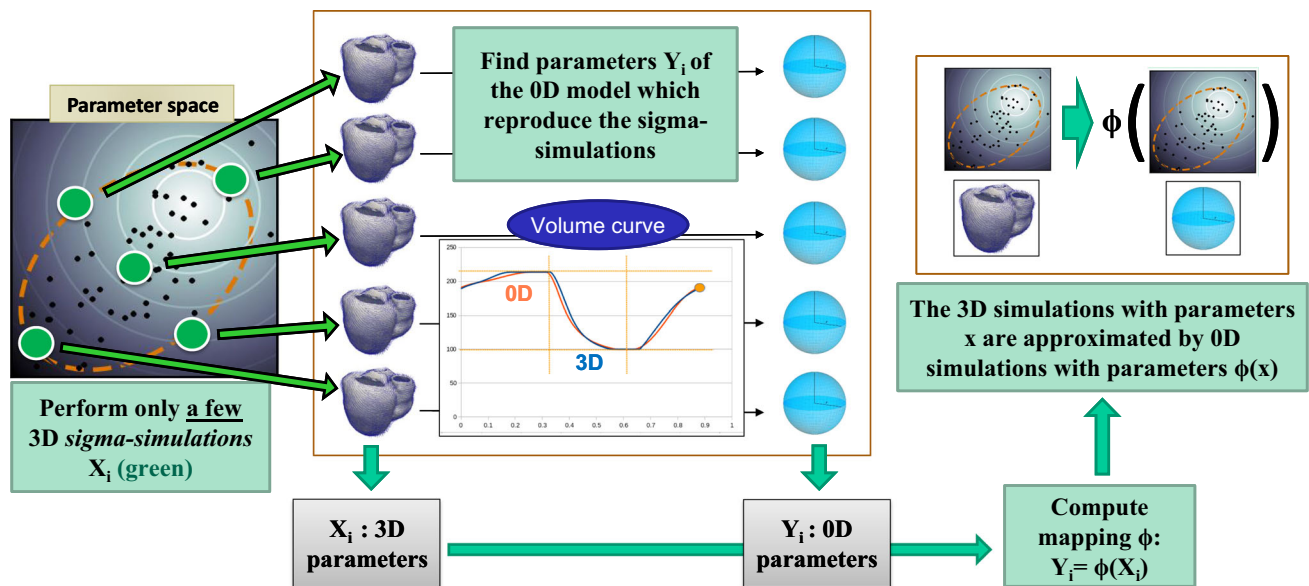


Fig. 2 Multifidelity coupling: After performing a few 3D *sigma-simulations*, we find *coupled 0D simulations* for each of those simulations. Then, we build a *parameter mapping* which converts parameters

of the 3D model into parameters of the 0D model, in order to approximate 3D simulations outputs with the outputs of 0D simulations

One *sigma-simulation* is then performed at the centre (X_0) of the domain of interest Ω_{3D} and pairs are performed equidistant of X_0 in each of the N_2 extracted directions ($X_k^+ = X_0 + d_k$ and $X_k^- = X_0 - d_k$ for $k = 1..N_2$). This gives a total of $2N_2 + 1$ representative simulations in the domain.

In our example, the three haemodynamic parameters R_p , C and P_{ve} have the same values and the same equations in both models, so we can use the same values directly in the two models. On the other hand, the *contractility* σ and the *stiffness* c_1 do not have the same values in both models so we need to assess how their variation is going to impact the global outputs. We then extract the $N_2 = 2$ directions for which the variations of σ and c_1 are maximal and perform a total of $2N_2 + 1 = 5$ sigma-simulations with parameters X_0 , X_1^+ , X_2^+ , X_1^- and X_2^- .

3.3 Coupled 0D simulations: reproducing global outputs of the 3D sigma-simulations with 0D simulations

Then, for each sigma-simulation with parameters X_i , $i = 1..2N_2 + 1$ and output values $\mathcal{O}_{3D}(X_i)$, we want to find a corresponding 0D simulation which has similar global outputs values. To this end, we consider another set \mathcal{O}' of global outputs called the *coupling outputs*, and a set of 0D parameters \mathcal{P}'_{0D} called the *coupled 0D parameters*.

We then find values Y_i of the parameters \mathcal{P}'_{0D} for which the *coupling outputs* values $\mathcal{O}'_{0D}(Y_i)$ of the 0D model simulations are the closest from the *coupling outputs* values $\mathcal{O}'_{3D}(X_i)$ of the 3D model sigma-simulations, with all other

parameters being the same in both models. This is what we call a *coupled 0D simulation*.

This is done by performing, for each 3D sigma-simulation $k = 1..N_2$, an independent parameter estimation of the 0D model parameters \mathcal{P}_{0D} using the method presented in Sect. 2.3. The target values $\hat{\mathcal{O}}'$ for the coupling outputs \mathcal{O}' are their values in the corresponding 3D sigma-simulation.

In our example, since we want to approximate outputs from the volume and pressure curves (see Table 1), we need to approximate these curves with the 0D model. We then chose a set of 4 *coupling outputs* \mathcal{O}' from these curves, and a set of 3 *coupled 0D parameters* \mathcal{P}_{0D} of the 0D model to estimate, both listed in Table 2.

After performing the 5 parameter estimations for the 5 sigma-simulations, we found 5 *coupled 0D simulations* with parameters Y_i , $i = 1..5$ which have similar *coupling outputs* values, which we report in Table 3. We also display the pressure and volume curves of the 3D sigma-simulations and *coupled 0D simulations* in Fig. 3.

It is worth noting there is no guarantee that we can find a set of parameters for which the 0D simulation has exactly the same global outputs values as the 3D simulation. In fact, we can observe in Table 3 that some coupling outputs do not have the same values in a 3D sigma-simulation and the 0D corresponding coupled simulations. We will see in a subsequent section how this obstacle can be overcome.

We point out that there are many possibilities to choose the sets of *coupling outputs* \mathcal{O}' and *coupled 0D parameters*. For example, another possibility would have been to use directly the set of outputs to approximate \mathcal{O} . This would have led to 0D and 3D simulations with the same stroke volume, but

Table 2 Coupling outputs, Coupled 3D parameters, Coupled OD parameters which are estimated and Directly Mapped Parameters in the example of

Coupling outputs \mathcal{O}
Maximal volume V_{\max}
Minimal volume V_{\min}
Mean aortic pressure MP
Diastolic aortic pressure DP
Coupled 3D parameters \mathcal{P}'_{0D}
Contractility σ
Stiffness c_1
Coupled OD parameters \mathcal{P}_{0D}
Contractility σ
Stiffness c_1
Resting radius r_0
Directly mapped parameters
Peripheral resistance R_p
Aortic compliance C
Central venous pressure P_{ve}

Table 3 Coupling outputs values for the 3D sigma-simulations with parameters X_i and the corresponding coupled OD simulations with parameters Y_i

	Vmax (ml)	Vmin (ml)	MP(Pa)	DP (Pa)
X_0	129	51.8	10278	7290
X_1^+	129	35.5	10034	7590
X_2^+	129	64.4	9556	7614
X_1^-	129	69.8	10743	7906
X_2^-	129	40.4	10761	6664
Y_0	128	53.9	10318	7310
Y_1^+	125	41.0	10107	7679
Y_2^+	128	66.3	9591	7634
Y_1^-	129	70.4	10759	7910
Y_2^-	128	42.5	10806	6683

not necessarily the same minimal and maximal volumes. In general, the sets of \mathcal{O}' and \mathcal{O} have to be related so that it is possible to calculate the values of the outputs \mathcal{O} from the values of the coupling outputs \mathcal{O}' .

