Preserving the positivity of the deformation gradient determinant in intergrid interpolation by combining RBFs and SVD: Application to cardiac electromechanics

Michele Bucelli[a], Francesco Regazzoni[a], Luca Dede'[a,*], Alfio Quarteroni[a,b]

[a] *MOX, Laboratory of Modeling and Scientific Computing, Dipartimento di Matematica, Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133, Milano, Italy*
[b] *Institute of Mathematics, École Polytechnique Fédérale de Lausanne, Station 8, Av. Piccard, CH-1015, Lausanne, Switzerland[1]*

Abstract

The accurate, robust and efficient transfer of the deformation gradient tensor between meshes of different resolution is crucial in cardiac electromechanics simulations. This paper presents a novel method that combines rescaled localized Radial Basis Function (RBF) interpolation with Singular Value Decomposition (SVD) to preserve the positivity of the determinant of the deformation gradient tensor. The method involves decomposing the evaluations of the tensor at the quadrature nodes of the source mesh into rotation matrices and diagonal matrices of singular values; computing the RBF interpolation of the quaternion representation of rotation matrices and the singular value logarithms; reassembling the deformation gradient tensors at quadrature nodes of the destination mesh, to be used in the assembly of the electrophysiology model equations. The proposed method overcomes limitations of existing interpolation methods, including nested intergrid interpolation and RBF interpolation of the displacement field, that may lead to the loss of physical meaningfulness of the mathematical formulation and then to solver failures at the algebraic level, due to negative determinant values. Furthermore, the proposed method enables the transfer of solution variables between finite element spaces of different degrees and shapes and without stringent conformity requirements between different meshes, thus enhancing the flexibility and accuracy of electromechanical simulations. We show numerical results confirming that the proposed method enables the transfer of the deformation gradient tensor, allowing to successfully run simulations in cases where existing methods fail. This work provides an efficient and robust method for the intergrid transfer of the deformation gradient tensor, thus enabling independent tailoring of mesh discretizations to the particular characteristics of the individual physical components concurring to the of the multiphysics model.

1. Introduction

The multiple physical models involved in the mathematical representation of cardiac electromechanics [1–14] and electro-fluid-mechanics [15–21] are characterized by very different spatial and temporal scales. In particular, cardiac

---

* Corresponding author.
  E-mail address: luca.dede@polimi.it (L. Dede').
[1] Emeritus Professor.

electrophysiology [22–26] features very fast transients and sharp propagating fronts [27,28], requiring, upon finite element discretization, very fine computational meshes to be captured, with a typical size for the mesh elements of around 0.3 mm [29–31]. Conversely, cardiac mechanics does not require such a fine discretization [2,4]. Solving electrophysiology and mechanics using the same spatial resolution (dictated by the accuracy requirements of the former) entails an excessive computational cost. For this reason, it can be computationally convenient to solve the two problems using different spatial discretizations (a fine one for electrophysiology and a coarse one for mechanics), relying on suitable intergrid operators to transfer variables between the two models [4,32]. The intergrid operators must primarily transfer the intracellular calcium concentration from the electrophysiology model to the active mechanics one. Moreover, when considering mechano-electric feedback effects [33–36], the deformation gradient tensor should also be transferred from the mechanics model back to the electrophysiology one.

A simple approach to intergrid interpolation is based on using the same mesh for electrophysiology and mechanics, but using higher order polynomials for electrophysiology, as done e.g. in [15,37], where quadratic finite elements are used for electrophysiology and linear elements are used for solid mechanics. Higher polynomial orders may require the introduction of appropriate high-order methods [27,28,38], which in turn may call for significant changes in existing computational pipelines.

Alternatively, the computational framework presented in [2,4] relies on interpolation between nested meshes to solve cardiac mechanics on a coarse hexahedral mesh, and electrophysiology on a finer mesh obtained by subdividing the coarse one. That approach relies on an octree implementation of the mesh data structures [39,40], which makes the interpolation efficient for quadrilateral and hexahedral discretizations. Nested meshes, however, pose significant restrictions. For example, if the solid mechanics mesh is locally refined to capture geometrical features or material discontinuities, such inhomogeneities are inherited by the electrophysiology mesh, even though a non-constant mesh resolution may lead to artificial spatial variations in conduction velocity [30,41]. Moreover, the geometrical detail captured by the fine mesh is limited by that of the coarse mesh, partially countering the advantages of mesh refinement.

A more flexible approach is offered by RBF interpolation [32,42–44]. In this case, the two meshes can be independent, both geometrically and parametrically. RBF interpolation was applied to cardiac electromechanics in [32], showing that the interpolation allows for an accurate segregated electromechanical solver significantly faster than its monolithic counterpart [45].

The interpolation of the deformation gradient $\mathbf{F} = \mathbf{I} + \nabla \mathbf{d}$, with $\mathbf{d}$ being the tissue displacement field, from the coarse to the fine mesh requires special care. Indeed, if mechano-electrical feedbacks are included in the model [33–36], the electrophysiology equations involve the inverse of the deformation gradient $\mathbf{F}$ and its determinant J. For the problem to be well posed and physically meaningful, J should be positive on the whole domain. However, naive tensor interpolation methods cannot guarantee that this is verified [46].

In this work, we combine rescaled localized RBF interpolation [43] with singular value decomposition (SVD), to obtain an interpolation method for the tensor $\mathbf{F}$ that preserves the sign of its determinant, in an approach similar to the one presented in [46]. The interpolation method is applied to ventricular electromechanical simulations, showing that it improves the robustness of the method over simpler techniques. We compare the newly introduced method against alternative interpolation methods [4,32] in terms of numerical accuracy and computational costs.

Our results show that the proposed method allows to successfully perform simulations where previously introduced methods would fail. Indeed, previous methods yield negative values for the deformation gradient determinant J after interpolation, whereas the newly proposed one guarantees by construction its positivity. This avoids the appearance of highly unphysical values of J, and makes sure that the discrete problem remains well posed throughout the simulation. Furthermore, we demonstrate that the computational cost of interpolation remains negligible with respect to the overall cost of the simulation, and is linearly scalable in a parallel computing framework.

The rest of the paper is structured as follows. Section 2 briefly reviews the models used for cardiac electromechanics. In Section 3, we recall the rescaled localized RBF interpolation method, and describe the interpolation procedure based on SVD. Section 4 describes the discretization strategy used for the electromechanical model, and Section 5 presents some numerical experiments highlighting the properties of the proposed method. Finally, in Section 6, we draw some conclusive remarks.

2. Electromechanical modeling of the heart

The sketch of the electromechanics model used in this work is displayed in Fig. 1. We refer the interested reader to [4] for further details on the model and its numerical solution.
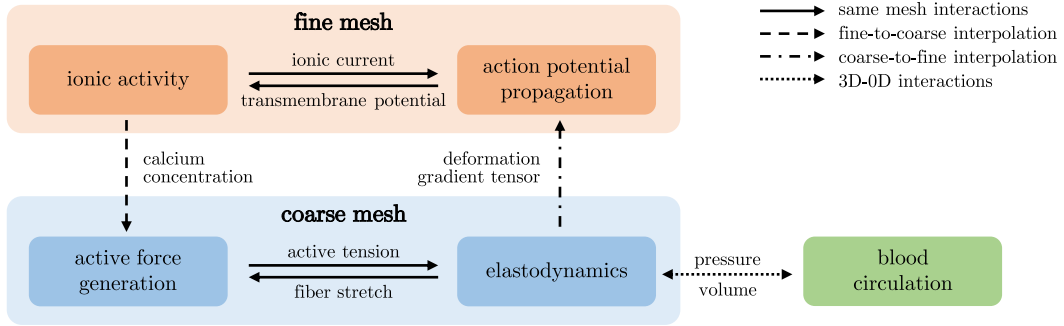
Fig. 1. Cardiac electromechanical model and its subsystems. The large shaded areas enclose models that are discretized using the same computational mesh. Arrows denote the variables that realize the coupling between the subsystems. Solid lines correspond to quantities that are defined on the same computational mesh; dashed and dot-dashed lines correspond to quantities that need to be transferred from one mesh to another; the dotted line corresponds to quantities associated with the coupling between the 3D electromechanical model and the 0D model (i.e. lumped-parameter model) of blood circulation.
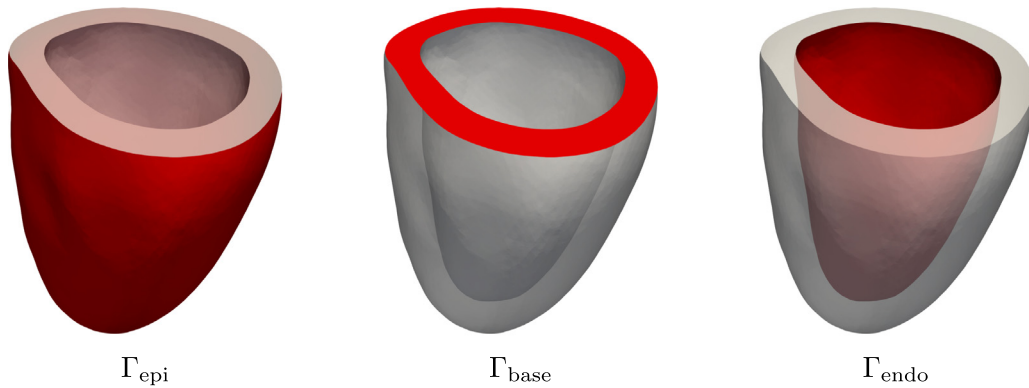

Fig. 2. Computational domain Ω and its boundaries Γ_{epi}, Γ_{base} and Γ_{endo}.

Let $\Omega \subset \mathbb{R}^3$ be an open, bounded domain representing a left ventricle (shown in Fig. 2), and let $T > 0$. We consider a coupled problem involving cardiac electrophysiology, active force generation, muscular mechanics and the circulatory system, with the following unknowns:

$$v : \Omega \times (0, T) \to \mathbb{R} \qquad\qquad \text{transmembrane potential ,} \qquad\qquad (1)$$

$$\mathbf{w} : \Omega \times (0, T) \to \mathbb{R}^{N_{\text{ion}}} \qquad\qquad \text{ionic variables ,} \qquad\qquad (2)$$

$$\mathbf{s} : \Omega \times (0, T) \to \mathbb{R}^{N_{\text{act}}} \qquad\qquad \text{activation variables ,} \qquad\qquad (3)$$

$$\mathbf{d} : \Omega \times (0, T) \to \mathbb{R}^3 \qquad\qquad \text{solid displacement ,} \qquad\qquad (4)$$

$$\mathbf{c} : (0, T) \to \mathbb{R}^{N_{\text{circ}}} \qquad\qquad \text{circulation state variables .} \qquad\qquad (5)$$

We model the evolution of the transmembrane potential v with the monodomain equation [22,24], including geometry-mediated mechano-electrical feedback effects [33–35]:

$$
\begin{cases}
J \chi C_m \frac{\partial v}{\partial t} - \nabla \cdot \left(J \mathbf{F}^{-1} \mathbf{D}_{\text{m}} \mathbf{F}^{-T} \nabla v \right) + J \chi I_{\text{ion}}(v, \mathbf{w}) = J \chi I_{\text{app}} & \text{in } \Omega \times (0, T) , \\
J \mathbf{F}^{-1} \mathbf{D}_{\text{m}} \mathbf{F}^{-T} \nabla v \cdot \mathbf{n} = 0 & \text{on } \partial\Omega \times (0, T) , \\
v = v_0 & \text{in } \Omega \times \{0\} .
\end{cases}
\qquad (6)
$$

In the above, $\mathbf{F} = \mathbf{I} + \nabla \mathbf{d}$ is the deformation gradient associated with the displacement of the cardiac muscle, and $J = \det \mathbf{F}$ is its determinant. Their presence in (6) accounts for the effect of the deformation onto the propagation of the electrical activation. The tensor $\mathbf{D}_{\text{m}}$ accounts for the anisotropic conductivity of the cardiac muscle, and it is

defined as

$$\mathbf{D}_\mathrm{m} = \sigma_\mathrm{f} \frac{\mathbf{F}\mathbf{f}_0 \otimes \mathbf{F}\mathbf{f}_0}{\|\mathbf{F}\mathbf{f}_0\|^2} + \sigma_\mathrm{s} \frac{\mathbf{F}\mathbf{s}_0 \otimes \mathbf{F}\mathbf{s}_0}{\|\mathbf{F}\mathbf{s}_0\|^2} + \sigma_\mathrm{n} \frac{\mathbf{F}\mathbf{n}_0 \otimes \mathbf{F}\mathbf{n}_0}{\|\mathbf{F}\mathbf{n}_0\|^2} \ , \tag{7}$$

wherein $\{\mathbf{f}_0, \mathbf{s}_0, \mathbf{n}_0\}$ is a space-dependent orthonormal triplet that represents the local direction of fibers, fiber sheets and cross-fibers, respectively [23,47,48], and σ_f, σ_s and σ_n are constant conductivity coefficients in the three directions. The coefficients χ and C_m in (6) are the membrane surface-to-volume ratio and capacitance, respectively, while the space- and time-dependent function I_app is a forcing term used to provide an initial stimulus to trigger the electrical activation of the ventricle [4].

