MODELLISTICA E CALCOLO SCIENTIFICO

MOX

MODELING AND SCIENTIFIC COMPUTING

MOX-Report No. 26/2024

Robust radial basis function interpolation based on geodesic distance for the numerical coupling of multiphysics problems

Bucelli, M.; Regazzoni, F.; Dede', L.; Quarteroni, A.

Robust radial basis function interpolation based on geodesic distance for the numerical coupling of multiphysics problems

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 (Professor Emeritus)*

Abstract

Multiphysics simulations frequently require transferring solution fields between subproblems with non-matching spatial discretizations, typically using interpolation techniques. Standard methods are usually based on measuring the closeness between points by means of the Euclidean distance, which does not account for curvature, cuts, cavities or other non-trivial geometrical or topological features of the domain. This may lead to spurious oscillations in the interpolant in proximity to these features. To overcome this issue, we propose a modification to rescaled localized radial basis function (RL-RBF) interpolation to account for the geometry of the interpolation domain, by yielding conformity and fidelity to geometrical and topological fieatures. The proposed method, referred to as RL-RBF-G, relies on measuring the geodesic distance between data points. RL-RBF-G removes all oscillations appearing in the RL-RBF interpolant, resulting in increased accuracy in domains with complex geometries. We demonstrate the effectiveness of RL-RBF-G interpolation through a convergence study in an idealized setting. Furthermore, we discuss the algorithmic aspects and the implementation of RL-RBF-G interpolation in a distributed-memory parallel framework, and present the results of a strong scalability test yielding nearly ideal results. Finally, we show the effectiveness of RL-RBF-G interpolation in multiphysics simulations by considering an application to a whole-heart cardiac electromecanics model.

Keywords: intergrid interpolation, radial basis functions, geodesic distance, multiphysics modeling, cardiac electromechanics

1. Introduction

We consider the problem of transferring data between two different meshes of the same spatial domain. This problem often arises in the context of finite elements multiphysics simulations [47, 52, 69, 70]. Indeed, two or more coupled problems defined on the same domain may have different accuracy requirements, which should be addressed with different, possibly non-matching discretizations, to balance computational cost and accuracy [25], or to allow the coupling of independent solvers [15]. In all these cases, it is therefore crucial to introduce suitable intergrid transfer operators to realize the coupling. Operators of this kind are relevant also for remeshing techniques or geometric multigrid methods [17, 37].

To this purpose, we consider rescaled localized radial basis function (RL-RBF) interpolation, introduced in [23] and previously employed for interface-coupled [22, 64] and volume-coupled [14, 55] problems. When constructing the RL-RBF interpolant, the closeness between points is measured using the Euclidean norm. This may prevent from capturing the correct geometry of the domain under consideration. Indeed, in realistic applications, disconnected regions of the domain may be close in the Euclidean norm (see e.g. Figure 1a). Therefore, an interpolant constructed by relying exclusively on the Euclidean distance will not account for
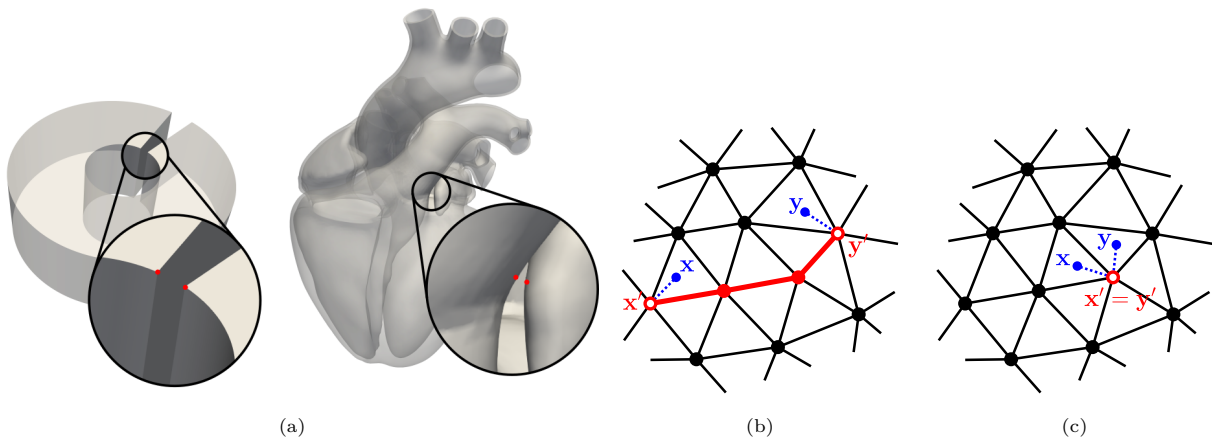

Figure 1: (a) Examples of domains with complex geometrical features. The red dots mark points on the domains that are close in the Euclidean metric, but that belong to topologically distant regions. The heart domain on the right is the Zygote heart model [72]. (b) A schematic representation of the discrete geodesic distance. (c) An example where, even if $\mathbf{x} \neq \mathbf{y}$, $g_h(\mathbf{x}, \mathbf{y}) = 0$.

the fact that those regions are disconnected. When considering data representing the solution of a physical problem, the points in regions of the domain that are not connected may feature very different solution values, which can lead to large artificial oscillations in the interpolant. These oscillations mostly appear in proximity to non-trivial geometrical features such as holes or cuts in the domain, as showcased by the examples of this paper.

We propose a modification to RL-RBF interpolation that relies on an approximation of the geodesic distance within the domain of interest to take its geometry into account. The geodesic distance is used to threshold the Euclidean distance, so that points that belong to disconnected regions of the domain are regarded as infinitely distant. We discuss the computational aspects of the proposed method by describing in detail the algorithms used to construct and evaluate the interpolant. We focus in particular on a distributed-memory parallel implementation, in which the interpolation and evaluation points are distributed across several processes.

Finally, we present numerical examples that illustrate the issues of RL-RBF interpolation in topologically complex domains, and we show that our geodesic distance thresholding method can overcome those issues. We consider both an idealized example and a realistic multiphysics application to three-dimensional whole-heart electromechanics [7, 8, 26, 29, 30, 32, 35, 36, 40, 59, 63, 65]. The numerical examples are used to assess the computational efficiency and parallel scalability of the proposed method.

This paper is structured as follows. In Section 2 we recall the RL-RBF intergrid transfer and describe the proposed modification leading to the RL-RBF-G method. In Section 3 we discuss the computational and algorithmic details of our implementation. Section 4 describes numerical experiments that showcase the properties of the proposed method. Finally, we draw some conclusive remarks in Section 5.

2. RBF intergrid transfer with geodesic thresholding

We briefly recall the RL-RBF interpolation procedure, and present our novel geodesic-based distance thresholding method to interpolate in complex domains accounting for their geometry.

2.1. Intergrid transfer using rescaled, localized RBF interpolation

Let $\Omega \subset \mathbb{R}^n$ be an open, bounded domain, and let $\mathcal{M}^{\mathrm{src}}$ and $\mathcal{M}^{\mathrm{dst}}$ be two meshes of Ω. The two meshes are not necessarily conforming or nested, and may feature different resolutions and different element types (tetrahedral or hexahedral). In other words, the two meshes represent two different approximations of the

domain Ω, which we will denote by Ω^{src} and Ω^{dst} respectively. We remark that, in general, the boundaries of Ω^{src} and Ω^{dst} do not necessarily coincide.

Let $f : \Omega^{\mathrm{src}} \to \mathbb{R}$ be a known function. Let $\{\mathbf{x}_i^{\mathrm{src}}\}_{i=1}^{N^{\mathrm{src}}}$, with $\mathbf{x}_i^{\mathrm{src}} \in \Omega^{\mathrm{src}}$, be a set of interpolation points, and $f_i^{\mathrm{src}} = f(\mathbf{x}_i^{\mathrm{src}})$ for $i = 1, \ldots, N^{\mathrm{src}}$. The radial basis function (RBF) interpolant of the function f at points $\mathbf{x}_i^{\mathrm{src}}$ is a function $\Pi_f : \mathbb{R}^n \to \mathbb{R}$ in the form

$$\Pi_f(\mathbf{x}) = \sum_{j=1}^{N^{\mathrm{src}}} \gamma_j \, \phi(d(\mathbf{x}, \mathbf{x}_j^{\mathrm{src}}), r_j) \,, \tag{1}$$

wherein $\gamma_j \in \mathbb{R}$ are unknown interpolation coefficients, $d : \mathbb{R}^n \times \mathbb{R}^n \to [0, +\infty)$ measures the distance between two points, and $\phi : \mathbb{R} \times [0, +\infty) \to [0, 1]$ is the compactly supported C^2-continuous Wendland basis function RBF [66] with support radius $r_j > 0$, defined as

$$\phi(t, r) = \max\left\{1 - \frac{t}{r}, 0\right\}^4 \left(1 + 4\frac{t}{r}\right) \quad \text{for } r \geq 0 \,. \tag{2}$$

The distance is typically measured in the Euclidean norm [15, 23, 28, 43, 64, 67], that is

$$d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\| \,. \tag{3}$$

However, as discussed in the following, this may introduce artificial oscillations when the domain features a complex geometry.