Similarly, there are many possibilities to choose the sets of coupling OD parameters. Here we could also have set the resting radius in the OD model to a value for which the “resting volume” is the same than in the 3D model and then estimate only the stiffness and contractility of the OD model. Empirically, it seems to be a good thing to use more param-

eters to avoid ending in a local minimum during the parameter estimation of the OD simulations.

3.4 Parameter mapping: a function to convert 3D model parameters into OD model parameters

We now have a corresponding coupled OD simulation with parameters $Y_i \in \Omega_{0D}$ for each sigma-simulation with parameters $X_i \in \Omega_{3D}$. The second idea of the coupling is to build a mapping ϕ between the 3D and OD model parameters using the X_i and Y_i . This mapping will then be used to approximate global outputs values $\mathcal{O}_{3D}(x_i)$ of the 3D simulations with parameters x_i , from the values $\mathcal{O}_{0D}(\phi(x_i))$ of the OD simulations with parameters $\phi(x_i)$.

The parameters $X_i \in \Omega_{3D}$ were chosen in a specific way in Sect. 3.2: one (X_0) is at the centre of the x_i , and there are two equidistant of X_0 (X_k^+ and X_k^- for $k = 1..N_2$) for each of the N_2 axis, which are orthogonal from each other. However, the Y_i were independently estimated for each sigma-simulation so there is no such relationship.

For the mapping ϕ , we use here a degree 2 hypersurface which interpolates the Y_i in the points X_i . In dimension 1, this is equivalent to finding a degree 2 polynomial which interpolates three specific points. In higher dimension (N_2 in our case), there is a straightforward formula because of the specific organisation of X_i along orthogonal axis:

$$\begin{cases} c_k = \frac{(\mathbf{x} - \mathbf{X}_0) \cdot \mathbf{d}_k}{\|\mathbf{d}_k\|^2}, & k = 1..N_2 \\ \mathbf{F}_k^+ = \frac{(Y_k^+ - Y_0) + (Y_k^- - Y_0)}{2}, & k = 1..N_2 \\ \mathbf{F}_k^- = \frac{(Y_k^+ - Y_0) - (Y_k^- - Y_0)}{2}, & k = 1..N_2 \\ \phi(x) = Y_0 + \sum_{k=1}^{N_2} c_k \cdot (c_k \cdot \mathbf{F}_k^+ + \mathbf{F}_k^-). \end{cases} \quad (1)$$

This formula leads to $\phi(X_i) = Y_i$ for all the $i = 1..2N_2 + 1$ sigma-simulations, so the parameters of the 3D sigma-simulations are mapped to the parameters of the coupled OD simulations of the previous section. We will then use this mapping to approximate global outputs of 3D simulations with parameters x_i from OD simulations with parameters $\phi(x_i)$.

3.5 Approximating global outputs: correcting bias

Ideally in the computation of coupled OD simulations in Sect. 3.3, we find OD simulations with the same coupled outputs values than the 3D sigma-simulations, i.e. $\mathcal{O}'_{3D}(X_i) = \mathcal{O}'_{3D}(Y_i)$. As illustrated in Table 3, this is not always the case and the coupled outputs values can be different between the coupled OD simulations and the sigma-simulations. This also means that the direct approximation of the sigma-simulations output values $\mathcal{O}_{3D}(X_i)$ by the values $\mathcal{O}_{0D}(Y_i)$ through the mapping has a bias due to this difference.

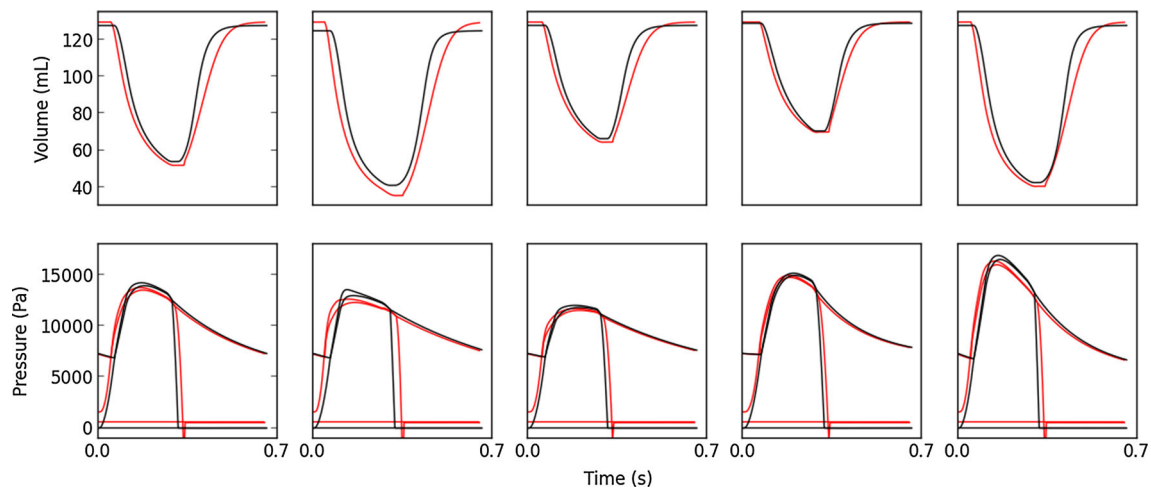


Fig. 3 Comparison between the volume (top) and pressure (bottom) curves of the sigma-simulations simulated with the 3D model (red), and the corresponding *coupled 0D simulations* (black). The 5 columns correspond, respectively, to the sigma-simulations with parameters X_0 , X_1^+ , X_2^+ , X_1^- and X_2^-

In order to correct this approximation bias, both for the output values of the sigma-simulations and all the subsequent 3D simulations with parameters x_i , we build a new degree 2 hypersurface ψ between the parameters of the sigma-simulations X_i and the bias. The formula is exactly the same as in Eq. 1 where the Y_i are replaced by the bias values ($\mathcal{O}_{3D}(X_i) - \mathcal{O}_{0D}(Y_i)$).

The final *approximating function* $\mathcal{C}_{\phi,\psi}$ used to approximate the $\mathcal{O}_{3D}(x_i)$ is then given by the following formula:

$$\mathcal{C}_{\phi,\psi}(x_i) = \mathcal{O}_{0D}(\phi(x_i)) + \psi(x_i) \approx \mathcal{O}_{3D}(x_i), \quad (2)$$

and interpolates in particular the *global outputs* values $\mathcal{O}_{3D}(X_i)$ of the sigma-simulations.

3.6 Approximation results

Results are given here for the approximation of the *global outputs* values $\mathcal{O}_{3D}(x_i)$ of the 30 simulations with parameters x_i . We compute the mean absolute error made on the approximation of the 3 global outputs \mathcal{O} , first with the biased approximation with $\mathcal{O}_{0D}(\phi(x_i))$ (MAE_{Biased}) and then with the corrected approximation with $\mathcal{C}_{\phi,\psi}$ ($MAE_{\text{Corrected}}$). Results are reported in Table 4.

We observe a good approximation of the output values compared to the range of values to be approximated and that the *corrected approximation* makes a better approximation of the outputs values than the *biased approximation*. This means the hypersurface ψ indeed corrects errors due to the differences between the *coupled 0D simulations* and the 3D sigma-simulations.