The evolution of ionic variables is prescribed by a suitable ionic model, in the general form

$$\begin{cases} \frac{\partial \mathbf{w}}{\partial t} = \mathbf{F}_\mathrm{ion}(v, \mathbf{w}) & \text{in } \Omega \times (0, T) \ , \\ \mathbf{w} = \mathbf{w}_0 & \text{in } \Omega \times \{0\} \ . \end{cases} \tag{8}$$

We consider the ventricular ionic model by ten Tusscher and Panfilov [49].

Remark 1. Although not considered in this work, the system (6) can be endowed with additional reactive terms that account for other sources of mechano-electrical feedbacks, including e.g. stretch-activated currents [34]. Although they have a minor effect in sinus rhythm, they may become extremely relevant in pathological scenarios involving arrhythmias [34], so that an accurate evaluation of $\mathbf{F}$ can become extremely important.

One of the ionic variables $\mathbf{w}$ represents the intracellular calcium concentration $[\mathrm{Ca}^{2+}]_\mathrm{i}$, which is used as input to the force generation model describing the evolution of the contraction state $\mathbf{s}$. To this end, we consider the RDQ18 [50], that can be expressed as a system of ordinary differential equations (ODEs):

$$\begin{cases} \frac{\partial \mathbf{s}}{\partial t} = \mathbf{F}_\mathrm{act} \left(\mathbf{s}, [\mathrm{Ca}^{2+}]_\mathrm{i}, \mathrm{SL} \right) & \text{in } \Omega \times (0, T) \ , \\ \mathbf{s} = \mathbf{s}_0 & \text{in } \Omega \times \{0\} \ . \end{cases} \tag{9}$$

In the above, SL is the sarcomere length, defined as $\mathrm{SL} = \mathrm{SL}_0 \sqrt{I_\mathrm{4f}}$, where $I_\mathrm{4f} = \mathbf{F}\mathbf{f}_0 \cdot \mathbf{F}\mathbf{f}_0$. The contraction state is then used to compute an active stress tensor through

$$\mathbf{P}_\mathrm{act}(\mathbf{d}, \mathbf{s}) = T_\mathrm{act}^\mathrm{max} \, P_\mathrm{act}(\mathbf{s}) \frac{\mathbf{F}\mathbf{f}_0 \otimes \mathbf{F}\mathbf{f}_0}{\sqrt{I_\mathrm{4f}}} \ , \tag{10}$$

where $T_\mathrm{act}^\mathrm{max}$ is the maximum active tension and $P_\mathrm{act}(\mathbf{s}) \in [0, 1]$ is the permissivity. We refer to [50] for the precise definition of $\mathbf{F}_\mathrm{act}$ and P_act. Due to the complexity of the model (which features $N_\mathrm{act} = 2176$ state variables for each mesh node), we consider a surrogate version obtained through artificial neural networks, featuring only $N_\mathrm{act} = 2$ state variables, as presented in [51,52].

The displacement of the muscle is modeled by the elastodynamics equation in the hyperelastic framework:

$$\begin{cases} \rho \frac{\partial^2 \mathbf{d}}{\partial t^2} - \nabla \cdot \mathbf{P}(\mathbf{d}, \mathbf{s}) = \mathbf{0} & \text{in } \Omega \times (0, T) \ , \\ \mathbf{P}(\mathbf{d}, \mathbf{s})\mathbf{n} + \mathbf{K}_\mathrm{epi}\mathbf{d} + \mathbf{C}_\mathrm{epi} \frac{\partial \mathbf{d}}{\partial t} = \mathbf{0} & \text{on } \Gamma_\mathrm{epi} \times (0, T) \ , \\ \mathbf{P}(\mathbf{d}, \mathbf{s})\mathbf{n} = p_\mathrm{LV}(t) |J\mathbf{F}^{-T}\mathbf{n}| \frac{\int_{\Gamma_\mathrm{endo}} J\mathbf{F}^{-T}\mathbf{n}\, d\Gamma}{\int_{\Gamma_\mathrm{base}} |J\mathbf{F}^{-T}\mathbf{n}|\, d\Gamma} & \text{on } \Gamma_\mathrm{base} \times (0, T) \ , \\ \mathbf{P}(\mathbf{d}, \mathbf{s})\mathbf{n} = -p_\mathrm{LV}(t) J\mathbf{F}^{-T}\mathbf{n} & \text{on } \Gamma_\mathrm{endo} \times (0, T) \ , \end{cases} \tag{11}$$

where $\mathbf{P}(\mathbf{d}, \mathbf{s}) = \mathbf{P}_\mathrm{act}(\mathbf{d}, \mathbf{s}) + \mathbf{P}_\mathrm{pas}(\mathbf{d})$ is the first Piola–Kirchhoff stress tensor, decomposed into the sum of the active stress $\mathbf{P}_\mathrm{act}$ and the passive stress $\mathbf{P}_\mathrm{pas}$. The latter is obtained as the derivative of a suitable strain energy density function representing the constitutive law of the solid:

$$\mathbf{P}_\mathrm{pas} = \frac{\partial \mathcal{W}}{\partial \mathbf{F}} \ . \tag{12}$$

We consider the orthotropic constitutive law of [53], with an additional penalization term imposing near-incompressibility as presented in [4], by setting

$$\mathcal{W} = \frac{C}{2} \left(e^Q - 1 \right) + \frac{B}{2} (J - 1) \log J \ , \tag{13}$$

$$Q = b_\mathrm{ff} E_\mathrm{ff}^2 + b_\mathrm{ss} E_\mathrm{ss}^2 + b_\mathrm{nn} E_\mathrm{nn}^2 + b_\mathrm{fs} \left(E_\mathrm{fs}^2 + E_\mathrm{sf}^2 \right) + b_\mathrm{fn} \left(E_\mathrm{fn}^2 + E_\mathrm{nf}^2 \right) + b_\mathrm{sn} \left(E_\mathrm{sn}^2 + E_\mathrm{ns}^2 \right) \ , \tag{14}$$

with $E_{ab} = \mathbf{E}\mathbf{a}_0 \cdot \mathbf{b}_0$, for a, b $\in$ {f, s, n}, and $\mathbf{E} = \frac{1}{2}\left(\mathbf{F}^T\mathbf{F} - \mathbf{I}\right)$. We remark that the discussion that follows generalizes to cases where different constitutive laws are used, for instance the one proposed in [54]. Γ_{epi}, Γ_{base} and Γ_{endo} are subsets of $\partial\Omega$, shown in Fig. 2, and $\mathbf{n}$ is the outward-directed normal unit vector. The boundary condition on Γ_{epi} expresses the interaction of the heart with the pericardium and the surrounding tissue [6,55]. The tensors $\mathbf{K}_{\text{epi}}$ and $\mathbf{C}_{\text{epi}}$ are defined as

$$\mathbf{K}_{\text{epi}} = K_{\text{epi}}^{\perp}(\mathbf{n} \otimes \mathbf{n}) + K_{\text{epi}}^{\parallel}(\mathbf{I} - \mathbf{n} \otimes \mathbf{n}) , \tag{15}$$

$$\mathbf{C}_{\text{epi}} = C_{\text{epi}}^{\perp}(\mathbf{n} \otimes \mathbf{n}) + C_{\text{epi}}^{\parallel}(\mathbf{I} - \mathbf{n} \otimes \mathbf{n}) . \tag{16}$$

The condition on Γ_{base} is known as energy-consistent boundary condition, and it surrogates the portion of ventricle that has been cut from the computational domain at the base [4,52].

The intracardiac pressure p_{LV} is obtained by solving a lumped-parameter model for the circulatory system [4,56,57]. The model can be expressed as a system of algebraic–differential equations:

$$\begin{cases} \mathbf{F}_{\text{circ}}\left(\frac{d\mathbf{c}}{dt}, \mathbf{c}, t\right) = \mathbf{0} & \text{in } (0, T) , \\ \mathbf{c}(0) = \mathbf{c}_0 . \end{cases} \tag{17}$$

Two of the variables of the circulation state $\mathbf{c}$ are the left ventricular pressure p_{LV} and the left ventricular volume V_{LV}. The coupling of the mechanics Eqs. (11) and of the circulation system (17) is obtained by imposing the pressure condition on Γ_{endo} in (11), together with the constraint $V_{\text{LV}} = V_{\text{LV}}^{\text{3D}}$, where $V_{\text{LV}}^{\text{3D}}$ is the volume enclosed by Γ_{endo} in the deformed configuration, computed as described in [4].

3. Intergrid interpolation for electromechanics

In the following sections, we recall the RBF interpolation method, and present the algorithm used in this work for interpolating the deformation gradient $\mathbf{F}$ from the mesh of the mechanics problem to that of the electrophysiology problem.

3.1. Rescaled localized radial basis function interpolation

Let $\{\mathbf{x}_i^{\text{src}}\}_{i=1}^{N^{\text{src}}}$, with $\mathbf{x}_i^{\text{src}} \in \mathbb{R}^3$, be a set of distinct points. Let $f : \mathbb{R}^3 \to \mathbb{R}$ be a function, and let $f_i^{\text{src}} = f(\mathbf{x}_i^{\text{src}})$. The RBF interpolant of f at the points $\mathbf{x}_i^{\text{src}}$ is a function $\Pi_f : \mathbb{R}^3 \to \mathbb{R}$ in the form

$$\Pi_f(\mathbf{x}) = \sum_{j=1}^{N^{\text{src}}} \gamma_j \, \phi\left(\|\mathbf{x} - \mathbf{x}_j^{\text{src}}\|, r_j\right) , \tag{18}$$

where γ_j are the interpolation coefficients, ϕ is the RBF and r_j is the RBF support radius associated with each point $\mathbf{x}_j^{\text{src}}$. Following [32,43], we consider the compactly supported C^2 Wendland basis function [58], shown in Fig. 3(a) and defined as

$$\phi(t, r) = \max\left\{1 - \frac{t}{r}, 0\right\}^4 \left(1 + 4\frac{t}{r}\right) . \tag{19}$$

The coefficients γ_i in (18) are determined by imposing the interpolation condition $\Pi_f(\mathbf{x}_i^{\text{src}}) = f_i^{\text{src}}$ for all $i = 1, 2, \ldots, N^{\text{src}}$. This gives rise to the linear system

$$\Phi_{\text{int}}\boldsymbol{\gamma} = \mathbf{f}^{\text{src}} , \tag{20}$$

where $\Phi_{\text{int}} \in \mathbb{R}^{N^{\text{src}} \times N^{\text{src}}}$ is a matrix whose entries are

$$(\Phi_{\text{int}})_{ij} = \phi(\|\mathbf{x}_i^{\text{src}} - \mathbf{x}_j^{\text{src}}\|, r_j) , \tag{21}$$

with $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \ldots, \gamma_{N^{\text{src}}})^T \in \mathbb{R}^{N^{\text{src}}}$ and $\mathbf{f}^{\text{src}} = (f_1^{\text{src}}, f_2^{\text{src}}, \ldots, f_{N^{\text{src}}}^{\text{src}})^T \in \mathbb{R}^{N^{\text{src}}}$.

Assume now that the interpolant must be evaluated on a new set of points $\{\mathbf{x}_i^{\text{dst}}\}_{i=1}^{N^{\text{dst}}}$. Let $f_i^{\text{dst}} = \Pi_f(\mathbf{x}_i^{\text{dst}})$. There holds, for all $i = 1, 2, \ldots, N^{\text{dst}}$:

$$f_i^{\text{dst}} = \sum_{j=1}^{N^{\text{src}}} \gamma_j \phi(\|\mathbf{x}_i^{\text{dst}} - \mathbf{x}_j^{\text{src}}\|, r_j) , \tag{22}$$
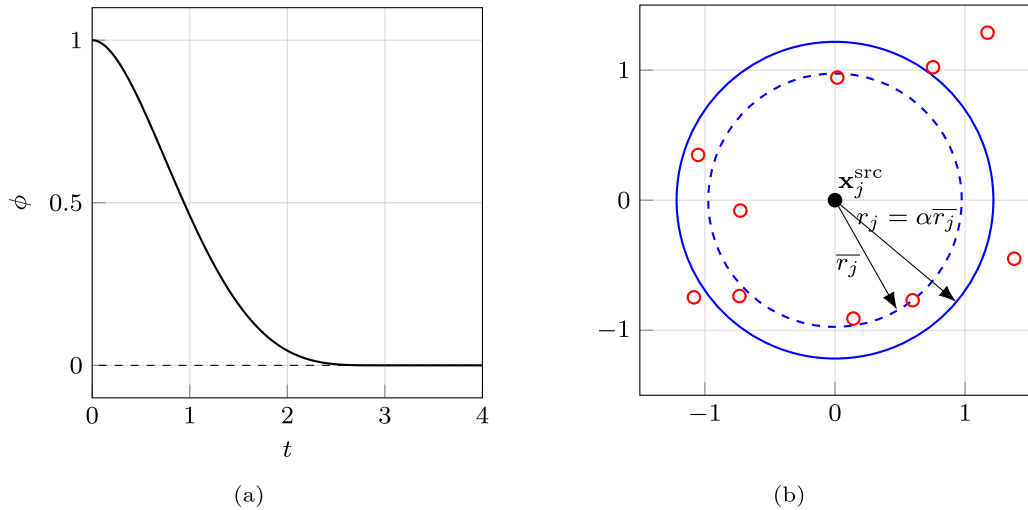
(a)   (b)

Fig. 3. (a) Plot of the C^2 Wendland basis function $\phi(t, 3)$. (b) Selection of the adaptive RBF radius r_j for point $\mathbf{x}_j^{\mathrm{src}}$. The radius is such that the support encloses $M = 4$ interpolation points (indicated by the red void circles), with $\alpha = 1.2$ (see (25)). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

which can be expressed compactly as

$$\mathbf{f}^{\mathrm{dst}} = \varPhi_{\mathrm{eval}} \boldsymbol{\gamma} \,, \tag{23}$$

where $\varPhi_{\mathrm{eval}} \in \mathbb{R}^{N^{\mathrm{dst}} \times N^{\mathrm{src}}}$ is a matrix whose entries are

$$(\varPhi_{\mathrm{eval}})_{ij} = \phi(\|\mathbf{x}_i^{\mathrm{dst}} - \mathbf{x}_j^{\mathrm{src}}\|, r_j) \,, \tag{24}$$

and $\mathbf{f}^{\mathrm{dst}} = (f_1^{\mathrm{dst}}, f_2^{\mathrm{dst}}, \ldots, f_{N^{\mathrm{dst}}}^{\mathrm{dst}})^T \in \mathbb{R}^{N^{\mathrm{dst}}}$.