The coefficients γ_j in (1) are determined by imposing the interpolation constraints $\Pi_f(\mathbf{x}_i^{\mathrm{src}}) = f_i^{\mathrm{src}}$ for all $i = 1, \ldots, N^{\mathrm{src}}$, which is equivalent to solving the following linear system:

$$\Phi_{\mathrm{int}}\boldsymbol{\gamma} = \mathbf{f}^{\mathrm{src}} \,, \tag{4}$$

wherein $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \ldots, \gamma_{N^{\mathrm{src}}})^T$ is the vector of interpolation coefficients, the vector $\mathbf{f}^{\mathrm{src}} = (f_1^{\mathrm{src}}, f_2^{\mathrm{src}}, \ldots, f_{N^{\mathrm{src}}}^{\mathrm{src}})^T$ collects the evaluations of f on the interpolation points, and Φ_{int} is an $N^{\mathrm{src}} \times N^{\mathrm{src}}$ matrix of entries

$$(\Phi_{\mathrm{int}})_{ij} = \phi(d(\mathbf{x}_i^{\mathrm{src}}, \mathbf{x}_j^{\mathrm{src}}), r_j) \,. \tag{5}$$

The evaluation of the interpolant Π_f onto a set of points $\{\mathbf{x}_i^{\mathrm{dst}}\}_{i=1}^{N^{\mathrm{dst}}}$, with $\mathbf{x}_i^{\mathrm{dst}} \in \Omega^{\mathrm{dst}}$, can be obtained by computing the matrix-vector product $\mathbf{f}^{\mathrm{dst}} = \Phi_{\mathrm{eval}}\boldsymbol{\gamma}$, where $\mathbf{f}^{\mathrm{dst}} = (\Pi_f(\mathbf{x}_1^{\mathrm{dst}}), \Pi_f(\mathbf{x}_2^{\mathrm{dst}}), \ldots, \Pi_f(\mathbf{x}_{N^{\mathrm{dst}}}^{\mathrm{dst}}))^T$ and Φ_{eval} is an $N^{\mathrm{dst}} \times N^{\mathrm{src}}$ matrix of entries

$$(\Phi_{\mathrm{eval}})_{ij} = \phi(d(\mathbf{x}_i^{\mathrm{dst}}, \mathbf{x}_j^{\mathrm{src}}), r_j) \,. \tag{6}$$

The RBF interpolant may yield oscillations even if the interpolated data is constant [23]. To avoid this, we adopt the RL-RBF interpolation introduced in [23]. The RBF support radius r_j associated with interpolation node $\mathbf{x}_j^{\mathrm{src}}$ is chosen as the distance between $\mathbf{x}_j^{\mathrm{src}}$ and the M-th nearest neighbor in the set $\{\mathbf{x}_i^{\mathrm{src}}\}_{i=1}^{N^{\mathrm{src}}}$, magnified by a factor α [14, 64], where $M \in \mathbb{N}$ and $\alpha \in [1, \infty)$ are user-defined parameters. Furthermore, introducing the constant function $g(\mathbf{x}) = 1$, we rescale Π_f by the interpolant Π_g of g, leading to the following definition of the rescaled interpolant Π_f^{res}:

$$\Pi_f^{\mathrm{res}}(\mathbf{x}) = \frac{\Pi_f(\mathbf{x})}{\Pi_g(\mathbf{x})} \,, \tag{7}$$

To keep the notation light, we shall drop the superscript "res" from here on, and denote the rescaled interpolant simply by Π_f.

The definition of the interpolant and its evaluation generalize to vectorial functions by interpolating every component independently. We refer to [14, 23] for further details on RL-RBF interpolation, and to [20] for a discussion of its convergence properties.

2.2. Geodesic distance thresholding

The RBF interpolant is a linear combination of the interpolation coefficients γ_j, where the weights of the combination are larger the closer the evaluation point $\mathbf{x}$ is to the interpolation point $\mathbf{x}_j^{\text{src}}$. In qualitative terms, this means that the value of $\Pi_f(\mathbf{x})$ is influenced by the value of f at the interpolation points that are close to $\mathbf{x}$.

This may not be appropriate for complex domains that are characterized by disconnected regions, high curvature or cavities, such as those represented in Figure 1a. In those cases, the Euclidean distance may not be an adequate measure of the closeness between two points of Ω. This may lead to large artificial oscillations in the interpolant.

To avoid this issue, we need to replace the Euclidean distance (3) with a notion of distance that better accounts for the geometry of Ω. As a starting point, we consider the geodesic distance within Ω.

Definition 1 (Geodesic distance)**.** Given two points $\mathbf{x}, \mathbf{y} \in \Omega$, their geodesic distance $g(\mathbf{x}, \mathbf{y})$ is the length of the shortest curve contained in Ω and featuring $\mathbf{x}$ and $\mathbf{y}$ as its endpoints.

Since we consider discrete approximations of the domain Ω, we replace the geodesic distance defined above with a discrete counterpart. We begin by defining the distance between mesh vertices.

Definition 2 (Discrete geodesic distance)**.** Let $\mathcal{M}$ be a mesh representing an approximation $\widetilde{\Omega}$ of Ω. Given two points $\mathbf{x}, \mathbf{y} \in \widetilde{\Omega}$, vertices of $\mathcal{M}$, we define their discrete geodesic distance $g_h(\mathbf{x}, \mathbf{y})$ as the length of the shortest path from $\mathbf{x}$ to $\mathbf{y}$ traveling along edges and element diagonals of $\mathcal{M}$.

Applications of the interpolation procedure often require the interpolation or evaluation points to be internal to mesh elements (such as Gauss-Legendre quadrature points [14]). Furthermore, the evaluation of the interpolant requires us to compute the distance between interpolation points in Ω^{src} and evaluation points in Ω^{dst}. Therefore, we extend the above definition to arbitrary points in $\mathbb{R}^n$ as follows.

Definition 3 (Discrete geodesic distance)**.** Let $\mathcal{M}$ be a reference mesh approximating Ω. Given two points $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, we set $g_h(\mathbf{x}, \mathbf{y}) = g_h(\mathbf{x}', \mathbf{y}')$, where $\mathbf{x}'$ and $\mathbf{y}'$ are the vertices of $\mathcal{M}$ closest to $\mathbf{x}$ and $\mathbf{y}$, respectively.

Figure 1b displays a schematic representation of the above notions. The computation of the discrete geodesic distance g_h is computationally efficient and easy to implement [24, 38] (see also Section 3.4). However, g_h may be a poor approximation of the exact geodesic distance g [38, 39]. Furthermore, as shown in Figure 1c, it may happen that $g_h(\mathbf{x}, \mathbf{y}) = 0$ even if $\mathbf{x} \neq \mathbf{y}$, which may hinder the well-posedness of the interpolation problem [64]. Instead of resorting to more accurate yet more complex (and computationally demanding) discretization methods, we use g_h to apply a threshold to the Euclidean distance, replacing (3) with the following.

Definition 4 (Thresholded Euclidean distance)**.** Let $\mathcal{M}$ be a reference mesh approximating Ω. Given two points $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, we define

$$d(\mathbf{x}, \mathbf{y}; R) = \begin{cases} +\infty & \text{if } g_h(\mathbf{x}, \mathbf{y}) > R, \\ g_h(\mathbf{x}, \mathbf{y}) & \text{if } \beta h_{\max} + \|\mathbf{x} - \mathbf{y}\| < g_h(\mathbf{x}, \mathbf{y}) \leq R, \\ \|\mathbf{x} - \mathbf{y}\| & \text{if } g_h(\mathbf{x}, \mathbf{y}) \leq \beta h_{\max} + \|\mathbf{x} - \mathbf{y}\|, \end{cases} \tag{8}$$

where the discrete geodesic distance $g_h(\mathbf{x}, \mathbf{y})$ is evaluated with $\mathcal{M}$ as a reference mesh, $h_{\max}$ is the maximum diameter of the elements of $\mathcal{M}$, and $\beta > 0$ is a constant coefficient.

The coefficient β has the role of detecting regions of high curvature, where the discrete geodesic distance between points is significantly larger than the Euclidean distance. In those regions it is preferrable to compute $d(\mathbf{x}, \mathbf{y})$ through $g_h(\mathbf{x}, \mathbf{y})$, instead of the Euclidean distance $\|\mathbf{x} - \mathbf{y}\|$. Indeed, although g_h is a potentially inaccurate approximation of g, this approximation will generally be better than the Euclidean distance. Conversely, in regions of low curvature, the Euclidean distance enjoys better smoothness properties than g_h, and is therefore a better choice for evaluating $d(\mathbf{x}, \mathbf{y})$.