Finally, we compare our method to an interpolation with a degree 2 hypersurface ($MAE_{\text{Hypersurface}}$). To this end, we use the same formula than Eq. 1, where the Y_i are replaced

Table 4 Error in the approximation of the global outputs values $\mathcal{O}_{3D}(x_i)$ with the various methods

Global output	SV(ml)	DP (Pa)	MP(Pa)
Range	38.23	3010	2254
$MAE_{\text{Corrected}}$	1.59	56.4	137
MAE_{Biased}	4.58	62.8	140
$MAE_{\text{Hypersurface}}$	2.09	511	408
$MAE_{\text{Hypersurface-11}}$	0.25	174	93

by the output values $\mathcal{O}_{3D}(X_i)$. We see in particular that our method performs better on all the outputs ($MAE_{\text{Corrected}} < MAE_{\text{Hypersurface}}$), in particular on the pressure values. This is because the sigma-simulations are computed only in the directions of maximal variations of the parameters σ_0 and c_1 (see Sect. 3.2). There is then a few directions of the parameter space in which the variations of global output values could not be evaluated by the interpolation.

In order to compare more fairly to an interpolation method, we computed the sigma-simulations in all the directions of the domain by selecting all the eigenvectors in Sect. 3.2, leading to $2 \cdot N_1 + 1 = 11$ sigma-simulations. We performed the degree 2 interpolation from these 11 sigma-simulations and report the results ($MAE_{\text{Hypersurface-11}}$). The degree 2 hypersurface performs better than our method on the stroke volume and the mean pressure but not on the diastolic pressure.

We conclude that the approximation using the coupling of the 0D and 3D models gives competitive approximation results compared to the classical regression methods, and with the lowest computational cost. This is because the variations of some outputs (which rely on the same equations in both models) can be directly approximated in some direc-

tions of the parameter space, without having to compute 3D simulations in these directions. Here in particular, the pressure outputs variations due to changes in the haemodynamic parameters C , R_p and P_{ve} are correctly predicted with the coupling (especially the diastolic aortic pressure (DP) variations), even though no sigma-simulation was computed in the directions of maximal variation of these parameters (Sect. 3.2). As a consequence, only 5 sigma-simulations are required to approximate all the outputs values within the parameter space with the coupling, while the hypersurface interpolation needs 11 sigma-simulations to achieve similarly accurate results.

4 Multifidelity optimisation for efficient 3D cardiac model personalisation

Here we present our *multifidelity personalisation method* for the 3D model. We suppose that a parameter estimation with CMA-ES was set up over N_1 parameters \mathcal{P} of the 3D model as described in Sect. 2.3, some global outputs \mathcal{O} , some target values $\hat{\mathcal{O}}$ and a population size m . The idea of the method is to replace the scores of 3D simulations in CMA-ES with *approximate scores* calculated through *multifidelity coupling*.

We illustrate the method with the same set of 5 parameters \mathcal{P} and 3 outputs \mathcal{O} as in Sect. 3 and the same number $m = 30$ for the population size. Target values $\hat{\mathcal{O}}$ for the optimisation are, respectively, **60 ml** for the stroke volume (SV), **7315 Pa** for the diastolic aortic pressure (DP) and **10,152 Pa** for the mean aortic pressure (MP). The normalisation coefficients for this problem (in the vector \mathcal{N} defined in Sect. 2.3.1) are **10 ml** for the stroke volume (SV), **200 Pa** for the diastolic aortic pressure (DP) and the mean aortic pressure (MP).

4.1 Multifidelity-CMA: CMA-ES optimisation with the multifidelity coupling

At each iteration, the algorithm CMA-ES asks for the scores of m simulations of the 3D model, whose parameters x_j are drawn from a multivariate distribution.

A first approach to replace the computation of the 3D simulations by 0D simulations is to perform the coupling described in Sect. 3 for each generation of CMA-ES. This means recomputing sigma-simulations, *coupled 0D simulations* and a parameter mapping for each set of x_j . This was our approach (called *Coupled-CMA*) in Mollero et al. (2016). We showed that the optimisation could converge with *approximate scores*, even as fast as with the real scores in some cases. We also personalised 34 hearts with this method, thus exhibiting a practical personalisation method with a lower computational burden than the original CMA-ES algorithm

(because only the *sigma-simulations* were computed for each generation instead of the m 3D simulations).

Here we present an improved approach called *multifidelity-CMA*. Instead of recomputing the coupling for each generation, we approximate scores of 3D simulations of *successive generations of CMA-ES*. Indeed, because the sets of parameters x_j^n and x_j^{n+1} asked by CMA-ES in two consecutive generations n and $n + 1$ are usually close, the function $\mathcal{C}_{\phi,\psi}$ computed at the iteration n to approximate 3D simulations with parameters x_j^n can give a good approximation for 3D simulations with parameters x_j^{n+1} as well.

On the other hand, after a few iterations $n + 1..n + p$, the points asked by CMA-ES can be increasingly far from the *sigma-simulations* of the *multifidelity coupling* performed at n . This can lead to approximations of the scores which are increasingly inaccurate, making the optimisation impossible.

We then developed a criterion to evaluate the accuracy of the approximation for a few successive iteration of CMA-ES and then decide at which step a new *multifidelity coupling* has to be computed. This is done by iterating on the following steps:

1. *Coupling step* At a generation n_0 of CMA-ES, we first perform a *multifidelity coupling*, as explained in 3.4. This leads to the computation of the function $\mathcal{C}_{\phi,\psi}$.
2. *Exploration step* Then, we perform N iterations $n = n_0 + 1..n_0 + N$ of the CMA-ES algorithm, where all the outputs $\mathcal{O}_{3D}(x_j^n)$ of the 3D simulations with parameters x_i^n are approximated by $\mathcal{C}_{\phi,\psi}(x_j^n)$.
3. *Control step* For each of these N iterations, we compute a *control-simulation*: the 3D simulation whose parameters o_n are the mean of the population parameters x_j^n .
4. *Selection step* We compute our *criterion* $M(o_n)$ as the *Mahalanobis distance* between the vector of outputs values $\mathcal{O}_{3D}(o_n)$ of the control-simulation and the set of vectors of approximated outputs values $\mathcal{C}_{\phi,\psi}(x_j^n)$. Finally, we select the iteration n^* at which the next **coupling step** is performed with the following criteria:

$$n^* = \operatorname{argmin}_{M(o_n) < \gamma \sqrt{|\mathcal{O}|}} \mathcal{O}_{3D}(o_n)$$

The process is illustrated in Fig. 4. The Mahalanobis distance $M(o_n)$ is a ratio between the approximation error on the *control-simulation* output values, and the range of *approximate outputs values* for this generation. Roughly, this gives an indication on “how accurate the coupling is” on the *control-simulation*, compared to “how accurate it needs to be” so that CMA-ES can rank the scores accurately.