The interpolation radii r_j are selected adaptively. Intuitively, the radius should be smaller in regions of space where the interpolation points $\mathbf{x}_j^{\mathrm{src}}$ are more densely clustered. In [43], the authors select the radius based on the connectivity of the mesh where the interpolated data is defined. In a parallel computing context, where the mesh is distributed across several processors, the connectivity may not be easily accessible. Moreover, this notion does not generalize to the case where the interpolation and evaluation points are not the nodes of a mesh (as considered in Section 3.3.2). For this reason, we follow an approach similar to the one used in [44], and choose

$$r_j = \alpha \overline{r_j} \,, \tag{25}$$

where $\overline{r_j}$ is the smallest value such that the sphere centered at $\mathbf{x}_j^{\mathrm{src}}$ with radius $\overline{r_j}$ contains at least M other interpolation points. This procedure is represented in Fig. 3(b).

As discussed in [43], the RBF interpolation procedure described above may yield large oscillations in the interpolant, and is very sensitive to the choice of the interpolation radius. To avoid this issue, we consider the following rescaled interpolant, introduced in [43]:

$$\varPi_f^{\mathrm{res}}(\mathbf{x}) = \frac{\varPi_f(\mathbf{x})}{\varPi_g(\mathbf{x})} \,, \tag{26}$$

where $\varPi_g$ is the RBF interpolant of the constant function $g(\mathbf{x}) = 1$ at the nodes $\mathbf{x}_i^{\mathrm{src}}$. All the results presented in this paper make use of the rescaled interpolant.

The interpolation of vector fields is obtained by separately interpolating each component.

3.2. Preconditioning of the interpolation system by means of approximate cardinal functions

The matrix $\varPhi_{\mathrm{int}}$ is sparse due to the RBF ϕ having compact support. We solve system (20) by means of the preconditioned GMRES method [59]. The choice of a suitable preconditioner is crucial towards the efficient construction of the interpolant.

Following [60–62], we consider a preconditioner based on approximated cardinal functions. The procedure to assemble it is described in Algorithm 1. The idea behind the preconditioner is to construct, for each point $\mathbf{x}_i^{\mathrm{src}}$, an approximate cardinal function ψ_i, such that

$$\psi_i(\mathbf{x}) = \sum_{j \in S^i} \lambda_j^i \, \phi(\|\mathbf{x}_j^{\mathrm{src}} - \mathbf{x}\|, r_j) \, , \tag{27}$$

$$\psi_i(\mathbf{x}_i^{\mathrm{src}}) = 1 \, , \tag{28}$$

$$\psi_i(\mathbf{x}_j^{\mathrm{src}}) = 0 \quad \text{for } j \in S^i, \; j \neq i \, , \tag{29}$$

where S^i is a set of points that are sufficiently close to $\mathbf{x}_i^{\mathrm{src}}$. Finding the cardinal function is itself an interpolation problem over the points specified by S^i. The preconditioner matrix $P^{-1} \in \mathbb{R}^{N^{\mathrm{src}} \times N^{\mathrm{src}}}$ has entries

$$\left(P^{-1}\right)_{ji} = \begin{cases} \lambda_j^i & \text{if } j \in S^i \, , \\ 0 & \text{otherwise.} \end{cases} \tag{30}$$

We select S^i to be the set of indices j such that $(\Phi_{\mathrm{int}})_{ij} \neq 0$. We refer to [60–62] for further considerations on this preconditioning strategy.

Algorithm 1 Construction of the preconditioner P for system (20).

Input: interpolation matrix Φ_{int}
Output: preconditioner matrix P

1: initialize $P = 0$
2: **for** $i = 0, 1, \ldots, N^{\mathrm{src}}$ **do**
3:     $S^i \leftarrow \{j \in \mathbb{N} : (\Phi_{\mathrm{int}})_{ij} \neq 0\}$, and let s_m^i be its elements, with $m = 0, 1, \ldots, n_i$
4:
5:     compute the matrix $L^i \in \mathbb{R}^{n_i \times n_i}$, with entries $L_{lm}^i = \phi(\|\mathbf{x}_{s_l}^{\mathrm{src}} - \mathbf{x}_{s_m}^{\mathrm{src}}\|, r_{s_m})$
6:     compute the vector $\mathbf{r}^i \in \mathbb{R}^{n_i}$, with entries $\mathbf{r}_l^i = \delta_{i s_l^i}$
7:     solve the system $L^i \boldsymbol{\lambda}^i = \mathbf{r}^i$ with the GMRES method
8:
9:     **for** $m = 0, 1, \ldots, n_i$ **do**
10:       $\left(P^{-1}\right)_{s_m^i, i} \leftarrow \boldsymbol{\lambda}_m^i$
11:     **end for**
12: **end for**

The construction of the preconditioner can be very costly, requiring the solution of a small yet dense linear system for each point $\mathbf{x}_i^{\mathrm{src}}$. However, if the interpolant must be constructed and evaluated several times, the computational saving in the repeated solution of (20) amortizes the cost of the preconditioner assembly.

We use a Gauss–Seidel preconditioner to accelerate the convergence of the GMRES method for the linear system $L^i \boldsymbol{\lambda}^i = \mathbf{r}^i$ in Algorithm 1. Due to the approximate nature of the preconditioner, the interpolation problem to compute ψ_i can be solved to low accuracy without significantly damaging the effectiveness of P. In all the tests described in this paper, we use a relative tolerance of 10^{-1}.

Remark 2. With respect to the strategy outlined in [60], our choice of S^i allows to compute P by efficiently reusing data structures and quantities already computed for the construction of Φ_{int}. We remark that this choice is not the only possible one, and alternative options may result in a better preconditioner, at the price of additional computational overhead for its initialization.

3.3. RBF interpolation in the finite element framework

In the context of finite elements, let us consider a domain $\Omega \subset \mathbb{R}^3$, and two independent meshes approximating Ω, $\mathcal{T}^{\mathrm{src}}$ and $\mathcal{T}^{\mathrm{dst}}$. Each of the two meshes can be composed of tetrahedral or hexahedral elements (not necessarily the same on both meshes). On the two meshes, we consider the finite element spaces V^{src} and V^{dst}, composed of

piecewise polynomials of degree p^{src} and p^{dst}, respectively, spanned by suitable interpolatory basis functions (such as the Lagrangian basis functions [63,64]) on meshes $\mathcal{T}^{\text{src}}$ and $\mathcal{T}^{\text{dst}}$.

We consider two different interpolation strategies: interpolation between degrees of freedom (DoFs) and interpolation between quadrature nodes. Alternative strategies (e.g. interpolation from DoFs to quadrature nodes, or vice versa) can also be considered, although they are not relevant for our target application.

3.3.1. Interpolation between degrees of freedom

The points $\mathbf{x}_i^{\text{src}}$ are the support points of the DoFs on $\mathcal{T}^{\text{src}}$, and the points $\mathbf{x}_i^{\text{dst}}$ are those on $\mathcal{T}^{\text{dst}}$. If f is a function belonging to the finite element space V^{src}, then the vector $\mathbf{f}^{\text{src}}$ is the vector of its control variables. Similarly, $\mathbf{f}^{\text{dst}}$ is the vector of control variables of the finite element interpolation of Π_f^{res} in V^{dst}. Therefore, the computation of the interpolant as described in Section 3.1 allows to easily interpolate from the finite element space V^{src} onto the space V^{dst}. This is the approach we follow to interpolate the intracellular calcium concentration $[\text{Ca}^{2+}]_i$ from the mesh used for electrophysiology onto the one used for mechanics.

3.3.2. Interpolation between quadrature nodes

In some instances, one may need to interpolate a function that is not well defined at the DoFs of $\mathcal{T}^{\text{src}}$. That is the case of the interpolation of $\mathbf{F}$ from the mechanics to the electrophysiology mesh. Indeed, since the finite element space is globally continuous but only piecewise differentiable, $\mathbf{F}$ is not well defined at those DoFs that lie on the boundary of a mesh element.

In that case, we choose the points $\mathbf{x}_i^{\text{src}}$ to be internal to the mesh elements, by considering suitable Gaussian quadrature nodes for each element. We consider a Gaussian quadrature formula with q quadrature nodes in each coordinate direction.

As typical in finite elements, we approximate the integrals arising from the Galerkin formulation of (6) by using Gaussian quadrature. Therefore, we need to evaluate $\mathbf{F}$ on the quadrature nodes of the elements of $\mathcal{T}^{\text{dst}}$, and it is convenient to select $\mathbf{x}_i^{\text{dst}}$ to be those quadrature nodes.

Remark 3. The source points $\mathbf{x}_i^{\text{src}}$ do not need to be nodes of a quadrature formula, and any points lying on the interior of elements can be used. However, Gaussian quadrature points are usually convenient from an implementation viewpoint, since they can be easily computed in any finite element software, and the evaluation of $\mathbf{F}$ at those point is easily accessible.

3.4. Interpolation of the deformation gradient

For (6) to be well-posed after discretization, the interpolation of $\mathbf{F}$ should be done in a way that preserves the positivity of $J = \det \mathbf{F} > 0$: indeed, if negative values of J arise in (6), its numerical solution may diverge. However, the set of tensors with positive determinant is not a linear space (hence, linear combinations of positive-determinant tensors might have negative determinant), nor it is convex (not even convex combinations of positive-determinant tensors are guaranteed to preserve the sign of the determinant). This explains why a naive interpolation of the deformation gradient tensor might yield non-physical results.

In [32], the deformation gradient is evaluated at the quadrature points of the fine mesh by combining the interpolation between DoFs of the displacement with the Zienkiewicz–Zhu gradient recovery technique [65]. This approach, however, does not guarantee $J > 0$ after interpolation, and indeed we observed that in some situations it results in a breakdown of the numerical solver. To enforce $J > 0$, we combine RBF interpolation between quadrature nodes (Section 3.3.2) with SVD, in an approach similar to [46].

Let $\mathcal{T}^{\text{M}}$ be the mesh used for the mechanics problem, and $\mathcal{T}^{\text{EP}}$ the one used for the electrophysiology problem. Let $\mathbf{d}$ be the displacement field, defined on $\mathcal{T}^{\text{M}}$. We denote by $\mathbf{x}_i^{\text{M}}$, with $i = 0, 1, \ldots, N_q^{\text{M}}$, the quadrature nodes on mesh $\mathcal{T}^{\text{M}}$, and by $\mathbf{x}_j^{\text{EP}}$, with $j = 0, 1, \ldots, N_q^{\text{EP}}$, those on mesh $\mathcal{T}^{\text{EP}}$.

The main idea behind the interpolation procedure is to represent $\mathbf{F}$ at each point $\mathbf{x}_i^{\text{M}}$ as the composition of rotations and scalings, through SVD. We then interpolate the individual factors independently while preserving key properties: in particular, the sign of scalings is conserved, and we guarantee that rotations remain rotations after interpolation. The latter property is achieved by representing rotations through quaternions [66]. Finally, we recompose the tensor at the quadrature nodes $\mathbf{x}_j^{\text{EP}}$.

The procedure, reported in Algorithm 2, assumes that the following routines exist:

- `rotationToQuaternion`, to convert rotation matrices into their quaternion representation [66];
- `quaternionToRotation`, to convert a quaternion into its rotation matrix representation (normalizing it if necessary) [66];
- `SVD`, to compute the SVD of a given input matrix, with the alignment procedure of Section 3.5.