Finally, we replace the definition of the interpolant in (1) with

$$\Pi_f(\mathbf{x}) = \sum_{j=1}^{N^{\mathrm{src}}} \gamma_j \, \phi(d(\mathbf{x}, \mathbf{x}_j^{\mathrm{src}}; r_j), r_j) \, , \tag{9}$$

where the threshold R of (8) is chosen equal to the RBF support radius. The entries of Φ_{int} and Φ_{eval} are updated accordingly. To evaluate the discrete geodesic distance, we use the finest of the two meshes $\mathcal{M}^{\mathrm{src}}$ and $\mathcal{M}^{\mathrm{dst}}$, to obtain the best approximation. We refer to the resulting method as RL-RBF-G interpolation.

3. Algorithms and parallel implementation

We describe in this section the algorithms and data structures implemented to construct and evaluate the RL-RBF-G interpolant, discussing computational cost and parallel scalability. Our implementation is designed for a distributed-memory parallel architecture, based on the MPI standard.

All numerical methods presented here are available as open source within the core module of `life`$^{\mathrm{x}}$ [1, 41], a library based on `deal.II` [5, 6, 21] for finite element simulations of the cardiac function [2–4].

The algorithm presented in this section significantly improves upon the one we previously discussed in [14]. The major advancements are the use of R-trees to perform spatial queries and a significant reduction in inter-process communication. We report in Algorithm 1 an overview of the proposed algorithm. The rest of this section describes in more detail its individual steps.

3.1. Coarse representations of local points

Since we work in a distributed-memory parallel framework, each process r only stores a subset X_r^{src} of the source points $\{\mathbf{x}_i^{\mathrm{src}}\}_{i=1}^{N^{\mathrm{src}}}$. However, when building the interpolation matrix Φ_{int}, each process needs to evaluate the distance of each of the points it owns from all other points, including some that are owned by other processes. Those points must therefore be retrieved through inter-process communication.

To do so, we build for every process r a coarse representation $B_r \subset \mathbb{R}^n$ such that $X_r^{\mathrm{src}} \subset B_r$. We define B_r as the union of multiple axis-aligned bounding boxes, obtained by successively splitting in half the axis-aligned box containing all the points. Then, the coarse representation is communicated to all other processes. Finally, each process r uses the coarse representation of every other process p to determine which of the points X_r^{src} must be communicated to p. This communication is obtained by means of a point-to-point communication algorithm as implemented by `deal.II`'s `Utilities::MPI::some_to_some`. Then, each process r merges all the points received from other processes into a single set Y_r^{src} (Algorithm 1, line 13).

The evaluation points $\{\mathbf{x}_i^{\mathrm{dst}}\}_{i=1}^{N^{\mathrm{dst}}}$ are treated in a similar way. The corresponding set of points owned by process r is denoted by X_r^{dst}, and the set of points stored by process r after inter-process communication is denoted by Y_r^{dst}. We remark that this procedure does not assume that the meshes $\mathcal{M}^{\mathrm{src}}$ and $\mathcal{M}^{\mathrm{dst}}$ are partitioned in a consistent way.

This step corresponds to lines 1–14 of Algorithm 1.

3.2. Finding the M-th nearest neighbor to a point

In line 17 of Algorithm 1, we retrieve the M-th nearest neighbor to point $\mathbf{x}_j^{\mathrm{src}}$ within the set Y_r^{src}. This is achieved by iteratively building a binary max-heap [68] of M elements. The point $\mathbf{z}$ is the root node of the heap. We remark that this task could be optimized by resorting to an M-tree data structure [16].

Furthermore, the process of finding the M-th nearest neighbor is made faster by assuming that it is no further than a user-specified threshold $r_{\max}$. While an optimal value for $r_{\max}$ depends on M and α, we empirically selected it to be 10 times the average diameter $h_{\mathrm{avg}}^{\mathrm{src}}$ of elements of $\mathcal{M}^{\mathrm{src}}$.

Algorithm 1 Parallel assembly of the matrices Φ_{int} and Φ_{eval}. The index r denotes the rank of the current process in the MPI parallel communicator.

Input: sets of locally owned points X_r^{src} and X_r^{dst}
Input: $h_{\max}^{\text{src}}$, maximum diameter of elements of $\mathcal{M}^{\text{src}}$
Output: matrices Φ_{int} and Φ_{eval}

1: construct a coarse representation B_r of the local points X_r^{src}
2: send B_r to every parallel process different than r
3:
4: $Y_{rr}^{\text{src}} \leftarrow X_r^{\text{src}}$
5: $Y_{rr}^{\text{dst}} \leftarrow X_r^{\text{dst}}$
6: **for** each parallel process p different than r **do**
7: $\quad$ $Y_{rp}^{\text{src}} \leftarrow X_r^{\text{src}} \cap B_p$
8: $\quad$ send the points Y_{rp}^{src} to process p
9:
10: $\quad$ $Y_{rp}^{\text{dst}} \leftarrow X_r^{\text{dst}} \cap B_p$
11: $\quad$ send the points Y_{rp}^{dst} to process p
12: **end for**
13: $Y_r^{\text{src}} \leftarrow \cup_{p=1}^{N\text{proc}} Y_{pr}^{\text{src}}$
14: $Y_r^{\text{dst}} \leftarrow \cup_{p=1}^{N\text{proc}} Y_{pr}^{\text{dst}}$
15:
16: **for** every point $\mathbf{x}_j^{\text{src}} \in X_r^{\text{src}}$ **do**
17: $\quad$ find $\mathbf{z}$, the M-th nearest neighbor (with respect to g_h) to $\mathbf{x}_j^{\text{src}}$ in Y_r^{src}
18: $\quad$ $r_j \leftarrow \max\left\{ \alpha g_h(\mathbf{x}_j^{\text{src}}, \mathbf{z}), r_{\max} \right\}$
19:
20: $\quad$ compute the axis-aligned bounding box C_j centered at $\mathbf{x}_j^{\text{src}}$ and of dimension $2(r_j + 2h_{\max}^{\text{src}})$
21:
22: $\quad$ $Z^{\text{src}} \leftarrow Y_r^{\text{src}} \cap C_j$
23: $\quad$ **for** every point $\mathbf{x}_i^{\text{src}} \in Z^{\text{src}}$ **do**
24: $\quad\quad$ **if** $g_h(\mathbf{x}_j^{\text{src}}, \mathbf{x}_i^{\text{src}}) \le r_j$ **then**
25: $\quad\quad\quad$ $(\Phi_{\text{int}})_{ij} \leftarrow \phi(d(\mathbf{x}_j^{\text{src}}, \mathbf{x}_i^{\text{src}}; r_j), r_j)$
26: $\quad\quad$ **end if**
27: $\quad$ **end for**
28:
29: $\quad$ $Z^{\text{dst}} \leftarrow Y_r^{\text{dst}} \cap C_j$
30: $\quad$ **for** every point $\mathbf{x}_i^{\text{dst}} \in Z^{\text{dst}}$ **do**
31: $\quad\quad$ **if** $g_h(\mathbf{x}_j^{\text{src}}, \mathbf{x}_i^{\text{dst}}) \le r_j$ **then**
32: $\quad\quad\quad$ $(\Phi_{\text{eval}})_{ij} \leftarrow \phi(d(\mathbf{x}_j^{\text{src}}, \mathbf{x}_i^{\text{dst}}; r_j), r_j)$
33: $\quad\quad$ **end if**
34: $\quad$ **end for**
35: **end for**

3.3. Finding points sufficiently close to $\mathbf{x}_j^{src}$

Lines 20–22 of Algorithm 1 have the purpose of quickly excluding the points of Y_r^{src} that are sufficiently far from $\mathbf{x}_j^{\text{src}}$. To do so, we observe that, with the same notation of Section 2.2, the following holds:

$$\|\mathbf{x} - \mathbf{y}\| \le \|\mathbf{x} - \mathbf{x}'\| + \|\mathbf{x}' - \mathbf{y}'\| + \|\mathbf{y} - \mathbf{y}'\| \tag{10}$$

$$\le h_{\max}^{\text{src}} + g_h(\mathbf{x}, \mathbf{y}) + h_{\max}^{\text{src}} \tag{11}$$

$$= g_h(\mathbf{x}, \mathbf{y}) + 2h_{\max}^{\text{src}}, \tag{12}$$

6

where $h_{\max}^{\mathrm{src}}$ is the maximum diameter of the elements of $\mathcal{M}^{\mathrm{src}}$. Now, let C_j be an axis-aligned bounding box centered at $\mathbf{x}_j^{\mathrm{src}}$ of dimension $2(r_j + 2h_{\max}^{\mathrm{src}})$ along all coordinate directions (line 20). If a point $\mathbf{x}_i^{\mathrm{src}}$ lies outside the box, there holds

$$\|\mathbf{x}_j^{\mathrm{src}} - \mathbf{x}_i^{\mathrm{src}}\| \geq r_j + 2h_{\max}^{\mathrm{src}} \,, \tag{13}$$

$$g_h(\mathbf{x}_j^{\mathrm{src}}, \mathbf{x}_i^{\mathrm{src}}) \geq \|\mathbf{x}_j^{\mathrm{src}} - \mathbf{x}_i^{\mathrm{src}}\| - 2h_{\max}^{\mathrm{src}} \geq r_j \,, \tag{14}$$

so that the associated entry $(\Phi_{\mathrm{int}})_{ij}$ in the interpolation matrix is zero. Consequently, we can discard all points that lie outside of the bounding box, without explicitly computing their discrete geodesic distance from $\mathbf{x}_j^{\mathrm{src}}$.