For example in Fig. 5, we report for $N = 10$ iterations the scores of the control-simulations o_n which were predicted through the function $\mathcal{C}_{\phi,\psi}$ (in black), and the real scores

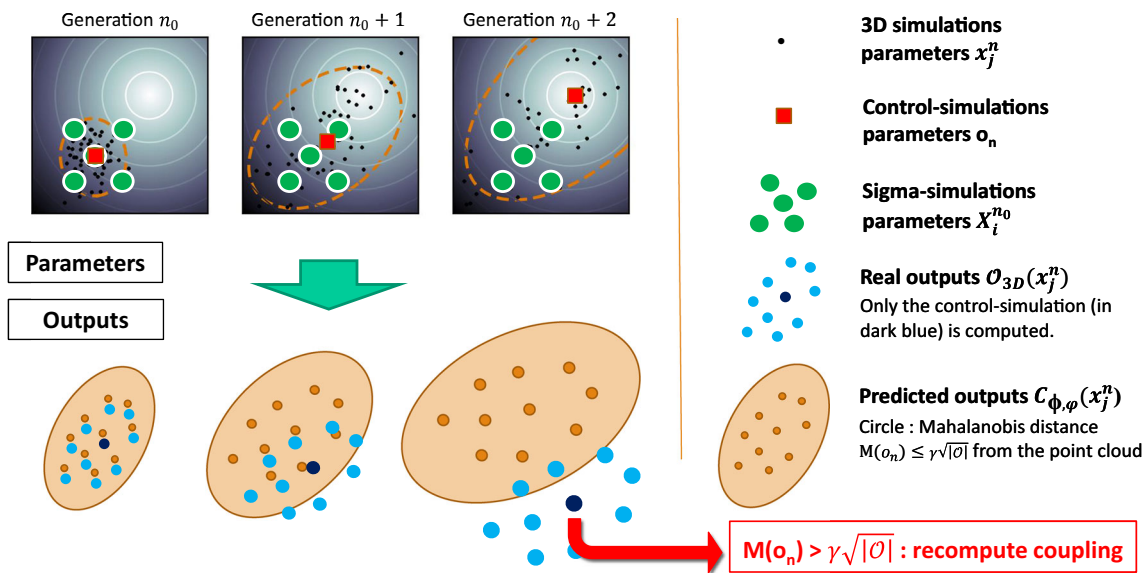


Fig. 4 Criteria for selecting the generation for the next **coupling step** in the **selection step**: as the 3D parameters of the simulations asked by CMA-ES (in black) are increasingly far from the sigma-simulations (in green) of the coupling, the predicted outputs values with 0D simulations

(in orange) are increasingly far from the real outputs values of the 3D simulations. We then recompute the coupling when this distance is too high ($M(o_n) > \gamma\sqrt{|\mathcal{O}|}$)

of these simulations (in blue). Simultaneously, we show the criterion $M(o_n)$ for these N iterations and the upper value (red line) $\gamma\sqrt{|\mathcal{O}|}$ for the criterion ($\gamma = 1.5$ here).

We can see that the score prediction (thus the approximation of the outputs $\mathcal{O}_{3D}(x_j^n)$ values by $\mathcal{C}_{\phi, \psi}$) is quite accurate for at least the 5 first iterations and is less accurate for $n \geq 6$. Then, even though the score prediction seems as accurate at the iteration 5 than at the generation 1, $M(o_n)$ is higher. This is because the prediction error is more important *relatively to the set of $\mathcal{C}_{\phi, \psi}(x_j^n)$* of the generation, in particular in directions where the set has a lower variance.

In this example, the iteration 5 was selected to recompute the coupling (black vertical line), which is also the iteration where the control-simulation has the minimal score over the 10 iterations. In some cases, later iterations can have a lower score but are not selected because the criterion $M(o_n)$ is too high for this iteration (such as the iteration 7).

The upper bound $\gamma\sqrt{|\mathcal{O}|}$ for the criterion has an important impact on the optimisation behaviour. If a high accuracy is imposed (small γ value), then one of the earlier iterations of the **exploration step** is usually selected for the subsequent **coupling step**, even if a later control-simulation has a lower score. This can lead to a slow optimisation. On the other with a small accuracy (high γ value), the CMA-ES algorithm can end up in local minima because it performed the optimisation on inaccurate values.

Therefore, the value of γ characterises a trade-off between maximising the optimisation gain with a single coupling and ensuring the approximation errors do not impact the opti-

misation process. Because of the probabilistic nature of the algorithm and the various nonlinearities of the score function, the optimal value of γ seems very dependent on the optimisation problem. We found $\gamma = 1.5$ to give good convergence results in our experiments, and the number n^* of the iteration selected in the **selection step** is 5.5 in average in our experiments.

4.2 Computational considerations: a parallelisable method

The main computational cost in personalisation methods comes from the computation of the 3D simulations. In our implementation, each simulation of one heartbeat with the 3D model uses one CPU, during a time T_{3D} which depends mostly on the size of the mesh, and the heartbeat duration.

Most of the modern research is performed on computer clusters which can perform many tasks at the same time. In particular, in our method, many steps can be parallelised. To compare different optimisation methods in a parallel setting, we introduce here two metrics: the classic *CPU time* which measures the total amount of CPU resources used and the *optimisation time* which measures the duration of the optimisation in (real) time.

During one complete iteration of *multifidelity-CMA*, the following steps are parallelised:

1. Computation of the $2N_2 + 1$ 3D sigma-simulations: the simulations are performed in parallel and each one takes

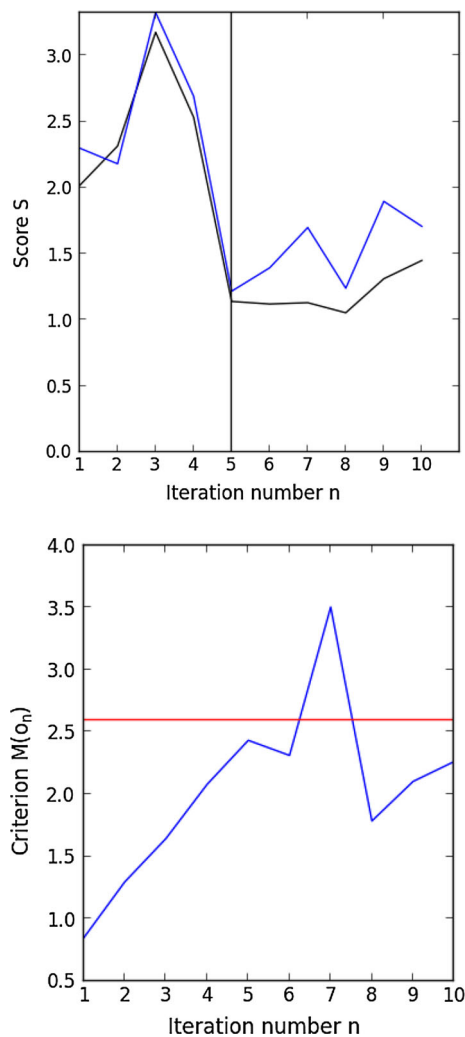


Fig. 5 Top: real scores (blue) and approximated scores (black) of the $N = 10$ control-simulations. Bottom: Value of the criterion $M(o_n)$ of the control-simulations

a CPU time T_{3D} . The whole step has then a CPU time of $(2N_2 + 1) \cdot T_{3D}$ and an optimisation time of T_{3D}

2. Computation of the coupled 0D simulations: all the parameter estimations are performed in parallel. Each one uses 4 CPUs during fixed time of around 3 minutes. The whole step has a CPU time of $(2N_2 + 1) \cdot 4 \cdot 3$ minutes and an optimisation time of 3 minutes.
3. Computation of the N 3D control-simulations: the simulations are performed in parallel and each one takes a CPU time T_{3D} . The whole step has a CPU time of $N \cdot T_{3D}$ and an optimisation time of T_{3D}

In our example, we have 5 sigma-simulation and 10 control-simulation, and the 3D simulation takes 15 minutes. Each iteration of multifidelity-CMA then takes a total CPU time of $5 \cdot 15 + 4 \cdot 5 \cdot 3 + 10 \cdot 15 = 285$ minutes and an optimisation time of 33 minutes.