Algorithm 2 starts by computing the SVD factorization of the deformation gradient $\mathbf{F}_i = \mathbf{F}(\mathbf{x}_i^{\mathrm{M}})$ at each point $\mathbf{x}_i^{\mathrm{M}}$, thus expressing it as

$$\mathbf{F}_i = \mathbf{U}_i \mathbf{\Sigma}_i \mathbf{V}_i^T \, , \tag{31}$$

where $\mathbf{U}_i$ and $\mathbf{V}_i$ are rotation matrices, and $\mathbf{\Sigma}_i = \mathrm{diag}(\sigma_i^1, \sigma_i^2, \sigma_i^3)$ is a diagonal matrix whose diagonal entries are the singular values of $\mathbf{F}_i$. The matrices $\mathbf{U}_i$ and $\mathbf{V}_i$ are converted to the corresponding quaternions $\mathbf{q}_{\mathbf{U}_i}$ and $\mathbf{q}_{\mathbf{V}_i}$, respectively, and RBF interpolation is used to evaluate at the points $\mathbf{x}_j^{\mathrm{EP}}$ the resulting 11 scalar fields (the three singular values and the four quaternion components of $\mathbf{q}_{\mathbf{U}_i}$ and $\mathbf{q}_{\mathbf{V}_i}$). Then, the deformation gradient $\mathbf{F}^{\mathrm{EP}}$ is reconstructed at every point $\mathbf{x}_j^{\mathrm{EP}}$, by converting the quaternions back to rotation matrices and reassembling the SVD factors. To guarantee that the determinant of $\mathbf{F}^{\mathrm{EP}}$ remains positive, we interpolate the logarithm of the singular values, and then take their exponential after interpolation [46]. Indeed, there holds:

$$\det \mathbf{F}_j^{\mathrm{EP}} = \det \mathbf{\Sigma}_j^{\mathrm{EP}} = \exp\left(\Pi_{\log \sigma^1}^{\mathrm{res}}(\mathbf{x}_j^{\mathrm{EP}}) \right) \exp\left(\Pi_{\log \sigma^2}^{\mathrm{res}}(\mathbf{x}_j^{\mathrm{EP}}) \right) \exp\left(\Pi_{\log \sigma^3}^{\mathrm{res}}(\mathbf{x}_j^{\mathrm{EP}}) \right) > 0 \, , \tag{32}$$

wherein we exploit the fact that $\det \mathbf{U}_i^{\mathrm{EP}} = \det \mathbf{V}_i^{\mathrm{EP}} = 1$, as both of them are rotation matrices. We remark that this property holds independently of the compressibility of the material (i.e. regardless of the value of the bulk modulus B), and even in the incompressible limit (i.e. when $B \to \infty$).

The SVD of the tensor $\mathbf{F}_i$ is not unique. The procedure we follow to define a unique decomposition is described in Section 3.5.

Algorithm 2 Interpolation of the deformation gradient $\mathbf{F}$ from the mesh $\mathcal{T}^{\mathrm{M}}$ to the mesh $\mathcal{T}^{\mathrm{EP}}$.

Input: evaluations of $\nabla \mathbf{d}$ at quadrature nodes $\mathbf{x}_i^{\mathrm{M}}$
Output: evaluations of $\mathbf{F}^{\mathrm{EP}}$ at quadrature nodes $\mathbf{x}_j^{\mathrm{EP}}$

1: **for** $i = 0, 1, \ldots, N_{\mathrm{q}}^{\mathrm{M}}$ **do**
2:      $\mathbf{F}_i = \mathbf{F}(\mathbf{x}_i^{\mathrm{M}}) \leftarrow I + \nabla \mathbf{d}(\mathbf{x}_i^{\mathrm{M}})$
3:      $(\mathbf{U}_i, \mathbf{\Sigma}_i, \mathbf{V}_i^T) \leftarrow \mathtt{SVD}(\mathbf{F}_i)$ $\triangleright$ With the alignment procedure of Section Section 3.5.
4:      $(a_{\mathbf{U}_i}, b_{\mathbf{U}_i}, c_{\mathbf{U}_i}, d_{\mathbf{U}_i}) \leftarrow \mathtt{rotationToQuaternion}(\mathbf{U}_i)$
5:      $(a_{\mathbf{V}_i}, b_{\mathbf{V}_i}, c_{\mathbf{V}_i}, d_{\mathbf{V}_i}) \leftarrow \mathtt{rotationToQuaternion}(\mathbf{V}_i)$
6: **end for**
7:
8: build the interpolants $\Pi_{a_{\mathbf{U}}}^{\mathrm{res}}, \Pi_{b_{\mathbf{U}}}^{\mathrm{res}}, \Pi_{c_{\mathbf{U}}}^{\mathrm{res}}, \Pi_{d_{\mathbf{U}}}^{\mathrm{res}}, \Pi_{a_V}^{\mathrm{res}}, \Pi_{b_V}^{\mathrm{res}}, \Pi_{c_V}^{\mathrm{res}}, \Pi_{d_V}^{\mathrm{res}}, \Pi_{\log \sigma^1}^{\mathrm{res}}, \Pi_{\log \sigma^2}^{\mathrm{res}}, \Pi_{\log \sigma^3}^{\mathrm{res}}$
9:
10: **for** $j = 0, 1, \ldots, N_{\mathrm{q}}^{\mathrm{M}}$ **do**
11:      $\mathbf{U}_j^{\mathrm{EP}} \leftarrow \mathtt{quaternionToRotation}\left(\Pi_{a_{\mathbf{U}}}^{\mathrm{res}}(\mathbf{x}_j^{\mathrm{EP}}), \Pi_{b_{\mathbf{U}}}^{\mathrm{res}}(\mathbf{x}_j^{\mathrm{EP}}), \Pi_{c_{\mathbf{U}}}^{\mathrm{res}}(\mathbf{x}_j^{\mathrm{EP}}), \Pi_{d_{\mathbf{U}}}^{\mathrm{res}}(\mathbf{x}_j^{\mathrm{EP}}) \right)$
12:      $\mathbf{V}_j^{\mathrm{EP}} \leftarrow \mathtt{quaternionToRotation}\left(\Pi_{a_V}^{\mathrm{res}}(\mathbf{x}_j^{\mathrm{EP}}), \Pi_{b_V}^{\mathrm{res}}(\mathbf{x}_j^{\mathrm{EP}}), \Pi_{c_V}^{\mathrm{res}}(\mathbf{x}_j^{\mathrm{EP}}), \Pi_{d_V}^{\mathrm{res}}(\mathbf{x}_j^{\mathrm{EP}}) \right)$
13:      $\mathbf{\Sigma}_j^{\mathrm{EP}} \leftarrow \mathrm{diag}\left(\exp\left(\Pi_{\log \sigma^1}^{\mathrm{res}}(\mathbf{x}_j^{\mathrm{EP}}) \right), \exp\left(\Pi_{\log \sigma^2}^{\mathrm{res}}(\mathbf{x}_j^{\mathrm{EP}}) \right), \exp\left(\Pi_{\log \sigma^3}^{\mathrm{res}}(\mathbf{x}_j^{\mathrm{EP}}) \right) \right)$
14:      $\mathbf{F}_j^{\mathrm{EP}} = \mathbf{F}^{\mathrm{EP}}(\mathbf{x}_j^{\mathrm{EP}}) \leftarrow (\mathbf{U}^{\mathrm{EP}})_j^T \mathbf{\Sigma}_j^{\mathrm{EP}} \mathbf{V}_j^{\mathrm{EP}}$
15: **end for**

We remark that, differently from the method presented in [46], the approach proposed in this paper relies on RBF interpolation rather than weighted least-squares approximation. As a consequence, the constructed interpolants exactly match the data on the source mesh at interpolation points.

Remark 4. Both the matrices Φ_{int} and Φ_{eval}, involved in the construction and evaluation of the interpolants of Algorithm 2, only depend on the location of the points $\mathbf{x}_i^{\mathrm{M}}$ and $\mathbf{x}_j^{\mathrm{EP}}$, which are the same for all interpolants and do not change over time. Therefore, the matrices are computed only once during the initialization phase.

Remark 5. The procedure described in Algorithm 2 performs linear interpolation between quaternions, as opposed to spherical interpolation [66], also considered in [46]. This ensures that the whole interpolation procedure is linear. However, it requires the quaternions to be normalized after interpolation, before converting them to rotation matrices.

3.5. Procedure for the alignment of singular vectors

In this section we describe the procedure (reported in Algorithm 3) we propose in order to deal with the indeterminacy of the SVD of the tensors to be interpolated. We notice that the SVD of a generic second-order tensor $\mathbf{F}_i$ can be written as follows

$$\mathbf{F}_i = \mathbf{U}_i \boldsymbol{\Sigma}_i \mathbf{V}_i^T = \sum_{j=1}^{3} \sigma_i^j \mathbf{u}_i^j \otimes \mathbf{v}_i^j \, , \tag{33}$$

where $\mathbf{u}_i^j$ (respectively, $\mathbf{v}_i^j$) is the jth column of $\mathbf{U}_i$ (respectively, $\mathbf{V}_i$), known as left (respectively, right) singular vector. From (33) it is clear that the decomposition of $\mathbf{F}_i$ is unaffected by (i) simultaneous reordering of the singular values and of the columns of $\mathbf{U}_i$ and $\mathbf{V}_i$, and by (ii) change of sign of corresponding columns of $\mathbf{U}_i$ and $\mathbf{V}_i$. Actually, (i) and (ii) are the only sources of indeterminacy [67]. Conventionally, singular values are ordered in a decreasing manner, consequently defining the ordering of the columns of $\mathbf{U}_i$ and $\mathbf{V}_i$. In this work, however, we adopt a different strategy for (i) ordering singular values/vectors and for (ii) defining the orientation of singular vectors.

Algorithm 3 Reordering procedure for singular values and vectors

---

Input: singular values and vectors $\widetilde{\sigma}_i$, $\widetilde{\mathbf{v}}_i$, $\widetilde{\mathbf{u}}_i$, for $i = 1, 2, 3$, in order of decreasing singular values
Input: reference triplet $(\mathbf{w}^1, \mathbf{w}^2, \mathbf{w}^3)$
Output: reordered singular values and vectors σ_i, $\mathbf{v}_i$, $\mathbf{u}_i$, for $i = 1, 2, 3$

  1:   $j_1 \leftarrow \arg\min_i |\mathbf{w}^1 \cdot \widetilde{\mathbf{v}}_i|$
  2:   $\sigma_1 \leftarrow \widetilde{\sigma}_{j_1}$
  3:   **if** $(\mathbf{w}^1 \cdot \mathbf{v}_{j_1}) < 0$ **then**
  4:      $\mathbf{v}_1 \leftarrow -\widetilde{\mathbf{v}}_{j_1}, \quad \mathbf{u}_1 \leftarrow -\widetilde{\mathbf{u}}_{j_1}$
  5:   **else**
  6:      $\mathbf{v}_1 \leftarrow \widetilde{\mathbf{v}}_{j_1}, \quad \mathbf{u}_1 \leftarrow \widetilde{\mathbf{u}}_{j_1}$
  7:   **end if**
  8:
  9:   $j_2 \leftarrow \arg\min_{i \neq j_1} |\mathbf{w}^2 \cdot \widetilde{\mathbf{v}}_i|$
10:   $\sigma_2 \leftarrow \widetilde{\sigma}_{j_2}$
11:   **if** $(\mathbf{w}^2 \cdot \widetilde{\mathbf{v}}_{j_2}) < 0$ **then**
12:      $\mathbf{v}_2 \leftarrow -\widetilde{\mathbf{v}}_{j_2}, \quad \mathbf{u}_2 \leftarrow -\widetilde{\mathbf{u}}_{j_2}$
13:   **else**
14:      $\mathbf{v}_2 \leftarrow \widetilde{\mathbf{v}}_{j_2}, \quad \mathbf{u}_2 \leftarrow \widetilde{\mathbf{u}}_{j_2}$
15:   **end if**
16:
17:   $j_3 \leftarrow \{1, 2, 3\} \backslash \{j_1, j_2\}$
18:   $\sigma_3 \leftarrow \widetilde{\sigma}_{j_3}$
19:   **if** $\det(\mathbf{v}_1, \mathbf{v}_2, \widetilde{\mathbf{v}}_{j_3}) < 0$ **then**
20:      $\mathbf{v}_3 \leftarrow -\widetilde{\mathbf{v}}_{j_3}, \quad \mathbf{u}_3 \leftarrow -\widetilde{\mathbf{u}}_{j_3}$
21:   **else**
22:      $\mathbf{v}_3 \leftarrow \widetilde{\mathbf{v}}_{j_3}, \quad \mathbf{u}_3 \leftarrow \widetilde{\mathbf{u}}_{j_3}$
23:   **end if**

To better explain our procedure, let us consider the application of a tensor $\mathbf{F}_i$ to a generic test vector $\mathbf{a}$:

$$\mathbf{F}_i \mathbf{a} = \mathbf{U}_i \boldsymbol{\Sigma}_i \mathbf{V}_i^T \mathbf{a} = \sum_{j=1}^{3} \sigma_i^j (\mathbf{v}_i^j \cdot \mathbf{a}) \mathbf{u}_i^j \, . \tag{34}$$

The application of $\mathbf{F}_i$ to the vector $\mathbf{a}$ can thus be interpreted as a three-step procedure: (i) we compute the components of $\mathbf{a}$ with respect to the orthonormal basis $(\mathbf{v}_i^1, \mathbf{v}_i^2, \mathbf{v}_i^3)$; (ii) we rescale them by the corresponding singular values; (iii) we compute a linear combination of the left singular vectors with coefficients computed in the previous step. The ordering of the terms at right-hand side of (34) clearly does not impact the result of the sum. However, it affects the results of the interpolation, as the singular values σ_i^j (more precisely, their logarithm) with the same index j are interpolated among points. Since, in (34), the jth singular value plays the role of rescaling the product $(\mathbf{v}_i^j \cdot \mathbf{a})$, in this work we create a correspondence among different points $\mathbf{x}_i^M$ (with $i = 0, 1, \ldots, N_q^M$) according to the directions of the right singular vectors $\mathbf{v}_i^j$. More precisely, we match singular values that correspond to right singular vectors that are most closely aligned with each other.