This is especially convenient because the intersection operation of line 22 can be performed very efficiently by storing the points Y_r^{src} into an R-tree structure [33, 42]. We rely on the R-tree implementation offered by the Boost library [12].

3.4. Evaluating the discrete geodesic distance

We use Dijkstra's shortest path algorithm [24] to evaluate g_h, as described in Algorithm 2. As usually done, the set of visited points A is stored in a binary min-heap data structure [68], providing constant-time retrieval of the element with minimum tentative distance (Algorithm 2, line 9).

Since we only use g_h to threshold the Euclidean distance in (8), we stop the algorithm if it reaches nodes that are further than the threshold R from the starting point (Algorithm 2, line 21). Similarly, if the two endpoints are further than R in Euclidean distance, we do not compute their discrete geodesic distance (Algorithm 2, lines 1–3). In both cases, we return an arbitrary value greater than or equal to R.

The loop of Algorithm 1, lines 16–35, requires to evaluate $g_h(\mathbf{x}_j^{\mathrm{src}}, \mathbf{y})$ multiple times with $\mathbf{x}_j^{\mathrm{src}}$ as first argument and varying the second argument $\mathbf{y}$. To avoid unnecessary computations, the sets A and B and the tentative distances $s_{\mathbf{z}}$ are reused between calls. If the point $\mathbf{y}$ was already added to the set B during a previous evaluation of the distance, we reuse the already computed distance $s_{\mathbf{y}}$.

When running in parallel, the mesh $\mathcal{M}$ is distributed over multiple processes, and each of the processes stores only a subset of the mesh vertices. However, each process needs to evaluate the discrete geodesic distance between arbitrary mesh vertices, regardless of whether they are owned by that process or not. A naive solution would be to send all the vertices and their adjacency information to every process. This however does not scale efficiently over large meshes or large number of processes.

To avoid this issue, we observe that, when evaluating g_h, each process r always passes a locally owned point $\mathbf{x}_j^{\mathrm{src}} \in X_r^{\mathrm{src}}$ as first argument (see Algorithm 1, lines 18, 24 and 31). Furthermore, we are only interested in destination points no further than the maximum RBF support radius $r_{\max}$. Therefore, it is sufficient that each process stores the points that are within $r_{\max}$ of any locally owned point, together with the associated adjacency information.

3.5. Evaluating the interpolant

After constructing the matrices Φ_{int} and Φ_{eval}, given an input data vector $\mathbf{f}$, the interpolation coefficients are obtained by solving the linear system (4) with the preconditioned GMRES method [53]. For the numerical experiments of this paper, we consider an incomplete LU (ILU) factorization preconditioner with 2 levels of fill-in and 1 level of inter-process overlap, as implemented by Trilinos IFPACK [54] and wrapped by `deal.II`. Alternatively, one can consider an ad-hoc preconditioner based on approximate cardinal basis functions [10, 13, 14], although in our numerical experiments it performed less effectively than the ILU preconditioner.

4. Numerical experiments

In the following sections, we examine the properties of the proposed interpolation method by means of numerical experiments, and compare it with RL-RBF interpolation. Unless otherwise specified, we set $\beta = 0.5$. The role of the high-curvature threshold is further discussed in Section 4.2.2.

Algorithm 2 Dijkstra's shortest path algorithm, adapted from [24].

Input: reference mesh $\mathcal{M}$
Input: start and end mesh vertices $\mathbf{x}$, $\mathbf{y}$
Input: threshold R
Output: returns $g_h(\mathbf{x}, \mathbf{y})$

1: **if** $\|\mathbf{x} - \mathbf{y}\| > R$ **then**
2:      **return** R
3: **end if**
4:
5: add $\mathbf{x}$ to the set A of visited points
6: set the tentative distance for $\mathbf{x}$: $s_{\mathbf{x}} \leftarrow 0$
7:
8: **while** A is not empty **do**
9:      find the point $\mathbf{z}$ in A for which the tentative distance $s_{\mathbf{z}}$ is minimum
10:      remove $\mathbf{z}$ from A and add it to the set of closed points B
11:
12:      **for** each neighbor $\mathbf{w}$ of $\mathbf{z}$ that is not in B **do**
13:          **if** $\mathbf{w} \in A$ **then**
14:              $s_{\mathbf{w}} \leftarrow \min\{s_{\mathbf{z}} + \|\mathbf{w} - \mathbf{z}\|, s_{\mathbf{w}}\}$
15:          **else**
16:              add $\mathbf{w}$ to A
17:              $s_{\mathbf{w}} \leftarrow s_{\mathbf{z}} + \|\mathbf{w} - \mathbf{z}\|$
18:          **end if**
19:      **end for**
20:
21:      **if** $\mathbf{z} = \mathbf{y}$ or $s_{\mathbf{z}} > R$ **then**
22:          **return** $s_{\mathbf{z}}$
23:      **end if**
24: **end while**

4.1. Idealized benchmark

We consider an idealized benchmark on the domain depicted in Figure 2a. The domain represents a toroid with square section, from which a thin slice has been removed, opening a slit. We interpolate the following function:

$$f(x,y,z) = \arctan_2(z, -x) = \begin{cases} \arctan\left(-\dfrac{z}{x}\right) & \text{if } x < 0 \,, \\ \arctan\left(-\dfrac{z}{x}\right) + \pi & \text{if } x > 0 \text{ and } z \geq 0 \,, \\ \arctan\left(-\dfrac{z}{x}\right) - \pi & \text{if } x > 0 \text{ and } z < 0 \,, \\ \dfrac{\pi}{2} & \text{if } x = 0 \text{ and } z > 0 \,, \\ -\dfrac{\pi}{2} & \text{if } x = 0 \text{ and } z < 0 \,, \\ \text{undefined} & \text{if } x = z = 0 \,, \end{cases} \tag{15}$$

which is discontinuous across the slit, as shown in Figure 2b. We interpolate the function f as described in previous sections, both with and without geodesic distance thresholding. Table 1 lists the meshes considered throughout this section.

Figure 2c shows an example of RL-RBF interpolant, using the mesh $\mathcal{M}_{\mathrm{ring},1}^{\mathrm{tet}}$ and setting $M = 6$, $\alpha = 2$. Close to the slit, the interpolant features large oscillations, significantly departing from the original function
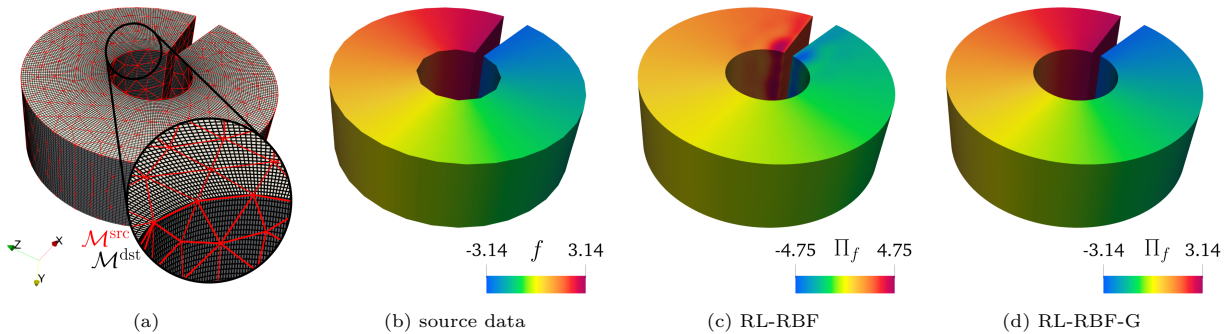
Figure 2: Idealized benchmark (Section 4.1). (a) Interpolation domain Ω, overlaid with the meshes $\mathcal{M}^{\mathrm{src}} = \mathcal{M}^{\mathrm{tet}}_{\mathrm{ring},1}$ (red) and $\mathcal{M}^{\mathrm{dst}} = \mathcal{M}^{\mathrm{hex}}_{\mathrm{ring},1}$ (black). (b) Source data $f(x,y,z) = \arctan_2(z, -x)$. (c) Interpolant obtained without geodesic thresholding. Notice the different colorbar scale and the spurious oscillations close to the slit. (d) Interpolant obtained from mesh $\mathcal{M}^{\mathrm{tet}}_{\mathrm{ring},1}$ with geodesic thresholding.

f. In intuitive terms, this is due to f having very different values on the two sides of the slit. Without geodesic thresholding, the interpolant does not take the slit into account, and the oscillations result from trying to fit the values on both sides. On the contrary, constructing the interpolant with geodesic thresholding, with the same mesh and parameters, leads to a much more accurate result, as shown in Figure 2d.