4.3 Results: comparison of optimisation time, CPU time for 4 personalisation methods

Here we compare the evolution of the CPU time and the score S during optimisation on a typical case, with the 4 following optimisation methods:

1. The multifidelity-CMA method with 0D/3D coupling.
2. The multifidelity-CMA method where the approximation of outputs is done with a degree 2 hypersurface interpolation relying on 11 sigma-simulations (as explained in Sect. 3.6).
3. The classic CMA-ES method with a population size of $m = 30$.
4. BOBYQA, which is another commonly used gradient-free optimiser, for example, to solve personalisation problems (Seegerer et al. 2015) or as a baseline to evaluate other personalisation methods (Neumann et al. 2016). It uses trust region method and forms successive quadratic models of the score function which interpolates the points computed during optimisation.

Results are shown in Fig. 6. We can see that BOBYQA (red) is slow to converge, but has also a low computational cost, both due to the fact that BOBYQA performs only one iteration at a time. The normal CMA-ES (blue) converges faster than BOBYQA, but with a very high computational cost because 30 simulations of the 3D model are computed at each generation.

Finally, both our multifidelity approaches are very fast to converge; however, the multifidelity-CMA which uses the 0D/3D multifidelity coupling is the one with the lowest CPU time (because only 5 sigma-simulations per complete iteration is computed instead of 11, as explained in Sect. 3.6).

We conclude that the multifidelity approach of the CMA-ES algorithm leads to considerable improvements in optimisation speed, both from the original CMA-ES algorithm and BOBYQA. Finally, the approximation of outputs with a 0D/3D multifidelity coupling instead of a generic hypersurface interpolation leads to additional computational gains.

4.4 Results: personalisation of a database of 121 cases

We finally present results on a large database of 121 cases. For each patient, a biventricular heart mesh geometry (between 10,000 and 15,000 nodes) was built from the available MRI image and the boundaries of the myocardium were tracked in the cine MRI images as described in Jolly et al. (2011) and Wang et al. (2013). This led to the computation of the volume curve and then the value of the stroke volume. Pressure measurements were also available for each heartbeat.

We applied our multifidelity-CMA method to personalise the whole cohort. The optimisation started from a vector

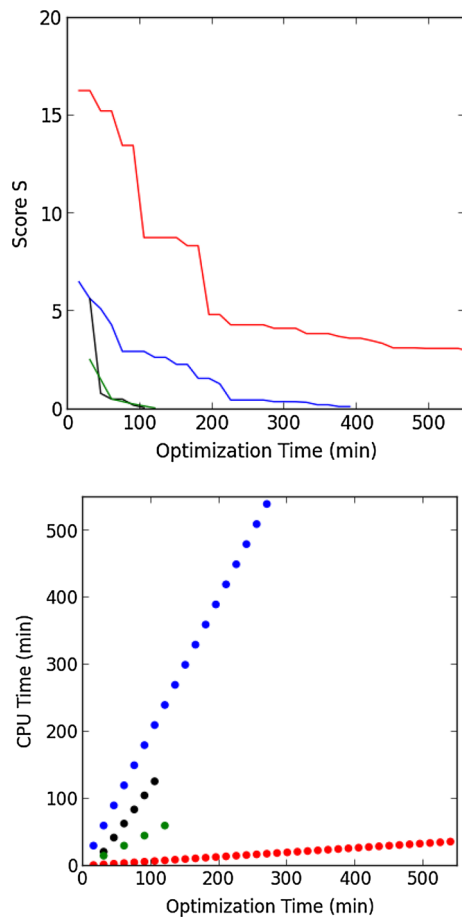


Fig. 6 Comparison of the evolution of the score S (top) and CPU time (bottom) during optimisation for the four methods. BOBYQA is in red, the classic CMA-ES is in blue, multifidelity-CMA with the hypersurface approximation is in black, and multifidelity-CMA with 0D/3D coupling is in green

x_{start} of parameter values which has the same values for every patient, except for P_{ve} , which is set at the value DP-2000 Pa (see Table 6). The algorithms ran for around 2.5 days, and the BOBYQA optimisation was ran on the same problems during this period as well.

We consider a personalisation to be *successful* when a set of parameter values was found with a score lower than $l_1 = 0.1$, and *acceptable* if the score is lower than $l_2 = 1$. This means the personalised simulation matches the target stroke volume within 1 ml and the pressure measurements within 20 Pa for the *successful* case, and within, respectively, 10 ml and 200 Pa in the *acceptable* case. In other cases, the personalisation is said *failed*. We report the number of *successful*, *acceptable* and *failed* cases on this database, for both methods in Table 5.

A high number of cases were successfully personalised (113 among 121 cases) with our method. For the acceptable cases, and one of the failed case, the optimisation had converged in a local minima. For the other failed case, the CMA-ES algorithm diverged to extreme parameter values

Table 5 Results of the personalisation on the database

Result	Successful	Acceptable	Failed
Multifidelity-CMA	113	6	2
BOBYQA	5	69	47

during optimisation. For BOBYQA, the convergence was not yet reached in most of the non-successful cases (the score is the lowest in the last iteration).

We finally report the mean and standard deviation of all the estimated parameter values, in Table 6, as well as the norm of their relative variation $|\Delta|$ compared to the *starting value* during the optimisation. This shows in particular that the stiffness c_1 did not change a lot during the personalisation process. The arterial compliance C and the contractility σ_0 are the parameters which changed the most.

5 Discussion and conclusion

We presented a novel multifidelity approach involving a 3D cardiac electromechanical cardiac model and a simplified 0D model, which relies on the same equations but with simplifying assumptions. We developed an original *multifidelity coupling* between the parameters of both models, which gives a good *multifidelity approximation* of *global output values* in 3D simulations from 0D simulations. We then used this approximation in an efficient parameter estimation process using the genetic algorithm CMA-ES, in order to have an efficient *multifidelity personalisation method* for the 3D model.

Our *multifidelity coupling* procedure computes a mapping between the parameters of a few representative 3D *sigma-simulations* within the domain, and the parameters of corresponding *coupled 0D simulations* with the same output values. This is done through parameter estimation on the 0D model parameters to compute *coupled 0D simulations* that have the same *global outputs values* than 3D *sigma-simulations*. The parameter mapping is then derived through an interpolation method.

This enables to get fast and accurate approximations of 3D simulations with the 0D model. These approximations are then used in the parameter estimation of 3D model parameters with CMA-ES, to replace 3D simulations while simultaneously controlling the accuracy of the approximation and recomputing a coupling when the accuracy is too low. Ultimately, this results into both an increase in the speed of the 3D parameter estimation process and a decrease in the computational cost.

Our multifidelity approach slightly differs from more classic multifidelity methods (Kennedy and O'Hagan 2000; Peherstorfer et al. 2016) where the same parameter values are used as input of both models, and the outputs of the low-

Table 6 Statistics of the estimated parameter values and their variations during the personalisation

	c_1 (kPa)	σ (MPa)	P_{ve} (Pa)	R (MPa m ³ s)	C (MPa ⁻¹ m ⁻³)
x_{start}	50.1e ¹	68.8	DP-2000	54.1	18.0e ⁻³
Mean	50.5e ¹	91.6	4760	68.2	8.17e ⁻³
Std.	0.36e ¹	36.5	1340	16.3	2.41e ⁻³
Mean $ \Delta $	4.74%	53.9%	26.2%	35.8%	54.7%

fidelity model are corrected *a posteriori* to fit the outputs of the high-fidelity model. Since the parameters of both models are not exactly the same, we had to find a mapping between the *parameters* instead of the *outputs*. This was tractable thanks to the fast parameter estimation in the 0D model.