With this aim, we define a reference orthonormal ordered triplet (which can be the canonical basis of Euclidean space, or the triplet that locally defines the direction of the fibers and sheets), and perform a reordering of the singular values and vectors, as well as a re-orientation of the latter, so as to maximize the alignment of the right singular vectors with the reference triplet. In this way, when we interpolate the singular values associated with different points $\mathbf{x}_i^M$ (with $i = 0, 1, \ldots, N_q^M$), we match singular values corresponding to right singular vectors that are as aligned as possible.

More in detail, the alignment procedure is the following. Let us denote by $(\mathbf{w}_i^1, \mathbf{w}_i^2, \mathbf{w}_i^3)$ the reference triplet, possibly depending on the point $\mathbf{x}_i^M$. We select as first column of the matrix $\mathbf{V}_i$ the right singular vector that maximizes the quantity $|\mathbf{w}_i^1 \cdot \mathbf{v}_i^j|$ over $j = 1, 2, 3$. If $\mathbf{w}_i^1 \cdot \mathbf{v}_i^j < 0$, that is if the cosine of the angle between $\mathbf{w}_i^1$ and $\mathbf{v}_i^j$ is negative, then we invert the sign of the components of the singular vector. Then, we select as second column of the matrix $\mathbf{V}_i$ the maximizer, between the remaining two right singular vectors, of $|\mathbf{w}_i^2 \cdot \mathbf{v}_i^j|$. As before, if $\mathbf{w}_i^2 \cdot \mathbf{v}_i^j < 0$ we invert the sign of its components. The last column of $\mathbf{V}_i$ is assigned as the remaining singular vector, and its sign is selected so that $\det(\mathbf{V}_i) > 0$. Finally, we reorder the singular values and the left singular vectors, and we change the sign of the latter consistently with what done for the right singular vectors.

In this work, we consider the reference triplet to be the canonical basis of the Euclidean space, i.e. $\mathbf{w}_i^k = \mathbf{e}^k$ for $k \in \{1, 2, 3\}$ and all $i \in \{0, 1, \ldots, N_q^M\}$.

4. Numerical approximation

We employ the finite element method [63,64] for the spatial discretization of (6) and (11). We consider either tetrahedral or hexahedral grids, with finite elements of order either 1 or 2 (although the methods proposed here generalize naturally to elements of higher order).

We discretize in time using finite difference schemes of order 1. The monodomain Eq. (6) is discretized in a semi-implicit way, by treating explicitly the ionic current term, thus resulting in a linear problem. The ionic model is discretized with an implicit-explicit (IMEX) scheme, allowing for its direct solution [4]. The mechanics model (11) is discretized with an implicit formulation, and the circulation model (17) is solved with an explicit Euler scheme.

The electrophysiology, force generation, mechanics and circulation models are coupled in a segregated staggered way, that is they are solved independently and sequentially at every time step [4]. The electrophysiology model is solved with a finer temporal discretization than the other models, to satisfy its stricter accuracy requirements [4,37]. Let Δt be the time discretization step, and $\Delta t^{EP} = \Delta t / n^{EP}$ be a smaller discretization step used for electrophysiology. Then, the time advancing scheme is the following:

1. solve n^{EP} time steps of the electrophysiology model (6) and (8), with time step Δt^{EP};
2. interpolate the intracellular calcium concentration from $\mathcal{T}^{EP}$ to $\mathcal{T}^M$ using interpolation between DoFs (Section 3.3.1);
3. solve the force generation model (9) using time discretization step Δt;
4. solve the mechanics model (11) coupled with the circulation model (17), using the time discretization step Δt;
5. interpolate the deformation gradient $\mathbf{F}$ from $\mathcal{T}^M$ to $\mathcal{T}^{EP}$, using interpolation between quadrature nodes (Section 3.4 and Algorithm 2.

We refer the interested reader to [4] for more details on the methods for the electrophysiology, force generation, mechanics and circulation problems (steps 1, 3 and 4 of the procedure above).

Table 1
Type and number of elements, number of nodes, and element diameter (minimum, average and maximum) of the meshes used in the numerical experiments.

Mesh	Element type	# elements	# nodes	Element diameter [mm]		
				Min.	Avg.	Max.
$\mathcal{T}_1$	hexahedra	140 120	159 443	1.1	2.0	3.5
$\mathcal{T}_2$	hexahedra	8 967 680	9 141 953	0.2	0.5	1.4
$\mathcal{T}_3$	tetrahedra	3 838 394	762 797	0.4	0.8	1.2

Remark 6. In principle, we could use interpolation between quadrature nodes to interpolate $[Ca^{2+}]_i$ from the fine mesh $\mathcal{T}^{EP}$ to the coarse one $\mathcal{T}^M$ (step 2 of the procedure above). However, using interpolation between DoFs is more convenient and computationally efficient, since both the ionic model and the force generation model are solved on the DoFs of the respective mesh [4].

4.1. Alternative interpolation methods

The interpolation method previously described will be referred to as RBF-**F**-SVD. For comparison, we will also consider the following alternative interpolation schemes (each of these approaches replaces step 5 of the procedure described in previous section):

- nested-**d**: the meshes $\mathcal{T}^{EP}$ and $\mathcal{T}^M$ are one nested into the other (i.e. the finer one is obtained from the coarser one through one or more refine-by-splitting steps). Then, the displacement field **d** is transferred from the coarse to the fine mesh through standard finite element interpolation, and the deformation gradient **F** is evaluated on quadrature nodes $\mathbf{x}_j^{EP}$. We refer to [2,4] for additional details;
- RBF-**d**: the displacement field **d** is interpolated from the coarse to the fine mesh, using RBF interpolation between DoFs (Section 3.3.1); then, the deformation gradient **F** is evaluated on quadrature nodes $\mathbf{x}_j^{EP}$ through standard finite element interpolation. We refer to [32] for further details;
- RBF-**F**-E: we perform interpolation between quadrature nodes (Section 3.3.2) of each cartesian component of the deformation gradient **F**; this is also known as Euclidean interpolation [46].

5. Numerical experiments

All simulations discussed below are performed using `life`$^{\text{x}}$, a C++ high-performance library for cardiac applications [68–70] based on the finite element core `deal.II` [39,71]. The RBF-**F**-SVD interpolation method is implemented in the `life`$^{\text{x}}$ core module,[2] available as open source under the GNU LGPLv3 license. We refer the interested reader to the `rbf_interpolation` example and the associated code documentation.

The computational domain is the left ventricle of the Zygote Heart Model [72], pre-processed using the techniques described in [73]. We report in Table 1 details on the meshes (see Fig. 4) used throughout the numerical experiments discussed below.

All simulations presented in the following sections are run on the GALILEO100 supercomputer[3] from the CINECA supercomputing center (Italy). Unless otherwise specified, simulations are run in parallel using 192 cores.

5.1. A comparison of interpolation methods

As a first test, we use $\mathcal{T}_1$ as a coarse mesh, for mechanics and force generation, and $\mathcal{T}_2$ as a fine mesh, for electrophysiology (see Table 1). The two meshes are one nested within the other, that is the mesh $\mathcal{T}_2$ is obtained by applying a refine-by-splitting procedure to $\mathcal{T}_1$ two times (thus subdividing every element into 64 smaller elements, see Fig. 4). This choice allows to compare all four interpolation methods described above (nested-**d**, RBF-**d**, RBF-**F**-E and RBF-**F**-SVD). We consider bilinear finite elements for all the problems involved.

---

[2] https://gitlab.com/lifex/lifex
[3] Technical specifications: https://wiki.u-gov.it/confluence/display/SCAIUS/UG3.3%3A+GALILEO100+UserGuide.
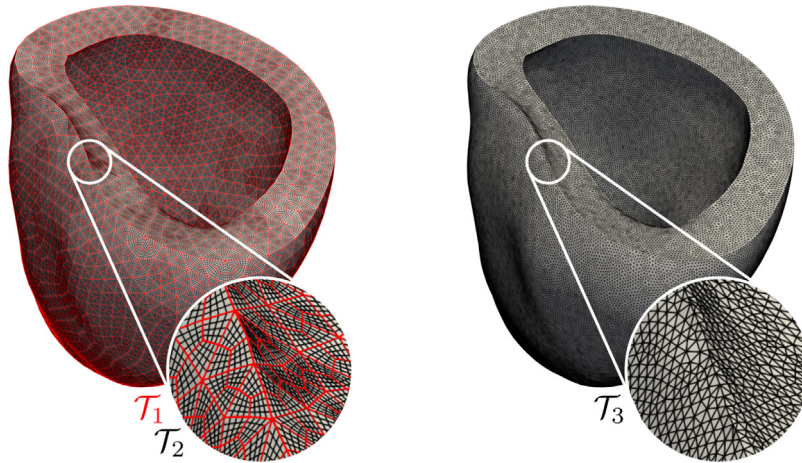
Fig. 4. Left: the meshes $\mathcal{T}_1$ (red) and $\mathcal{T}_2$ (black). Right: the mesh $\mathcal{T}_3$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 2
(a) Physical parameters for the simulations of Section 5.1. Conductivities are calibrated to obtain conduction velocities along fibers, sheets and cross-fibers of 0.6 m/s, 0.4 m/s and 0.2 m/s on the mesh used for electrophysiology [74]. We only report parameters whose value is different from that used in [4]. (b) Parameters controlling the adaptive RBF radius in the tests of Sections 5.1 and 5.3.

Parameter	Value	Unit
σ_f	1.68e−4	m^2/s
σ_s	0.769e−4	m^2/s
σ_n	0.248e−4	m^2/s
T_{act}^{max}	600	kPa

(a)

Interpolated quantity	M	α
$[Ca^{2+}]_i$	1	2.5
d	5	3.0
F (Euclidean)	2	2.0
F (SVD)	2	2.0

(b)

We perform the simulation of a heartbeat in healthy conditions, setting $T = 800$ ms, $\Delta t = 1$ ms and $n^{EP} = 10$ (so that $\Delta t^{EP} = 0.1$ ms).

Table 2 lists the value of the parameters used in setting up the model. For the sake of brevity, we only report the parameters whose value is different from that used in [4].

The parameters M and α for the RBF interpolation of $[Ca^{2+}]_i$, **d** and **F** were obtained by trial-and-error, to guarantee that no spurious oscillations appeared, while keeping the RBF radius as small as possible to reduce the computational cost. The development of more systematic methods for the choice of M and α will be the subject of further studies. For RBF interpolation between quadrature points, we test both $q = 1$ (i.e. the interpolation points are the barycenters of each element) and $q = 2$ (corresponding to 8 interpolation points per element). The parameters used to select the adaptive RBF radius for the different interpolation methods in this comparison are reported in Table 2.