4.1.1. Convergence test

We assess the convergence of the proposed method by constructing the interpolant of f on meshes $\mathcal{M}^{\mathrm{tet}}_{\mathrm{ring},i}$, for $i = 1, 2, \ldots, 7$, and evaluating it on $\mathcal{M}^{\mathrm{hex}}_{\mathrm{ring},1}$ (see Table 1), both with and without geodesic thresholding. We measure the error between the interpolant and f with the following approximation of the relative $L^\infty(\Omega)$ norm:

$$e_\infty = \frac{\max_{i=1,\ldots,N^{\mathrm{dst}}} \left| \mathbf{f}^{\mathrm{dst}}_i - f(\mathbf{x}^{\mathrm{dst}}_i) \right|}{\max_{i=1,\ldots,N^{\mathrm{dst}}} \left| f(\mathbf{x}^{\mathrm{dst}}_i) \right|} \, . \tag{16}$$

Figure 3 reports the error obtained for $M = 1$ and $M = 4$ with different choices of α, both with and without geodesic thresholding. For relatively large values of $h^{\mathrm{src}}_{\max}$, RL-RBF-G interpolation reduces the interpolation error approximately by one order of magnitude with respect to RL-RBF. Indeed, the latter is largely affected by the oscillations appearing near the slit.

The coefficient α rescales the RBF support radius. For large values of $h^{\mathrm{src}}_{\max}$, increasing α without geodesic thresholding allows points on one side of the slit to interact more with points on the other, leading to larger errors. Conversely, increasing α with geodesic thresholding reduces the error, since a larger RBF support radius removes small spurious oscillations in the interpolant, improving its accuracy.

For a sufficiently small $h^{\mathrm{src}}_{\max}$, RL-RBF and RL-RBF-G lead to approximately equal interpolation errors. This happens because the interpolation radii r_j are progressively reduced as the mesh is refined, so that for a sufficiently fine mesh the interpolation radius is small enough that points on one side of the slit do not interact with points on the other side. In that case, geodesic thresholding has no effect in practice, and the two methods yield almost identical results.

Finally, our numerical results indicate that the RL-RBF-G interpolant converges to f with approximate order 1 with respect to $h^{\mathrm{src}}_{\max}$. This is in agreement with the analysis presented in [20] for RL-RBF interpolation.

4.1.2. Computational efficiency and scalability

In the same setting as previous section, we run a strong scalability test by computing the interpolant from mesh $\mathcal{M}^{\mathrm{tet}}_{\mathrm{ring},8}$ onto mesh $\mathcal{M}^{\mathrm{hex}}_{\mathrm{ring}}$ varying the number of parallel processes. We measure the wall time

9

Domain	Mesh	Element type	# vertices	$h_{\min}$[mm]	h_{avg}[mm]	$h_{\max}$[mm]
ring	$\mathcal{M}_{\mathrm{ring},1}^{\mathrm{tet}}$	tet	608	25.00	40.02	55.88
ring	$\mathcal{M}_{\mathrm{ring},2}^{\mathrm{tet}}$	tet	1166	19.48	31.28	41.79
ring	$\mathcal{M}_{\mathrm{ring},3}^{\mathrm{tet}}$	tet	2453	13.41	23.30	31.12
ring	$\mathcal{M}_{\mathrm{ring},4}^{\mathrm{tet}}$	tet	6403	9.20	16.28	22.66
ring	$\mathcal{M}_{\mathrm{ring},5}^{\mathrm{tet}}$	tet	14957	7.14	11.82	15.38
ring	$\mathcal{M}_{\mathrm{ring},6}^{\mathrm{tet}}$	tet	42327	4.87	8.14	11.04
ring	$\mathcal{M}_{\mathrm{ring},7}^{\mathrm{tet}}$	tet	304313	2.47	4.08	5.64
ring	$\mathcal{M}_{\mathrm{ring},8}^{\mathrm{tet}}$	tet	4411070	0.99	1.64	2.36
ring	$\mathcal{M}_{\mathrm{ring}}^{\mathrm{hex}}$	hex	418569	3.73	4.33	5.17
heart	$\mathcal{M}_{\mathrm{heart}}^{\mathrm{fine}}$	tet	1260388	0.36	0.87	3.11
heart	$\mathcal{M}_{\mathrm{heart}}^{\mathrm{coarse}}$	tet	67553	0.82	2.51	4.81

Table 1: Meshes used in the numerical experiments. For each mesh, we report the type of elements (hexahedra or tetrahedra), the number of nodes, and the minimum, average and maximum element diameter h.

needed for the different stages of the construction of the interpolant and its evaluation. The scalability test was run on the GALILEO100 supercomputer[1] from the CINECA supercomputing center (Italy).

The results are shown in Figure 4, where we report the computational time associated with the wall time taken by the interpolant construction (which includes the initialization of the adjacency graph for evaluating the discrete geodesic distance, the construction of the matrices Φ_{int} and Φ_{eval}, the initialization of the preconditioner for the system (4), and the evaluation of Π_g for rescaling) and the time needed for the evaluation of the interpolant (i.e. for the solution of the linear system (4)). We also report separately the time to assemble Φ_{int}, Φ_{eval} and the preconditioner ("matrix assembly" in the plot) and the time required to initialize the adjacency graph for geodesic distance ("setup of geodesic distance" in the plot).

The time needed to assemble matrices scales ideally up to 3072 cores. The scalability of the overall interpolant construction is ideal up to around 768 cores (corresponding to approximately 5500 interpolation points per core), then starts to deteriorate due to parallel communication and load balancing issues.

The solution of the interpolation system (4) exhibits relatively poor scalability properties, mostly due to the scalability of the preconditioner. Furthermore, the time needed for the initialization of the discrete geodesic distance data structures is essentially independent of the number of cores. Nonetheless, both these steps have a negligible impact on the overall computational cost.

4.2. Cardiac electromechanics test

With the aim of providing an example of application to a challenging multiphysics problem of practical interest, we consider a numerical model of whole-heart cardiac electromechanics described in [26]. The term cardiac electromechanics refers to the interplay of electrical excitation (known as cardiac electrophysiology, [18, 31, 46, 60, 61]), active contraction and passive mechanical response of the cardiac muscle during the heartbeat [8, 26, 29, 32, 47–50, 52, 58, 59]. Such models have the potential of supporting the understanding and clinical decision making associated with cardiovascular pathologies [45, 56, 57].

Cardiac electromechanics is intrinsically multiphysics and multiscale. Indeed, electrophysiology is characterized by much smaller spatial and temporal scales than muscular mechanics (both active and passive). For this reason, the electrophysiology model is typically solved on a finer discretization than mechanics, both in time and in space, by combining temporal staggering and intergrid transfer operators [14, 26, 30, 47, 52, 55, 63].

The heart is composed of four different interconnected chambers, and it is connected to large blood vessels (including the aorta, the pulmonary artery), resulting in a topologically complex domain (see e.g.