A first extension of the *multifidelity coupling* would be to use additional shared parameters and equations in both models, to approximate a larger variety of outputs of the 3D model (e.g. flow velocities, timings of valve opening and closing). Since CMA-ES has already been proven successful on complex optimisation problems with a larger parameter space, we expect the personalisation method to scale well. A second extension would be to use the *multifidelity personalisation* to personalise “geometrical” or “local” measurements which are outputs of the 3D model but not of the 0D model (e.g. the septal shortening or the circumferential torsion). Indeed, even though they cannot be approximated through the 0D/3D *multifidelity coupling*, their values can still be locally approximated during personalisation using the hypersurface interpolation.

Finally, the lower-fidelity approximation could be used not only for personalisation but also for other applications that require many simulations, such as parameter sensitivity or uncertainty quantification (with Monte–Carlo methods, for example) and also for applications simulations that require the computation of many cardiac cycles. In particular, a case where the multifidelity approach could be useful is when the 3D model is coupled with a full-body circulation model as boundary conditions. Indeed, studies associated with such models (for example on the influence of physical exercise, increased heart rate and/or pressure loads) usually require many heartbeats to be computed. This can be computationally intensive with the 3D model, but it could be done faster using 0D simulations, through a similar coupling method than in this manuscript. In this case where the number of coupled parameters would be high, additional constraints could be added in the parameter mapping to impose correlations between parameters with different equations or values but a similar behaviour.

Acknowledgements This work has been partially funded by the European Union FP7-funded project MD-Paedigree (Grant Agreement 600932) and contributes to the objectives of the European Research Council advanced Grant MedYMA (2011-291080).

Compliance with ethical standards

Conflicts of interest The authors declare that they have no conflict of interest.

Informed consent Informed consent was obtained from the subjects, and the protocol was approved by the local research ethics committee.

6 Appendix A: Mechanical equations and haemodynamics

As described in [Marchesseau et al. \(2013a\)](#) our 3D electromechanical model is based on the Bestel–Clement–Sorine model (BCS) of sarcomere contraction as extended by [Chapelle et al. \(2012\)](#), in conjunction with a Mooney–Rivlin energy for the passive hyperelasticity. Haemodynamics are represented through global values of pressures and flows in the cardiac chambers and coupled to the mechanical equations with the Windkessel model of blood pressure for the afterload (aortic pressure).

6.1 The BCS model: active contraction and passive material

The BCS model describes the sarcomere forces as the sum of an active contraction force in the direction of the fibre, in parallel with a passive isotropic visco-hyperelastic component (see Fig. 1b). It is compatible with the laws of thermodynamics and allows to model physiological phenomena at the sarcomere scale which translate at the macroscopic scale (such as the Starling effect).

The active force in the sarcomere is modelled by the filament model of [Huxley \(1957\)](#), which describes the binding/unbinding process of the actin and myosin in the sarcomere at the nanoscopic scale. At the mesoscopic scale, it results ([Caruel et al. 2014](#)) in a differential equation which relates the active stress τ_c , the stiffness k_c and the strain e_c of the filament within the sarcomere:

$$\begin{cases} \dot{k}_c = -(|u|_+ + |u|_- + \alpha|\dot{e}_c|)k_c + k_0|u|_+, \\ \dot{\tau}_c = -(|u|_+ + |u|_- + \alpha|\dot{e}_c|)\tau_c + \dot{e}_c k_c + \sigma_0|u|_+, \end{cases} \quad (3)$$

where α is a constant related to the cross-bridge destruction during contraction, and k_0 and σ_0 are, respectively, the maximum stiffness and contraction. The values of $|u|_+$ and $|u|_-$

are, respectively, the rate of build-up k_{ATP} and decrease k_{RS} of the force during contraction and relaxation, which depends on the depolarisation and repolarisation times T_d and T_r of the sarcomere:

$$u = \begin{cases} k_{ATP} & \text{when } T_d \leq t \leq T_r \\ -k_{RS} & \text{otherwise} \\ |u|_+ = \max(u, 0), \\ |u|_- = -\min(u, 0). \end{cases} \quad (4)$$

This active force is applied in the direction of the fibre through the visco-elastic component, made of a spring E_s and a dissipative term μ (see Fig. 1b). As derived in Caruel et al. (2014), the resulting stress σ_{1D} in the fibre direction is given by:

$$\begin{cases} \sigma_{1D} = E_s \frac{e_{1D} - e_c}{(1 + 2e_c)^2}, \\ (\tau_c + \mu \dot{e}_c) = E_s \frac{(e_{1D} - e_c)(1 + 2e_{1D})}{(1 + 2e_c)^3}, \end{cases} \quad (5)$$

where $e_{1D} = \tau_1 \cdot \underline{e} \cdot \tau_1$ is the strain in the fibre direction τ_1 (\underline{e} is the Green–Lagrange strain tensor).

Finally, for the passive component the isotropic Mooney–Rivlin model of hyperelastic material is used, driven by the following strain energy:

$$W_e = c_1(I_1 - 3) + c_2(I_2 - 3) + \frac{K}{2}(J - 1)^2, \quad (6)$$

where I_1 , I_2 and J are the invariants of the Cauchy–Green deformation tensor, c_1 , c_2 and K are the parameters of the material.

6.2 Haemodynamic model

To model the influence of blood dynamics during the cardiac circle, the mechanical equations are coupled with a basic circulation model implementing the 4 phases of the cardiac cycle. For a given ventricle, if we note P_{at} the pressure in the atrium, P_{ar} the pressure in the artery and P_V the pressure in the ventricle, the phases are the following:

- *Diastolic filling* when $P_V \leq P_{at}$, the atrial valve is open and the ventricle fills up with blood.
- *Isovolumetric contraction* when contraction starts, P_V rises. $P_{at} \leq P_V \leq P_{ar}$ and all the valves are closed.
- *Systolic ejection* when $P_V \geq P_{ar}$, the arterial valve opens and the blood is ejected into the artery.
- *Isovolumetric relaxation* when the contractile forces disappear, P_V finally decreases. $P_{at} \leq P_V \leq P_{ar}$ again, and all the valves are closed.

We use the haemodynamic model introduced by Chapelle et al. (2012) which links the blood flow q to the ventricular, atrial and arterial pressures with the following equations:

$$q = \begin{cases} K_{at}(P_V - P_{at}) & \text{for } P_V \leq P_{at} \\ K_{iso}(P_V - P_{at}) & \text{for } P_{at} \leq P_V \leq P_{ar} \\ K_{ar}(P_V - P_{at}) + K_{iso}(P_{ar} - P_{at}) & \text{for } P_V \geq P_{ar} \end{cases} \quad (7)$$

Here the atrial pressure $P_{at}(t)$ (cardiac preload) is imposed at a constant value P_{at_lower} except for a *pressure bump* up to P_{at_upper} at the beginning of cardiac cycle, to account for the contraction of the atrium before the ventricular contraction. Finally, the pressure of the artery P_{ar} (cardiac afterload) is modelled with the 3-parameter Windkessel model (West-erhof et al. 1969) and coupled to the ventricular outflow q through the equation:

$$R_p C \dot{P}_{ar} + P_{ar} - P_{ve} = (R_p + Z_c)q + R_p Z_c C \dot{q}, \quad (8)$$

where R_p is the *Peripheral resistance*, Z_c is the characteristic impedance, C is the *arterial compliance*, and P_{ve} is the *central venous pressure*.