In this setting, the interpolation methods nested-**d** (used in [2,4]) and RBF-**d** (used in [32]) fail to complete the simulation. Indeed, both yield negative values of J on some of the quadrature nodes of the fine mesh, and this causes the electrophysiology solver to diverge at times $t = 158$ ms and $t = 213$ ms, respectively. This can be
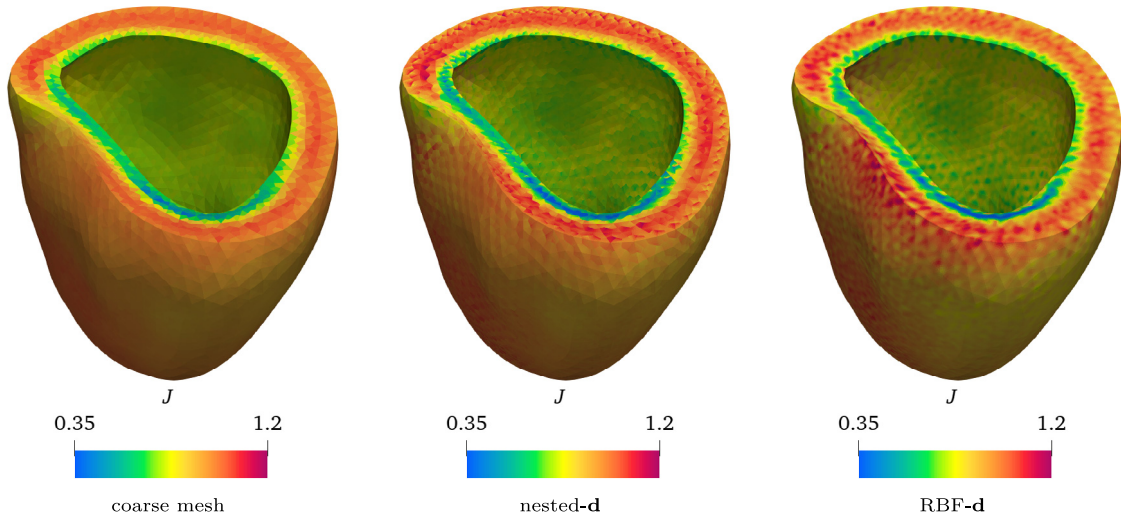
Fig. 5. For the test of Section 5.1, determinant of the deformation gradient J on the coarse mesh $\mathcal{T}_1$ (left) and on the fine mesh $\mathcal{T}_2$ using the interpolation methods nested-**d** (center) and RBF-**d** (right). The plots are done at time $t = 155$ ms.

explained by looking at the spatial distribution of J after interpolation, as reported in Fig. 5: although the overall distribution of J is captured on the fine mesh, spurious oscillations are introduced. It is our experience that whether or not the solver crashes is extremely sensitive to the physical and numerical parameters (including the space and time discretizations). Nonetheless, the solver failure in this setting is exemplar of the unreliability of interpolation methods that transfer the displacement field and then compute its gradient.

On the contrary, both the RBF-**F**-E and RBF-**F**-SVD interpolation methods allow the simulation to reach the final time. We attribute this difference in behavior to the fact that the interpolation methods nested-**d** and RBF-**d** lead to values of J on the fine mesh that are significantly different from those on the coarse mesh, including negative values, whereas RBF-**F**-E and RBF-**F**-SVD allow a more accurate interpolation and yield $J > 0$, ensuring the well-posedness of the discrete monodomain equation.

To quantify this effect, we report in Figs. 6 and 7 the histograms of the values of J on the quadrature nodes of $\mathcal{T}^{\text{EP}}$, computed with the different interpolation methods, for a representative time instant during systolic contraction ($t = 155$ ms). Although all four methods capture the overall distribution of J (as seen in Fig. 6), significant differences are present in the tails of the distributions (as highlighted by the logarithmic scale of Fig. 7). From the plots, we can notice how the evaluations of J with the methods based on the displacement (nested-**d** and RBF-**d**) yield values that fall outside of the range observed on the coarse mesh. These values are physically unrealistic (given the near-incompressibility of the solid constitutive law), and occasionally even negative (to which we attribute the failure of the solver). On the contrary, interpolation methods based on the deformation gradient (RBF-**F**-E and RBF-**F**-SVD) yield a range for J that is closer to the one on the coarse mesh, in particular avoiding negative values and thus preventing the failure of the solver.

We also observe that setting $q = 1$ in the methods RBF-**F**-E and RBF-**F**-SVD has a regularizing effect, filtering out the most extreme values of J. On the other hand, when setting $q = 2$, thus increasing the number of interpolation points, the distribution of J is more accurately recovered. This is especially true with the RBF-**F**-SVD scheme, whereas the RBF-**F**-E scheme yields a small number of points for which J is close to zero. Seeing as this approach does not provide a theoretical guarantee that $J > 0$, it is possible that under different simulation settings (e.g. with a higher contractility, or in pathological conditions) this may lead to the failure of the solver. Conversely, the RBF-**F**-SVD method guarantees by construction that $J > 0$ after interpolation (regardless of the choice of q), and as such it provides an accurate and robust tool to implement mechano-electrical feedbacks.

Finally, we report in Fig. 8 the pressure–volume loops associated with all simulations. We observe that, regardless of the procedure employed, ventricular pressure and volume are essentially identical. This is in agreement with the histograms of Fig. 6, which show that all the methods yield comparable results for the overall distribution of J.
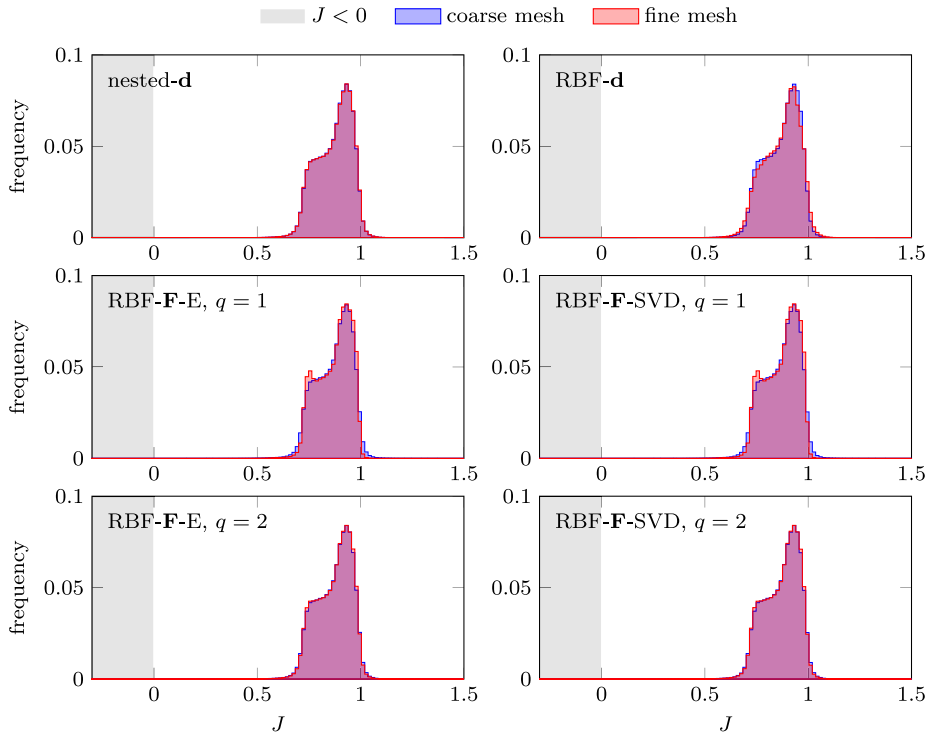
Fig. 6. Histograms of the evaluation of J at mesh quadrature nodes, at $t = 155$ ms, on the coarse mesh $\mathcal{T}^{M}$ (blue) and on the fine mesh $\mathcal{T}^{EP}$ (red), with the different interpolation methods. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Thus, the interpolation methods based on $\mathbf{F}$ do not introduce perturbations with respect to the ones based on $\mathbf{d}$ and previously used in validated electromechanical simulations [2,4,32], while improving over those in terms of robustness.

5.2. Parallel implementation and scalability

Cardiac electromechanical models are usually implemented in a parallel computing framework, distributing the computational load across multiple processors to keep the overall time-to-solution small in spite of the large amount of unknowns. It is therefore important that the chosen intergrid interpolation method is efficient and scalable in a parallel computing setting.

To quantify the parallel performance of our implementation, we performed a strong scalability test for the four interpolation methods discussed before, using the same setting as in Section 5.1. The results are reported in Fig. 9, where we separately report the wall time spent in the initialization and the evaluation of the interpolant. We observe that, while RBF interpolation has a significantly higher initialization cost than the nested approach (based on intergrid transfer operators implemented in deal.II [39]), the cost for initialization reduces linearly with the number of processors employed. Moreover, the initialization step consists in assembling the matrices Φ_{int}, Φ_{eval} and P^{-1}. Since all of these only depend on the location of the points $\mathbf{x}_i^{\text{src}}$ and $\mathbf{x}_i^{\text{dst}}$, this computation can be performed only once in an offline phase, and subsequently reused in multiple simulations on the same mesh, thus amortizing the initialization cost.

For all interpolation methods we observe linear scalability of the computational cost associated with the evaluation of the interpolant (Fig. 9, right). We also observe that RBF interpolants exhibit a slightly better performance than nested interpolation.

Finally, Fig. 10 reports a comparison of the computational cost associated with the evaluation of the interpolant and with the solution of the model equations themselves. The former is negligible with respect to the solution of
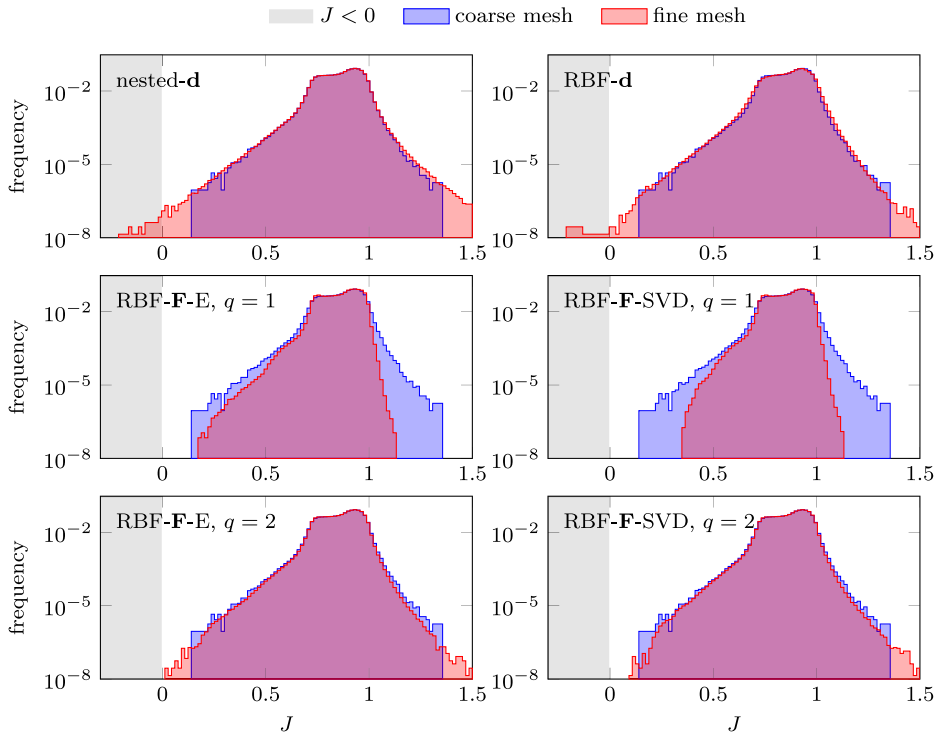
Fig. 7. Histograms of the evaluation of J at mesh quadrature nodes, at $t = 155\,\text{ms}$, on the coarse mesh $\mathcal{T}^{M}$ (blue) and on the fine mesh $\mathcal{T}^{EP}$ (red), with the different interpolation methods. The vertical axes are in logarithmic scale. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)
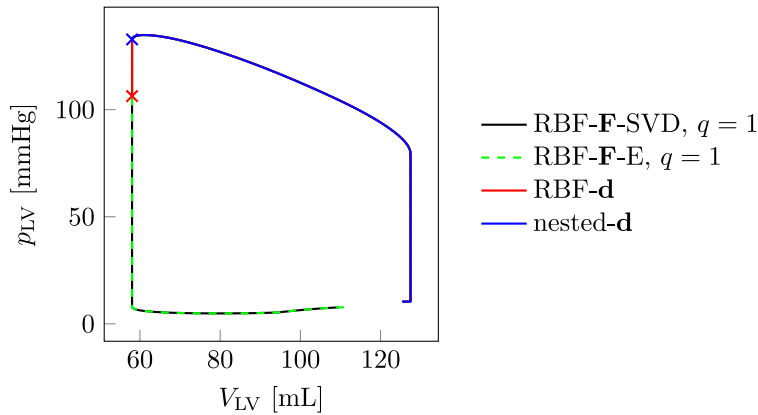

Fig. 8. Pressure–volume loops of obtained with different interpolation methods for the test case of Section 5.1. Cross markers indicate the failure of the solver. For readability, we only report the cases with $q = 1$, since those with $q = 2$ yield identical results.

the electrophysiology and mechanics equations (which dominate the overall computational time). We conclude that the proposed intergrid transfer method does not significantly affect the overall computational cost of the simulation.