---

[1]Technical specifications: `https://wiki.u-gov.it/confluence/display/SCAIUS/UG3.3%3A+GALILEO100+UserGuide`.
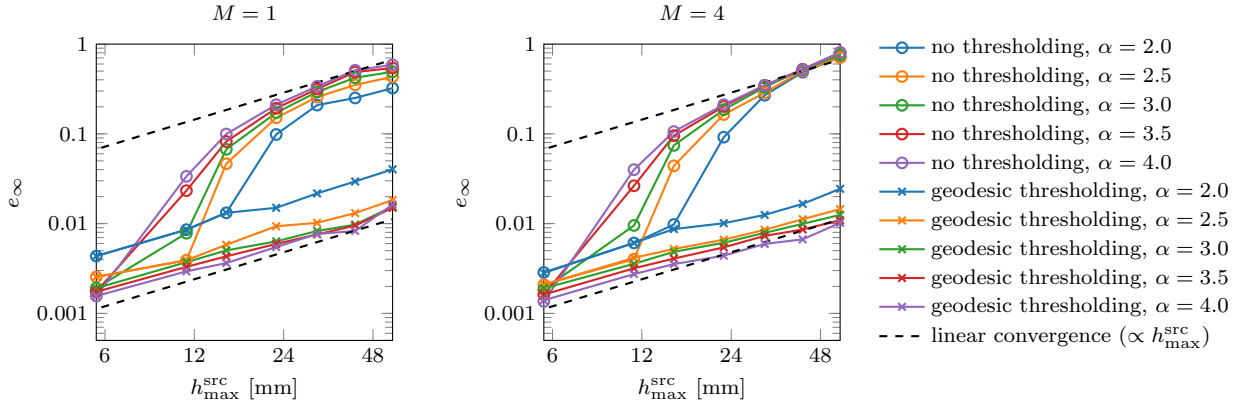
Figure 3: Idealized benchmark, convergence test (Section 4.1.1). Interpolation error, defined as in (16), against the maximum element diameter $h_{\max}^{\mathrm{src}}$ of the source mesh $\mathcal{M}^{\mathrm{src}}$. The interpolant was computed setting $M = 1$ (left) and $M = 4$ (right), for different values of α. Different choices of α correspond to different colors, while circles and crosses correspond to interpolants without and with geodesic thresholding, respectively.

Figure 1a). It is therefore crucial to use intergrid transfer operators that account for this complexity. To this end, we apply the newly introduced RL-RBF-G method in the electromechanical framework.

The whole-heart electromechanical model features the following components (see also Figure 5) [26]:

- the monodomain model of cardiac electrophysiology, for the spatial propagation of electrical excitation [18]. The model includes geometry-dependent mechano-electric feedback (MEF) effects: denoting by **d** the muscular displacement, the monodomain model accounts for the effect of the deformation onto the electrical conductivity, through $\mathbf{F} = \mathbf{I} + \boldsymbol{\nabla}\mathbf{d}$ and $J = \det \mathbf{F}$ [14, 29, 57];

- the ten Tusscher-Panfilov (TTP06) and Courtemanche-Ramirez-Nattel (CRN) ionic models, describing the electrical activity at the cellular scale [18, 19, 62]. The ionic models regulate in particular the evolution of the intracellular calcium concentration $[\mathrm{Ca}^{2+}]_\mathrm{i}$;

- the RDQ20-MF model of muscular contraction, that computes an active stress tensor $\mathbf{P}_{\mathrm{act}}$ in response to variations in $[\mathrm{Ca}^{2+}]_\mathrm{i}$ [51];

- the elastodynamics equation for muscular deformation, including both passive and active stress contributions [52];

- a lumped-parameter model of the circulatory system, describing pressures and flowrates through several circulation compartments by means of a system of ordinary differential equations (ODEs) [11, 26, 35].

The model equations are reported in full in Appendix A. We refer to [26] for further details on the electromechanical model and strategies for its numerical discretization, and to [71] for a discussion on its calibration.

Similarly to [14, 47, 52, 55], we solve the ionic and monodomain models on a fine mesh $\mathcal{M}_{\mathrm{heart}}^{\mathrm{fine}}$, and the force generation and mechanics models on a coarser mesh $\mathcal{M}_{\mathrm{heart}}^{\mathrm{coarse}}$ (see Table 1). We interpolate the calcium concentration $[\mathrm{Ca}^{2+}]_\mathrm{i}$, computed on the fine mesh by the ionic model, onto the coarse mesh, where it is needed by the force generation model. Conversely, we interpolate the deformation gradient **F** computed by solving the elastodynamics equation from the coarse to the fine mesh, using the method presented in [14], wherein we replace the RL-RBF interpolation with the newly introduced RL-RBF-G method. For the sake of post-processing, we also interpolate the displacement field **d** from the coarse to the fine mesh.

We run the simulation up to a final time of $T = 4.8\,\mathrm{s}$, corresponding to a total of six heartbeats, with each heartbeat lasting $t_{\mathrm{HB}} = 0.8\,\mathrm{s}$. We report results for the last hearbeat only (from $t = 4.0\,\mathrm{s}$ to $t = 4.8\,\mathrm{s}$). We consider the segregated-staggered scheme previously discussed in [26, 47, 52]. We set a time discretization step of $\Delta t_{\mathrm{EP}} = 5 \cdot 10^{-5}\,\mathrm{s}$ for the monodomain and ionic models, while force generation, muscular mechanics and circulation are solved with a time discretization step of $\Delta t_{\mathrm{M}} = 20\Delta t_{\mathrm{EP}} = 10^{-3}\,\mathrm{s}$. The simulation is
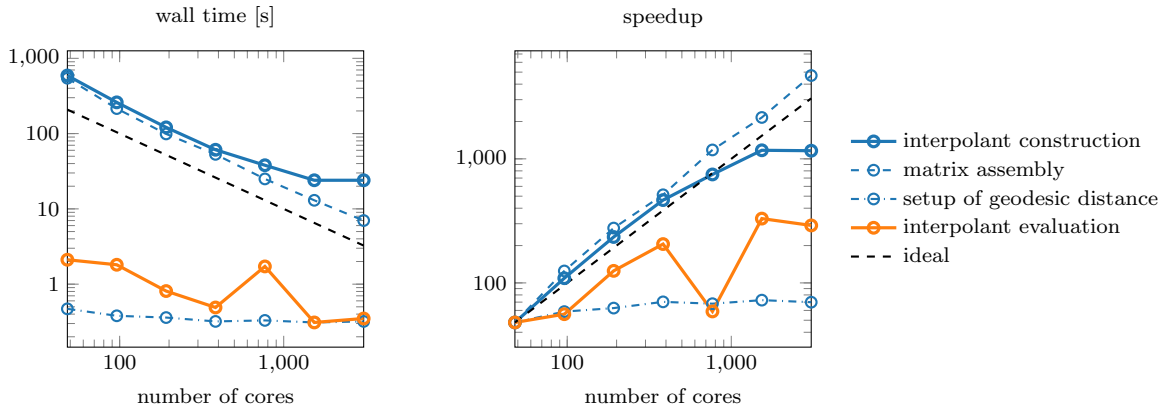
Figure 4: Idealized benchmark, scalability test (Section 4.1.2). Total wall-time (left) and parallel speedup for different steps of the construction and evaluation of the RL-RBF-G interpolant.

run using 92 cores from the iHEART computing node[2] available at MOX, Dipartimento di Matematica, Politecnico di Milano.

The two meshes have a significantly different size, with the fine mesh having almost 20 times as many nodes as the coarse one. Nonetheless, our numerical results indicate that the RL-RBF-G method allows to reliably transfer information between the two meshes. No spurious oscillations are observed in the interpolated data, and the overall results are in agreement with those of [26], as seen in Figures 6 to 8, reporting the evolution of $[Ca^{2+}]_i$, $\mathbf{d}$ and the pressure-volume loops of the four cardiac chambers.

4.2.1. Computational cost of interpolation

Table 2 reports a breakdown of the computational cost associated with the simulation, using either RL-RBF-G or RL-RBF interpolations. The assembly of interpolation matrices takes up approximately 17 % of the total cost of the initialization (which is mostly spent in computing the initial conditions for the mechanics problem). Interpolating the deformation gradient tensor $\mathbf{F}$ from the coarse to the fine mesh requires approximately 12 % of the average time needed to solve one electromechanics time step (comprising 20 time steps of electrophysiology, one time step of force generation and one of mechanics, one fine-to-coarse interpolation and one coarse-to-fine interpolation). We remark that, as described in [14], this entails the interpolation of 11 scalar fields that define the tensor $\mathbf{F}$. Finally, the interpolation of $[Ca^{2+}]_i$ from the coarse to the fine mesh has a very small impact on the cost of one electromechanics time step.

The cost of initializing RL-RBF interpolation is significantly lower than the one of RL-RBF-G. This is largely due to the need to compute geodesic distances in constructing the interpolation matrices with RL-RBF-G. Furthermore, evaluating the interpolant of $\mathbf{F}$ with RL-RBF-G is more costly than with RL-RBF, due to a large number of linear solver iterations in solving (4). Nonetheless, the large oscillations appearing in the RL-RBF interpolant prevent its use for this application.

Although not negligible, the overall cost of constructing and evaluating the RL-RBF-G interpolants does not dominate the wall time of the simulations. We believe this overhead is justified by the discretization flexibility and the reliability that the proposed interpolation method offers. Indeed, the RL-RBF-G interpolation allows to model cardiac electrophysiology to high accuracy without impacting the computational cost associated with the mechanics submodel (contrarily to the strategy of [26]). Therefore, it can support simulations where the electrophysiological accuracy is critical, e.g. including spatial heterogeneities and electrical dysfunctions [34, 44, 56].