6.2.1 Implementation

The passive Mooney–Rivlin energy is discretised on the 3D mesh with the MJED (multiplicative Jacobian energy decomposition) method described in Marchesseau et al. (2010), and the BCS fibre stress and stiffness are computed at each node, separately from the positions and velocities. This allows a fast assembly and a good conditioning of the system of mechanical equations. A Rayleigh damping is then added to account for the viscous global dissipation, and finally, the ventricular pressure is computed using a prediction–correction approach, performed after solving the first system of mechanical equations. This efficient algorithm and all the details of the mechanical equations and their 3D discretisations are fully discussed in Marchesseau et al. (2013a).

7 Appendix B: Reduced equations of the 0D model

7.1 Mechanical equations

The list of simplified equations of our 0D model is reported in Table 7. Equations (a), (b), (c) and (f) are the same sarcomere and visco-elastic equations than Eqs. 3 and 5, which are calculated once for the whole sphere. C in equations (d), (e), (g) and (h) denotes a component of the simplified Cauchy–Green deformation tensor which depends only on $y = R - R_0$. $\sigma_{passive}$ in equation (g) is the stress due to the passive law, and $\sigma_{viscosity}$ in equation (h) is the stress due to

Table 7 Mechanical equations of the 0D model

$$\begin{cases}
 \dot{k}_c = -(|u|_+ + |u|_- + \alpha|\dot{e}_c|)k_c + k_0|u|_+ & \text{(a)} \\
 \dot{\tau}_c = -(|u|_+ + |u|_- + \alpha|\dot{e}_c|)\tau_c + \dot{e}_c k_c + \sigma_0|u|_+ & \text{(b)} \\
 (\tau_c + \mu\dot{e}_c) = E_s \frac{(e_{1D} - e_c)(1 + 2e_{1D})}{(1 + 2e_c)^3} & \text{(c)} \\
 C = (1 + \frac{y}{R_0})^2 & \text{(d)} \\
 e_{1D} = \frac{C - 1}{2} & \text{(e)} \\
 \sigma_{1D} = E_s \frac{e_{1D} - e_c}{(1 + 2e_c)^2} & \text{(f)} \\
 \sigma_{\text{passive}} = 4(1 - C^{-3})(c_1 + c_2 C) & \text{(g)} \\
 \sigma_{\text{viscosity}} = 4\eta(1 + C^{-6})\dot{C} & \text{(h)} \\
 \Sigma_{\text{sph}} = \sigma_{1D} + \sigma_{\text{passive}} + \sigma_{\text{viscosity}} & \text{(i)} \\
 \rho d_0 \dot{y} = P_v(1 + \frac{y}{R_0})^2 - \frac{d_0}{R_0}(1 + \frac{y}{R_0})\Sigma_{\text{sph}} & \text{(j)} \\
 q = 4\pi R_0^2(1 + \frac{y}{R_0})^2 \dot{y} = \begin{cases} K_{\text{at}}(P_V - P_{\text{at}}) & \text{for } P_V \leq P_{\text{at}} \\ K_{\text{iso}}(P_V - P_{\text{at}}) & \text{for } P_{\text{at}} \leq P_V \leq P_{\text{ar}} \\ K_{\text{ar}}(P_V - P_{\text{at}}) + K_{\text{iso}}(P_{\text{ar}} - P_{\text{at}}) & \text{for } P_V \geq P_{\text{ar}} \end{cases} & \text{(k)} \\
 R_p C \dot{P}_{\text{ar}} + P_{\text{ar}} - P_{\text{ve}} = (R_p + Z_c)q + R_p Z_c C \dot{q} & \text{(l)}
 \end{cases}$$

Table 8 Electrical activation in the 0D model

$$f = \begin{cases} \frac{t - T_{d,\text{global}}}{QRS_{\text{duration}}} & \text{when } T_{d,\text{global}} \leq t \leq T_{d,\text{global}} + QRS_{\text{duration}} \\ 1 & \text{when } T_{d,\text{global}} + QRS_{\text{duration}} \leq t \leq T_{r,\text{global}} \\ (1 - \frac{t - T_{r,\text{global}}}{QRS_{\text{duration}}}) & \text{when } T_{r,\text{global}} \leq t \leq T_{r,\text{global}} + QRS_{\text{duration}} \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

$$\begin{aligned}
 |u|_+ &= k_{\text{ATP}} \cdot f_{\text{depo}} \\
 |u|_- &= k_{\text{RS}} \cdot (1 - f_{\text{depo}})
 \end{aligned}$$

an additional viscous damping η , both expressed as a simple function of C (see Caruel et al. (2014) for the full derivations). In equation (i), Σ_{sph} is the sum of all the stresses applied to the sphere. Equation (j) is the resulting equation of motion which, coupled with the haemodynamic model (k) and the Windkessel equation (l), gives the full system of 3 equations to be solved at each iteration.

7.2 Electrophysiology equations

Assuming synchronous and homogeneous electrical activation (and thus sarcomere force) means that all of the ventricle is depolarised simultaneously. This leads to a rate of ventricular pressure rise during the isovolumetric contraction (respectively, isovolumetric relaxation) which is very close to the rate of build-up k_{ATP} (respectively, decrease k_{RS}) of the active stress τ_c . However in 3D, this rate is also very dependent on the time for the ventricle to be fully depolarised, which is roughly the QRS duration.

In order to correct this discrepancy between the models, we adapted the electrical parameter u to take into account the QRS duration. We model the fraction f_{depo} of the ventricle which is currently depolarised as a piecewise linear function of time which depends on $T_{d,\text{global}}$, $T_{r,\text{global}}$ and QRS_{duration} .

Then, the values of $|u|_+$ and $|u|_-$ in Equation (a) are adapted to depend on the value of f_{depo} as described in Table 8.

References

- Baillargeon B, Rebelo N, Fox DD, Taylor RL, Kuhl E (2014) The living heart project: a robust and integrative simulator for human heart function. *Eur J Mech A Solids* 48:38–47
- Caruel M, Chabiniok R, Moireau P, Lecarpentier Y, Chapelle D (2014) Dimensional reductions of a cardiac model for effective validation and calibration. *Biomech Model Mechanobiol* 13(4):897–914
- Chabiniok R, Moireau P, Lesault P-F, Rahmouni A, Deux J-F, Chapelle D (2012) Estimation of tissue contractility from cardiac cine-MRI using a biomechanical heart model. *Biomech Model Mechanobiol* 11(5):609–630
- Chabiniok R, Wang VY, Hadjicharalambous M, Asner L, Lee J, Sermesant M, Kuhl E, Young AA, Moireau P, Nash MP, Chapelle D, Nordsletten DA (2016) Multiphysics and multiscale modelling, data-model fusion and integration of organ physiology in the clinic: ventricular cardiac mechanics. *Interface Focus* 6(2):20150083
- Chapelle D, Le Tallec P, Moireau P, Sorine M (2012) Energy-preserving muscle tissue model: formulation and compatible discretizations. *Int J Multiscale Comput Eng* 10(2):189–211
- Chen Z, Cabrera-Lozoya R, Relan J, Sohal M, Shetty A, Karim R, Delingette H, Gill J, Rhode K, Ayache N, Taggart P, Rinaldi CA, Sermesant M, Razavi R (2016) Biophysical modelling predicts ventricular tachycardia inducibility and circuit morphology: a combined clinical validation and computer modelling approach. *J Cardiovasc Electrophys* 27(7):851–860