5.3. Flexibility of the interpolation method

To further highlight the flexibility of the proposed interpolation method, we consider a new test using mesh $\mathcal{T}_1$ for mechanics (as done in previous sections), and $\mathcal{T}_3$ for electrophysiology (see Tables 1 and 4). The fine mesh is
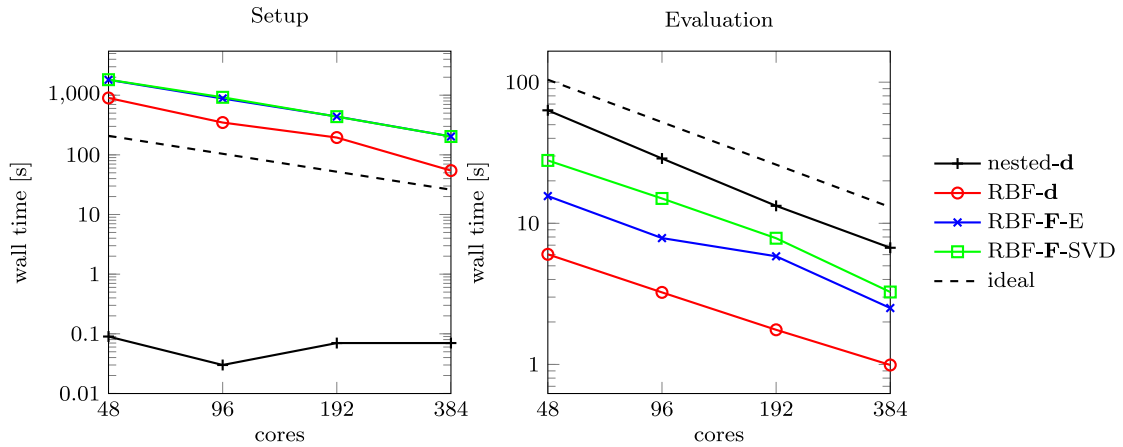
Fig. 9. Strong scalability test for the coarse-to-fine interpolation. Left: wall time for the initialization of the data structures for interpolation. Right: wall time spent in evaluating the interpolant during the simulation (with a final time $T = 10\,\text{ms}$).
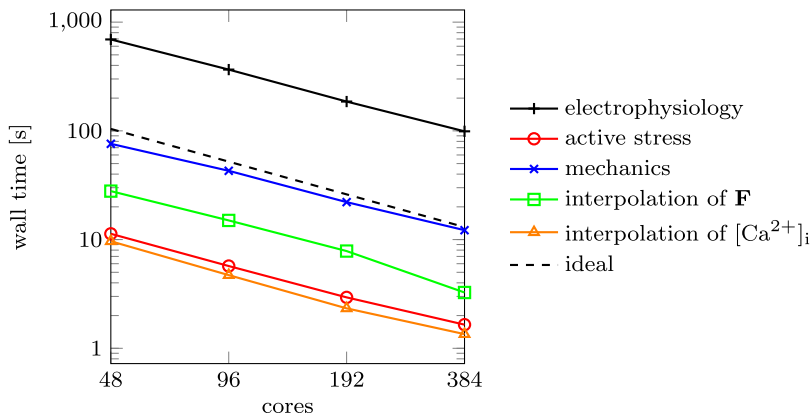

Fig. 10. Comparison of the computational cost associated with intergrid interpolation and with the solution of model equations (with final time $T = 10\,\text{ms}$). $\mathbf{F}$ is interpolated with the RBF-$\mathbf{F}$-SVD method.

not obtained through a refine-by-splitting procedure from the coarse mesh, as done in previous examples. Instead, it is a fully independent mesh, composed of tetrahedral elements (whereas the coarse mesh $\mathcal{T}_1$ is composed of hexahedra). The electrophysiology equations are discretized using quadratic finite elements (for a total of $5\,224\,245$ degrees of freedom). We use the RBF-$\mathbf{F}$-SVD scheme for the interpolation of $\mathbf{F}$, setting $q = 1$ and using the same parameters as in Section 5.1 to select the RBF support radius (see Table 2). We point out that the element type and the polynomial degree used for mechanics and electrophysiology are different. We use the same physical parameters as in [4], except for setting $T_{\text{act}}^{\max} = 600\,\text{kPa}$.

We report the pressure–volume loop for this test case, as well as some snapshots of the solution, in Fig. 11. The results are consistent with those of Section 5.1 (as well as with those of [4]), even if using two entirely independent (not nested) meshes for electrophysiology and mechanics. Minor quantitative differences with Section 5.1 are observed due to the different resolutions of $\mathcal{T}_2$ and $\mathcal{T}_3$.

This test shows how the proposed interpolation approach allows the transfer of solutions between radically different finite element spaces. Thus, it provides a useful tool to efficiently tackle the multiphysics nature of computational models of the heart.

We further remark that, since only the evaluation of $\mathbf{F}$ at individual points in the coarse mesh is required, the proposed method can be used in combination with PDE discretization methods other than finite elements, such as finite differences, discontinuous Galerkin methods, spectral elements or isogeometric analysis.
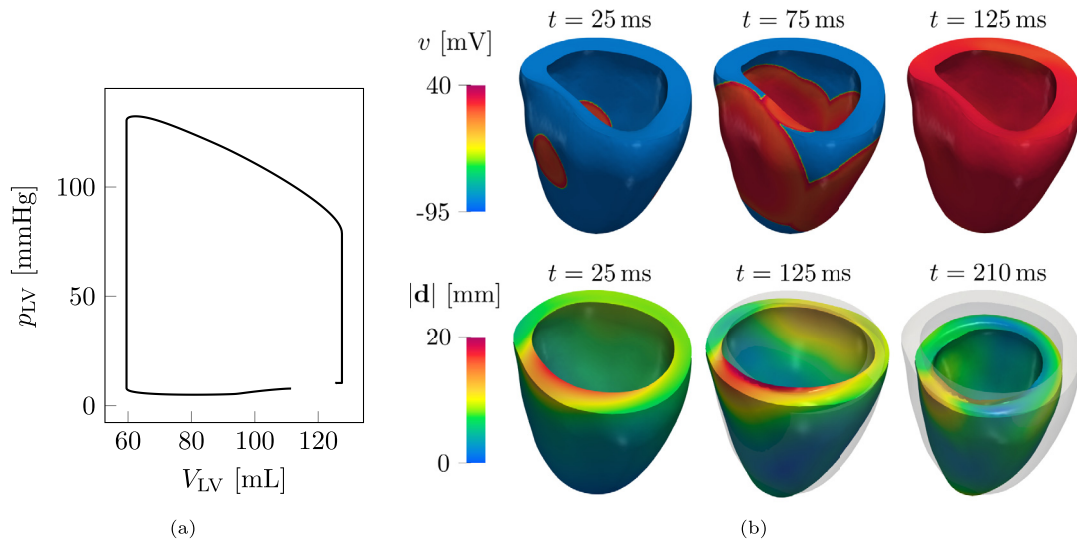
$t = 25\,\text{ms}$ $\quad$ $t = 75\,\text{ms}$ $\quad$ $t = 125\,\text{ms}$

$v\ [\text{mV}]$ 40 -95

$t = 25\,\text{ms}$ $\quad$ $t = 125\,\text{ms}$ $\quad$ $t = 210\,\text{ms}$

$|\mathbf{d}|\ [\text{mm}]$ 20 0

(a)

(b)

Fig. 11. (a) Pressure–volume loop for the simulation of Section 5.3 (black solid line). For comparison, we also report the pressure–volume loop obtained in Section 5.1 using RBF-**F**-SVD (red dashed line; also shown in Fig. 8). (b) Snapshots of the transmembrane potential v (top) and of the displacement magnitude $|\mathbf{d}|$ (bottom) for the simulation of Section 5.3. The bottom plots are warped by $\mathbf{d}$, and the initial configuration is superimposed in transparency. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

6. Conclusions

We introduced a method to transfer the deformation gradient from a coarse to a fine mesh in electromechanical simulations. This is a key ingredient when using electromechanical models that account for mechano-electrical feedback effects (both geometrical and physiological, such as stretch-activated currents). These models are essential to ensure accurate simulations, especially in pathological scenarios [34]. The proposed method is based on combining rescaled, localized RBF interpolation with the SVD factorization of the deformation gradient tensor **F**.

The proposed method is compared to alternative existing approaches in the literature, based on nested intergrid interpolation or RBF interpolation of the displacement field. Both these techniques have been previously applied to physiological simulation of cardiac electromechanics [2,4,32]. The numerical experiments carried out in this work highlight the shortcomings of the existing methods, especially in terms of their robustness, by considering a numerical setting under which they lead to the failure of the electromechanical solver. We attribute this failure to the fact that the existing methods do not preserve the positivity of $J = \det \mathbf{F}$, and indeed we observe points for which $J < 0$ in our numerical experiments, just before the solver failure.

Conversely, the proposed method guarantees that, after interpolation, $J > 0$, and therefore it ensures the robustness of the solver. Moreover, our results indicate that methods based on interpolating the deformation gradient, rather than the displacement field as previously done, do not introduce artificially (and unphysically) large or small values for J. We believe that this feature is especially relevant if physiological mechano-electric feedbacks (such as stretch-activated currents) or pathological scenarios are considered. An analysis of the performance of the method under pathological conditions will be the subject of future studies.

Finally, we highlighted through a numerical experiment how the proposed method allows to easily transfer solution variables between finite elements of different degree and even of different element shape (tetrahedral or hexahedral, in our case). This result indicates that the proposed interpolation method can be a valuable tool in improving not only the robustness, but also the geometrical and parametric flexibility of electromechanical simulations, allowing to independently tailor the discretization of each core model to its specific accuracy needs.

Declaration of competing interest

Education University and Research. Alfio Quarteroni reports financial support was provided by Government of Italy Ministry of Education University and Research.

Data availability

Acknowledgments

References

[1] S.A. Niederer, J. Lumens, N.A. Trayanova, Computational models in cardiology, Nat. Rev. Cardiol. 16 (2) (2019) 100–111.

[2] R. Piersanti, F. Regazzoni, M. Salvador, A.F. Corno, C. Vergara, A. Quarteroni, et al., 3D–0D closed-loop model for the simulation of cardiac biventricular electromechanics, Comput. Methods Appl. Mech. Engrg. 391 (2022) 114607.

[3] A. Quarteroni, T. Lassila, S. Rossi, R. Ruiz-Baier, Integrated heart – Coupling multiscale and multiphysics models for the simulation of the cardiac function, Comput. Methods Appl. Mech. Engrg. 314 (2017) 345–407.

[4] F. Regazzoni, M. Salvador, P.C. Africa, M. Fedele, L. Dede', A. Quarteroni, A cardiac electromechanical model coupled with a lumped-parameter model for closed-loop blood circulation, J. Comput. Phys. 457 (2022) 111083.

[5] M. Salvador, M. Fedele, P.C. Africa, E. Sung, L. Dede', A. Prakosa, J. Chrispin, N.A. Trayanova, A. Quarteroni, Electromechanical modeling of human ventricles with ischemic cardiomyopathy: numerical simulations in sinus rhythm and under arrhythmia, Comput. Biol. Med. 136 (2021) 104674.

[6] M. Strocchi, M.A. Gsell, C.M. Augustin, O. Razeghi, C.H. Roney, A.J. Prassl, E.J. Vigmond, J.M. Behar, J.S. Gould, C.A. Rinaldi, M.J. Bishop, G. Plank, S.A. Niederer, Simulating ventricular systolic motion in a four-chamber heart model with spatially varying Robin boundary conditions to model the effect of the pericardium, J. Biomech. 101 (2020) 109645.

[7] S. Stella, F. Regazzoni, C. Vergara, L. Dede', A. Quarteroni, A fast cardiac electromechanics model coupling the eikonal and the nonlinear mechanics equations, Math. Models Methods Appl. Sci. 32 (08) (2022) 1531–1556.

[8] A. Quarteroni, L. Dede', F. Regazzoni, Modeling the cardiac electromechanical function: A mathematical journey, Bull. Amer. Math. Soc. 59 (3) (2022) 371–403.

[9] F. Levrero-Florencio, F. Margara, E. Zacur, A. Bueno-Orovio, Z. Wang, A. Santiago, J. Aguado-Sierra, G. Houzeaux, V. Grau, D. Kay, M. Vázquez, R. Ruiz-Baier, B. Rodriguez, Sensitivity analysis of a strongly-coupled human-based electromechanical cardiac model: Effect of mechanical parameters on physiologically relevant biomarkers, Comput. Methods Appl. Mech. Engrg. 361 (2020) 112762.

[10] A. Quarteroni, L. Dede', A. Manzoni, C. Vergara, Mathematical modelling of the human cardiovascular system: data, numerical approximation, clinical applications, vol. 33, Cambridge University Press, 2019.

[11] 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) 1247.

[12] C.M. Augustin, A. Crozier, A. Neic, A.J. Prassl, E. Karabelas, T. Ferreira da Silva, J.F. Fernandes, F. Campos, T. Kuehne, G. Plank, Patient-specific modeling of left ventricular electromechanics as a driver for haemodynamic analysis, EP Europace 18 (2016) iv121–iv129.

[13] V. Gurev, T. Lee, J. Constantino, H. Arevalo, N.A. Trayanova, Models of cardiac electromechanics based on individual hearts imaging data: Image-based electromechanical models of the heart, Biomech. Model. Mechanobiol. 10 (3) (2011) 295–306.

[14] C.M. Augustin, M.A. Gsell, E. Karabelas, E. Willemen, F.W. Prinzen, J. Lumens, E.J. Vigmond, G. Plank, A computationally efficient physiologically comprehensive 3D–0D closed-loop model of the heart and circulation, Comput. Methods Appl. Mech. Engrg. 386 (2021) 114092.