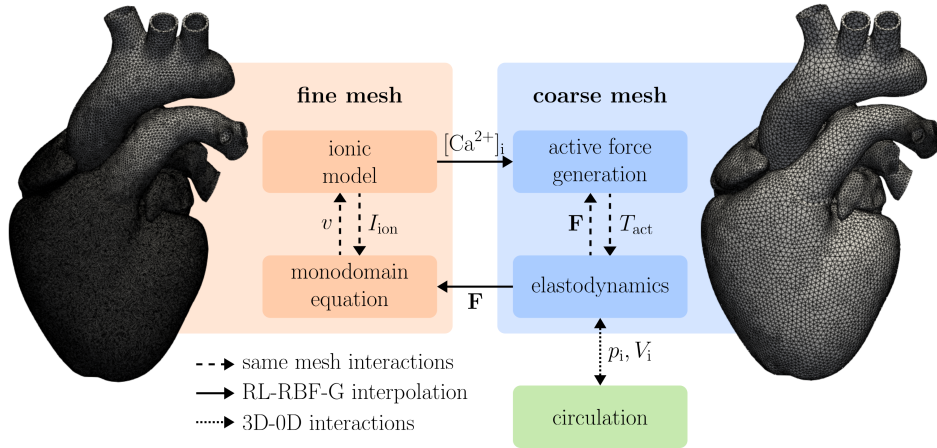[2]The node runs eight processors Intel Xeon Platinum 8160 @2.1 GHz.

Figure 5: Schematic representation of the electromechanics model of Section 4.2, reporting the multiphysics interactions.
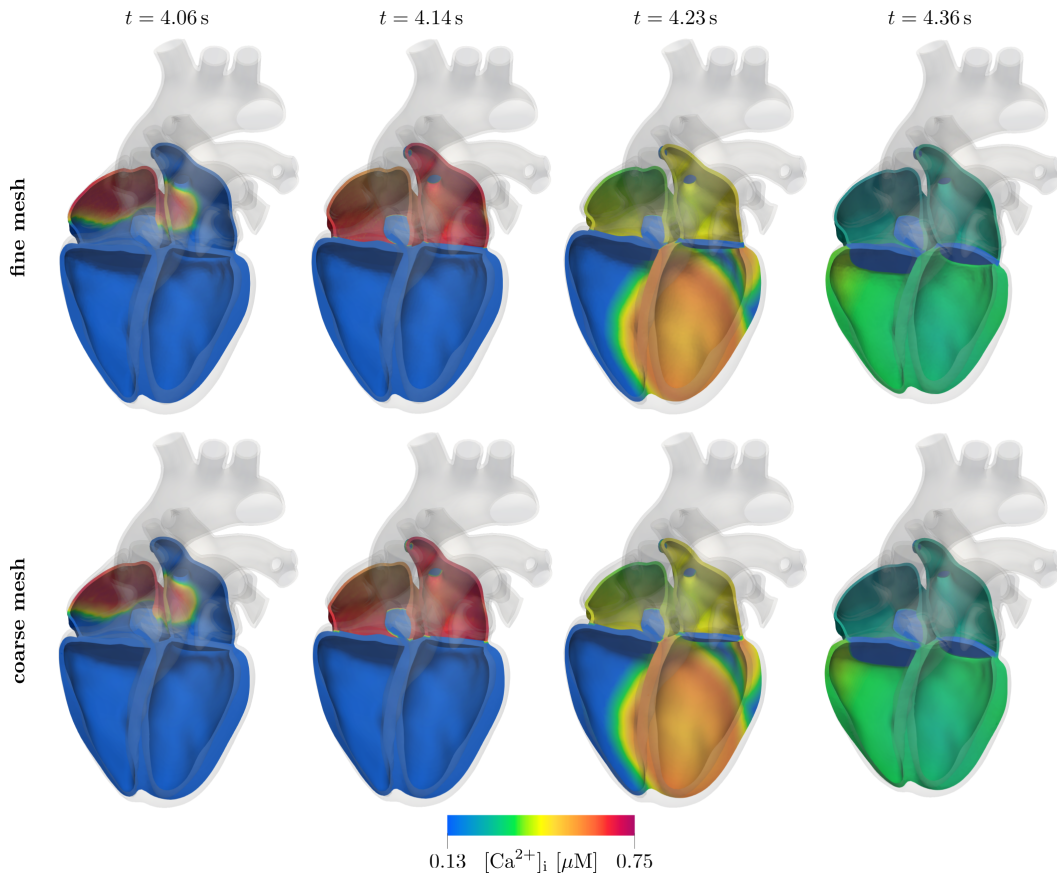

Figure 6: Whole-heart electromechanics (Section 4.2). Intracellular calcium concentration $[\mathrm{Ca}^{2+}]_i$ over time on the electrophysiology (top) and mechanics (bottom) meshes. The domain is warped according to the displacement $\mathbf{d}$.
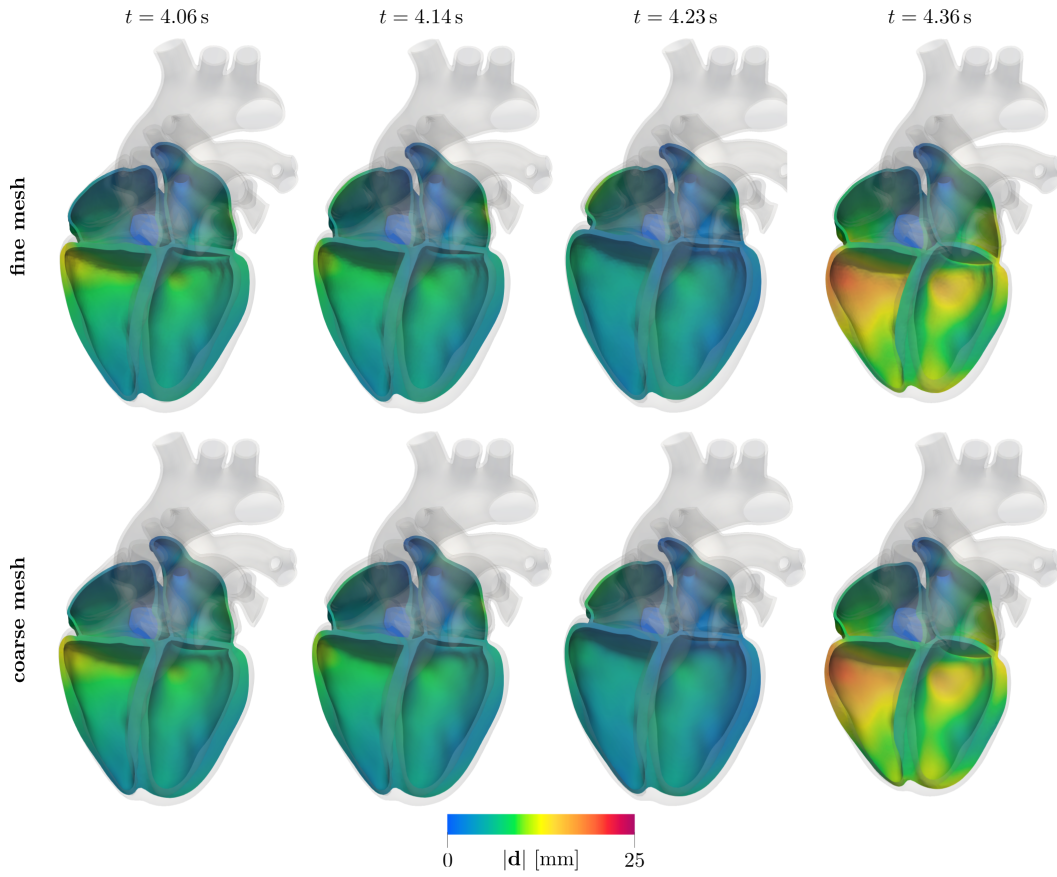
Figure 7: Whole-heart electromechanics (Section 4.2). Muscular displacement **d** over time on the electrophysiology (top) and mechanics (bottom) meshes. We remark that the displacement is interpolated for post-processing, while the monodomain equation relies on interpolating the deformation gradient **F**. The domain is warped according to **d**.
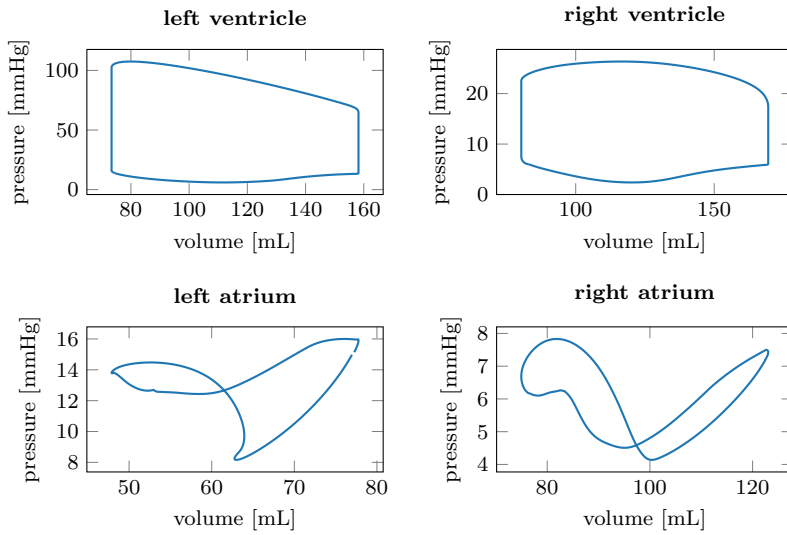

Figure 8: Whole-heart electromechanics (Section 4.2). Pressure-volume loops of the four cardiac chambers.