- Clayton R, Bernus O, Cherry E, Dierckx H, Fenton F, Mirabella L, Panfilov A, Sachse FB, Seemann G, Zhang H (2011) Models of cardiac tissue electrophysiology: progress, challenges and open questions. *Prog Biophys Mol Biol* 104(1):22–48
- Cuellar AA, Lloyd CM, Nielsen PF, Bullivant DP, Nickerson DP, Hunter PJ (2003) An overview of cellml 1.1, a biological model description language. *Simulation* 79(12):740–747
- Duchateau N, De Craene M, Allain P, Saloux E, Sermesant M (2016) Infarct localization from myocardial deformation: prediction and uncertainty quantification by regression from a low-dimensional space. *Trans Med Imaging* 35(10):2340–2352
- Garny A, Hunter PJ (2015) Opencor: a modular and interoperable approach to computational biology. *Front Physiol* 6:26
- Geijtenbeek T, van de Panne M, van der Stappen AF (2013) Flexible muscle-based locomotion for bipedal creatures. *ACM Trans Graph (TOG)* 32(6):206
- Hansen N (2006) The cma evolution strategy: a comparing review. Towards a new evolutionary computation. Springer, Berlin, Heidelberg, pp 75–102
- Huxley A (1957) Muscle structure and theories of contraction. *Prog Biophys Biophys Chem* 7:255–318
- Jolly M-P, Guetter C, Lu X, Xue H, Guehring J (2011) Automatic segmentation of the myocardium in cine mr images using deformable registration. *International workshop on statistical atlases and computational models of the heart*. Springer, Berlin, pp 98–108
- Kayvanpour E, Mansi T, Sedaghat-Hamedani F, Amr A, Neumann D, Georgescu B, Seegerer P, Kamen A, Haas J, Frese KS, Irawati M, Wirsz E, King V, Buss S, Mereles D, Zitron E, Keller A, Katus HA, Comaniciu D, Meder B (2015) Towards personalized cardiology: multi-scale modeling of the failing heart. *PLoS ONE* 10(7):e0134869
- Kennedy MC, O'Hagan A (2000) Predicting the output from a complex computer code when fast approximations are available. *Biometrika* 87(1):1–13
- Marchesseau S, Heimann T, Chatelin S, Willinger R, Delingette H (2010) Multiplicative jacobian energy decomposition method for fast porous visco-hyperelastic soft tissue model. *International conference on medical image computing and computer-assisted intervention*. Springer, Berlin, pp 235–242
- Marchesseau S, Delingette H, Sermesant M, Ayache N (2013a) Fast parameter calibration of a cardiac electromechanical model from medical images based on the unscented transform. *Biomech Model Mechanobiol* 12(4):815–831
- Marchesseau S, Delingette H, Sermesant M, Lozoya RC, Tobon-Gomez C, Moireau P, i Ventura RMF, Lekadir K, Hernández AI, Garreau M, Donal E, Leclercq C, Duckett SG, Rhode KS, Rinaldi CA, Frangi AF, Razavi R, Chapelle D, Ayache N (2013b) Personalization of a cardiac electromechanical model using reduced order unscented kalman filtering from regional volumes. *Med Image Anal* 17(7):816–829
- Mollero R, Pennec X, Delingette H, Ayache N, Sermesant M (2016) A multiscale cardiac model for fast personalisation and exploitation. *International conference on medical image computing and computer-assisted intervention*. Springer, Berlin, pp 174–182
- Neumann D, Mansi T, Itu L, Georgescu B, Kayvanpour E, Sedaghat-Hamedani F, Amr A, Haas J, Katus H, Meder B et al (2016) A self-taught artificial agent for multi-physics computational model personalization. *Med Image Anal* 34:52–64
- Panthee N, Okada J-I, Washio T, Mochizuki Y, Suzuki R, Koyama H, Ono M, Hisada T, Sugiura S (2016) Tailor-made heart simulation predicts the effect of cardiac resynchronization therapy in a canine model of heart failure. *Med Image Anal* 31:46–62
- Peherstorfer B, Willcox K, Gunzburger M (2016) Survey of multifidelity methods in uncertainty propagation, inference, and optimization. Technical Report TR-16-1, Aerospace Computational Design Laboratory, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge
- Powell MJ (2009) The bobyqa algorithm for bound constrained optimization without derivatives. Cambridge NA Report NA2009/06, University of Cambridge, Cambridge
- Schaerer J, Qian Z, Clarysse P, Metaxas D, Axel L, Magnin IE (2006) Fast and automated creation of patient-specific 3d heart model from tagged MRI. In: *Proceedings of the MICCAI 2006 SA2PM workshop*
- Seegerer P, Mansi T, Jolly M-P, Neumann D, Georgescu B, Kamen A, Kayvanpour E, Amr A, Sedaghat-Hamedani F, Haas J, et al. (2015) Estimation of regional electrical properties of the heart from 12-lead ECG and images, vol 8896 of LNCS
- Sermesant M, Chabiniok R, Chinchapatnam P, Mansi T, Billet F, Moireau P, Peyrat J-M, Wong KC, Relan J, Rhode KS, Ginks M, Lambiase P, Delingette H, Sorine M, Rinaldi CA, Chapelle D, Razavi R, Ayache N (2012) Patient-specific electromechanical models of the heart for prediction of the acute effects of pacing in CRT: a first validation. *Med Image Anal* 16(1):201–215
- Smith N, de Vecchi A, McCormick M, Nordsletten D, Camara O, Frangi AF, Delingette H, Sermesant M, Relan J, Ayache N, Krueger MW, Schulze WHW, Hose R, Valverde I, Beerbaum P, Staicu C, Siebes M, Spaan J, Hunter P, Weese J, Lehmann H, Chapelle D, Rezavi R (2011) euheart: personalized and integrated cardiac care using patient-specific cardiovascular modelling. *Interface Focus* 1(3):349–364
- Streeter DD (1979) Gross morphology and fiber geometry of the heart. *Handbook of physiology*. Williams & Wilkins, Baltimore, pp 61–112
- Wang VY, Hoogendoorn C, Frangi AF, Cowan BR, Hunter PJ, Young AA, Nash MP (2012) Automated personalised human left ventricular fe models to investigate heart failure mechanics. *International workshop on statistical atlases and computational models of the heart*. Springer, Berlin, pp 307–316
- Wang Y, Georgescu B, Chen T, Wu W, Wang P, Lu X, Ionasec R, Zheng Y, Comaniciu D (2013) Learning-based detection and tracking in medical imaging: a probabilistic approach. *Deformation Models*. Springer, Berlin, pp 209–235
- Westerhof N, Bosman F, De Vries CJ, Noordergraaf A (1969) Analog studies of the human systemic arterial tree. *J Biomech* 2(2):121–208
- Xi J, Lamata P, Lee J, Moireau P, Chapelle D, Smith N (2011) Myocardial transversely isotropic material parameter estimation from in-silico measurements based on a reduced-order unscented kalman filter. *J Mech Behav Biomed Mater* 4(7):1090–1102
- Yu T, Lloyd CM, Nickerson DP, Cooling MT, Miller AK, Garny A, Terkildsen JR, Lawson J, Britten RD, Hunter PJ, Nielsen PMF (2011) The physiome model repository 2. *Bioinformatics* 27(5):743