[15] M. Bucelli, A. Zingaro, P.C. Africa, I. Fumagalli, L. Dede', A. Quarteroni, A mathematical model that integrates cardiac electrophysiology, mechanics, and fluid dynamics: Application to the human left heart, Int. J. Numer. Methods Biomed. Eng. 39 (3) (2023) e3678.

[16] A. Santiago, J. Aguado-Sierra, M. Zavala-Aké, R. Doste-Beltran, S. Gómez, R. Arís, J.C. Cajas, E. Casoni, M. Vázquez, Fully coupled fluid-electro-mechanical model of the human heart for supercomputers, Int. J. Numer. Methods Biomed. Eng. 34 (12) (2018) e3140.

[17] F. Viola, V. Spandan, V. Meschini, J. Romero, M. Fatica, M.D. de Tullio, R. Verzicco, FSEI-GPU: GPU accelerated simulations of the fluid–structure–electrophysiology interaction in the left heart, Comput. Phys. Comm. 273 (2022) 108248.

[18] F. Viola, G. Del Corso, R. De Paulis, R. Verzicco, GPU accelerated digital twins of the human heart open new routes for cardiovascular research, Sci. Rep. 13 (1) (2023) 8230.

[19] R. Verzicco, Electro-fluid-mechanics of the heart, J. Fluid Mech. 941 (2022).

[20] A. Zingaro, M. Bucelli, R. Piersanti, F. Regazzoni, L. Dede', A. Quarteroni, An electromechanics-driven fluid dynamics model for the simulation of the whole human heart, 2023, arXiv preprint arXiv:2301.02148.

[21] E. Karabelas, M.A. Gsell, C.M. Augustin, L. Marx, A. Neic, A.J. Prassl, L. Goubergrits, T. Kuehne, G. Plank, Towards a computational framework for modeling the impact of aortic coarctations upon left ventricular load, Front. Physiol. 9 (2018) 538.

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

[23] 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. Engrg. 373 (2021) 113468.

[24] J. Sundnes, G.T. Lines, X. Cai, B.F. Nielsen, K.-A. Mardal, A. Tveito, Computing the Electrical Activity in the Heart, Vol. 1, Springer Science & Business Media, 2007.

[25] E. Sung, A. Prakosa, K.N. Aronis, S. Zhou, S.L. Zimmerman, H. Tandri, S. Nazarian, R.D. Berger, J. Chrispin, N.A. Trayanova, Personalized digital-heart technology for ventricular tachycardia ablation targeting in hearts with infiltrating adiposity, Circulation: Arrhythm. Electrophysiol. 13 (12) (2020) e008912.

[26] K. Gillette, M.A. Gsell, A.J. Prassl, E. Karabelas, U. Reiter, G. Reiter, T. Grandits, C. Payer, D. Štern, M. Urschler, J.D. Bayer, C.M. Augustin, A. Neic, T. Pock, E.J. Vigmond, G. Plank, A framework for the generation of digital twins of cardiac electrophysiology from clinical 12-leads ECGs, Med. Image Anal. 71 (2021) 102080.

[27] M. Bucelli, M. Salvador, L. Dede', A. Quarteroni, Multipatch isogeometric analysis for electrophysiology: simulation in a human heart, Comput. Methods Appl. Mech. Engrg. 376 (2021) 113666.

[28] L. Pegolotti, L. Dede', A. Quarteroni, Isogeometric analysis of the electrophysiology in the human heart: numerical simulation of the bidomain equations on the atria, Comput. Methods Appl. Mech. Engrg. 343 (2019) 52–73.

[29] H.J. Arevalo, F. Vadakkumpadan, E. Guallar, A. Jebb, P. Malamas, K.C. Wu, N.A. Trayanova, Arrhythmia risk stratification of patients after myocardial infarction using personalized heart models, Nature Commun. 7 (1) (2016) 1–8.

[30] S.A. Niederer, E. Kerfoot, A.P. Benson, M.O. Bernabeu, O. Bernus, C. Bradley, E.M. Cherry, R. Clayton, F.H. Fenton, A. Garny, et al., Verification of cardiac tissue electrophysiology simulators using an N-version benchmark, Phil. Trans. R. Soc. A 369 (1954) (2011) 4331–4351.

[31] N.A. Trayanova, F. Pashakhanloo, K.C. Wu, H.R. Halperin, Imaging-based simulations for predicting sudden death and guiding ventricular tachycardia ablation, Circulation: Arrhythm. Electrophysiol. 10 (7) (2017) e004743.

[32] M. Salvador, L. Dede', A. Quarteroni, An intergrid transfer operator using radial basis functions with application to cardiac electromechanics, Comput. Mech. 66 (2020) 491–511.

[33] A. Collet, Numerical Modeling of the Cardiac Mechano-Electric Feedback Within a Thermo-Electro-Mechanical Framework. Study of Its Consequences on Arrhythmogenesis (Ph.D. thesis), Université de Liège, Liège, Belgique, 2015.

[34] M. Salvador, F. Regazzoni, S. Pagani, N.A. Trayanova, A. Quarteroni, et al., The role of mechano-electric feedbacks and hemodynamic coupling in scar-related ventricular tachycardia, Comput. Biol. Med. 142 (2022) 105203.

[35] V. Timmermann, L.A. Dejgaard, K.H. Haugaa, A.G. Edwards, J. Sundnes, A.D. McCulloch, S.T. Wall, An integrative appraisal of mechano-electric feedback mechanisms in the heart, Prog. Biophys. Mol. Biol. 130 (2017) 404–417.

[36] T.A. Quinn, P. Kohl, Cardiac mechano-electric coupling: acute effects of mechanical stimulation on heart rate and rhythm, Physiol. Rev. 101 (1) (2021) 37–92.

[37] M. Fedele, R. Piersanti, F. Regazzoni, M. Salvador, P.C. Africa, M. Bucelli, A. Zingaro, A. Quarteroni, et al., A comprehensive and biophysically detailed computational model of the whole human heart electromechanics, Comput. Methods Appl. Mech. Engrg. 410 (2023) 115983.

[38] P.C. Africa, M. Salvador, P. Gervasio, L. Dede', A. Quarteroni, A matrix–free high–order solver for the numerical solution of cardiac electrophysiology, J. Comput. Phys. 478 (2023) 111984.

[39] D. Arndt, W. Bangerth, D. Davydov, T. Heister, L. Heltai, M. Kronbichler, M. Maier, J. Pelteret, B. Turcksin, D. Wells, The deal.II finite element library: design, features, and insights, Comput. Math. Appl. 81 (2021) 407–422.

[40] C. Burstedde, L.C. Wilcox, O. Ghattas, p4est: Scalable algorithms for parallel adaptive mesh refinement on forests of octrees, SIAM J. Sci. Comput. 33 (3) (2011) 1103–1133.

[41] P. Pathmanathan, G.R. Mirams, J. Southern, J.P. Whiteley, The significant effect of the choice of ionic current integration method in cardiac electro-physiological simulations, Int. J. Numer. Methods Biomed. Eng. 27 (11) (2011) 1751–1770.

[42] S. Deparis, D. Forti, P. Gervasio, A. Quarteroni, INTERNODES: An accurate interpolation-based method for coupling the Galerkin solutions of PDEs on subdomains featuring non-conforming interfaces, Comput. & Fluids 141 (2016) 22–41.

[43] S. Deparis, D. Forti, A. Quarteroni, A rescaled localized radial basis function interpolation on non-cartesian and nonconforming grids, SIAM J. Sci. Comput. 36 (6) (2014) A2745–A2762.

[44] Y. Voet, G. Anciaux, S. Deparis, P. Gervasio, The INTERNODES method for applications in contact mechanics and dedicated preconditioning techniques, Comput. Math. Appl. 127 (2022) 48–64.

[45] A. Gerbi, L. Dede', A. Quarteroni, A monolithic algorithm for the simulation of cardiac electromechanics in the human left ventricle, Math. Eng. 1 (1) (2018) 1–37.

[46] A. Satheesh, C.P. Schmidt, W.A. Wall, C. Meier, Structure-preserving invariant interpolation schemes for invertible second-order tensors, 2022, arXiv preprint arXiv:2211.16507.

[47] S. Rossi, T. Lassila, R. Ruiz-Baier, A. Sequeira, A. Quarteroni, Thermodynamically consistent orthotropic activation model capturing ventricular systolic wall thickening in cardiac electromechanics, Eur. J. Mech. A Solids 48 (2014) 129–142.

[48] 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.

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

[50] F. Regazzoni, L. Dede', A. Quarteroni, Active contraction of cardiac cells: A reduced model for sarcomere dynamics with cooperative interactions, Biomech. Model. Mechanobiol. 17 (6) (2018) 1663–1686.

[51] F. Regazzoni, L. Dede', A. Quarteroni, Machine learning for fast and reliable solution of time-dependent differential equations, J. Comput. Phys. 397 (2019) 108852.

[52] F. Regazzoni, L. Dede', A. Quarteroni, Machine learning of multiscale active force generation models for the efficient simulation of cardiac electromechanics, Comput. Methods Appl. Mech. Engrg. 370 (2020) 113268.

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

[54] G.A. Holzapfel, R.W. Ogden, Constitutive modelling of passive myocardium: A structurally based framework for material characterization, Phil. Trans. R. Soc. A 367 (1902) (2009) 3445–3475.

[55] M.R. Pfaller, J.M. Hörmann, M. Weigl, A. Nagler, R. Chabiniok, C. Bertoglio, W.A. Wall, The importance of the pericardium for cardiac biomechanics: from physiology to computational modeling, Biomech. Model. Mechanobiol. 18 (2) (2019) 503–529.

[56] P.J. Blanco, R.A. Feijóo, A 3D-1D-0D computational model for the entire cardiovascular system, Mec. Comput. 29 (59) (2010) 5887–5911.

[57] M. Hirschvogel, M. Bassilious, L. Jagschies, S.M. Wildhirt, M.W. Gee, A monolithic 3D-0D coupled closed-loop model of the heart and the vascular system: experiment-based parameter estimation for patient-specific cardiac mechanics, Int. J. Numer. Methods Biomed. Eng. 33 (8) (2017) e2842.

[58] H. Wendland, Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree, Adv. Comput. Math. 4 (1) (1995) 389–396.

[59] Y. Saad, Iterative Methods for Sparse Linear Systems, SIAM, 2003.

[60] R.K. Beatson, J.B. Cherrie, C.T. Mouat, Fast fitting of radial basis functions: Methods based on preconditioned GMRES iteration, Adv. Comput. Math. 11 (1999) 253–270.

[61] D. Brown, L. Ling, E. Kansa, J. Levesley, On approximate cardinal preconditioning methods for solving PDEs with radial basis functions, Eng. Anal. Bound. Elem. 29 (4) (2005) 343–353.

[62] N.A. Gumerov, R. Duraiswami, Fast radial basis function interpolation via preconditioned krylov iteration, SIAM J. Sci. Comput. 29 (5) (2007) 1876–1899.

[63] T.J.R. Hughes, the Finite Element Method: Linear Static and Dynamic Finite Element Analysis, Courier Corporation, 2012.

[64] A. Quarteroni, Numerical Models for Differential Problems, Vol. 16, Springer, 2017.

[65] O.C. Zienkiewicz, J.Z. Zhu, The superconvergent patch recovery and a posteriori error estimates. Part 1: The recovery technique, Internat. J. Numer. Methods Engrg. 33 (7) (1992) 1331–1364.

[66] K. Shoemake, Animating rotation with quaternion curves, in: Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques, 1985, pp. 245–254.

[67] S. Roman, S. Axler, F. Gehring, Advanced Linear Algebra, Vol. 3, Springer, 2005.

[68] P.C. Africa, lifex: A flexible, high performance library for the numerical solution of complex finite element problems, SoftwareX 20 (2022) 101252.

[69] P.C. Africa, I. Fumagalli, M. Bucelli, A. Zingaro, M. Fedele, L. Dede', A. Quarteroni, Lifex-cfd: An open-source computational fluid dynamics solver for cardiovascular applications, 2023, arXiv preprint arXiv:2304.12032.

[70] P.C. Africa, R. Piersanti, M. Fedele, L. Dede', A. Quarteroni, Lifex-fiber: An open tool for myofibers generation in cardiac computational models, BMC Bioinformatics 24 (2023) 143.

[71] D. Arndt, W. Bangerth, M. Feder, M. Fehling, R. Gassmöller, T. Heister, L. Heltai, M. Kronbichler, M. Maier, P. Munch, et al., The `deal.II` library, version 9.4, J. Numer. Math. 30 (3) (2022) 231–246.

[72] Zygote Media Group Inc., Zygote Solid 3D Heart Generation II Developement Report, Technical report, 2014.

[73] M. Fedele, A. Quarteroni, Polygonal surface processing and mesh generation tools for the numerical simulation of the cardiac function, Int. J. Numer. Methods Biomed. Eng. 37 (4) (2021) e3435.

[74] C.M. Augustin, A. Neic, M. Liebmann, A.J. Prassl, S.A. Niederer, G. Haase, G. Plank, Anatomically accurate high resolution modeling of human whole heart electromechanics: A strongly scalable algebraic multigrid solver method for nonlinear deformation, J. Comput. Phys. 305 (2016) 622–646.