	RL-RBF-G		RL-RBF	
	wall time [s]	**relative** [%]	**wall time** [s]	**relative** [%]
simulation setup	2940	100.0	2300	100.0
mechanics	2129	72.4	1994	86.7
fine-to-coarse interpolant	266	9.1	23	1.0
coarse-to-fine interpolant	226	7.7	50	2.2
electrophysiology	108	3.7	94	4.1
fibers	17	0.6	16	0.7
active stress	0.1	<0.01	0.1	<0.01
electromechanics time step	18.00	100.0	16.80	100.0
electrophysiology	6.24	34.7	6.22	37.0
mechanics	9.21	51.2	9.23	55.0
coarse-to-fine interpolation	2.18	12.1	0.98	5.9
fine-to-coarse interpolation	0.37	2.1	0.36	2.2

Table 2: Breakdown of the computational cost for the whole-heart electromechanics simulation (Section 4.2), using either RL-RBF-G or RL-RBF interpolation. We report the wall time associated with the simulation setup and with solving one time step of electromechanics (comprising of 20 steps of electrophysiology, one step of force generation, one step of mechanics, and one each of fine-to-coarse and coarse-to-fine interpolation). For each, we also report the time spent in substeps, with the "relative" column reporting the percentage of the substep time relative to the parent step.

4.2.2. Role of the high-curvature threshold

To better understand the role of the high-curvature detection coefficient β, we perform a test using the same setting of previous section, except that we set $\beta = +\infty$ in (8). This is equivalent to disabling the detection of high-curvature regions altogether, i.e. redefining

$$d(\mathbf{x}, \mathbf{y}; R) = \begin{cases} +\infty & \text{if } g_h(\mathbf{x}, \mathbf{y}) > R \,, \\ \|\mathbf{x} - \mathbf{y}\| & \text{otherwise.} \end{cases} \tag{17}$$

Figure 9 highlights two critical regions for the interpolation procedure, close to the left atrial appendage and to the fossa ovalis i.e. a region where the walls of the two atria are in contact. The figure displays the result of interpolating, from the coarse to the fine mesh, a displacement field that maps the starting configuration onto the stress-free reference configuration, computed during preprocessing as discussed in [9, 26].

We observe that in both regions the RL-RBF interpolation yields significant oscillations. RL-RBF-G without high-curvature detection (i.e. with $\beta = +\infty$) eliminates the oscillations near the left atrial appendage (top row of Figure 9), but not those near the fossa ovalis (bottom row of Figure 9). Conversely, introducing high-curvature detection by setting $\beta = 0.5$ removes spurious oscillations in both regions.

An intuitive explanation for this behavior is that the condition $\beta h_{\max} + \|\mathbf{x} - \mathbf{y}\| < g_h(\mathbf{x}, \mathbf{y})$ in (8) empirically distinguishes between cases in which the exact geodesic distance g is better approximated by the Euclidean distance (if the condition is not met) or by discrete geodesic distance (if the condition is met). In the latter case, it is preferable to evaluate $d(\mathbf{x}, \mathbf{y}; R)$ through $g_h(\mathbf{x}, \mathbf{y})$ (instead of $\|\mathbf{x} - \mathbf{y}\|$), despite g_h being a potentially inaccurate approximation of g.

5. Conclusions

We proposed an interpolation method based on RL-RBF that uses a discrete approximation of the geodesic distance to take into account the geometry of the interpolation domain. The interpolation method, referred to as RL-RBF-G, allows to transfer variables between non-matching meshes, and prevents the onset of spurious oscillations associated with holes, cuts and otherwise complex geometrical features. Our numerical
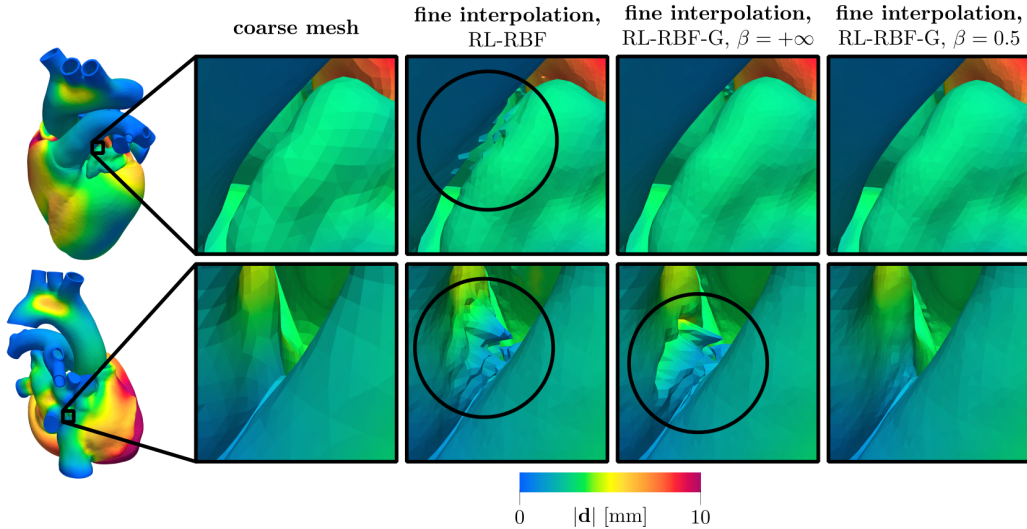
Figure 9: Whole-heart electromechanics (Section 4.2). Effect of RL-RBF-G interpolation and of the high-curvature detection in two regions close to the left atrial appendage and the fossa ovalis. Circles indicate spurious oscillations.

experiments demonstrate that the proposed method overcomes limitations of RL-RBF, while retaining good convergence properties.

Furthermore, we focused on the distributed-memory parallel implementation of RL-RBF-G interpolation, a non-trivial task due to the two meshes having independent parallel partitionings. The scalability tests carried out in this work indicate that the proposed implementation, available as open source within life$^{\texttt{x}}$ [1], has nearly ideal scalability properties, and is therefore well suited to large scale applications.

Finally, we demonstrate the effectiveness of RL-RBF-G interpolation in a realistic application to a whole-heart cardiac electromechanics model. This application is a particularly challenging setting for the proposed method, given the geometrical complexity of the domain. This test shows that the RL-RBF-G method is capable of transferring data between arbitrarily refined and even non-nested meshes, with an acceptable impact on the computational costs, thereby providing an effective tool in developing multiphysics computational models in geometrically complex domains.

Appendix A. Whole-heart electromechanics model

We briefly report the equations of the electromechanical model used in the numerical experiments of Section 4.2. The model used in this work was introduced in [26], and we refer to the original work for additional details.

The domain Ω, representing a four-chamber realistic human heart [72], is decomposed into several subsets: Ω_{LA} (left atrium), Ω_{RA} (right atrium), Ω_{V} (ventricles), Ω_{Ao} (ascending aorta), Ω_{PT} (pulmonary trunk) and Ω_{caps} (solid caps on vein inlets and valves). We define the myocardial subdomain as $\Omega_{\mathrm{myo}} = \Omega_{\mathrm{LA}} \cup \Omega_{\mathrm{RA}} \cup \Omega_{\mathrm{V}}$.

The domain boundary $\partial\Omega$ is also decomposed into several subsets: Γ_{rings} (the artificial boundaries where the veins and arteries are cut), $\Gamma_{\mathrm{epi}}^{\mathrm{PF}}$ (the epicardial portion of the boundary in contact with the pericardial fluid), $\Gamma_{\mathrm{epi}}^{\mathrm{EAT}}$ (the epicardial portion of the boundary in contact with the epicardial adipous tissue), $\Gamma_{\mathrm{epi}}^{\mathrm{art}}$ (the epithelial surface of the two arteries), the endocardial surfaces of the four chambers, denoted by Γ_{LA}, Γ_{LV}, Γ_{RA}, Γ_{RV}, and the endothelial surfaces of the two arteries, Γ_{Ao} and Γ_{PT}.

The domain Ω is obtained starting from a known configuration corresponding to medical imaging data. We consider the Zygote Heart Model [72], preprocessed with VMTK using the algorithms described in [27]. The model represents a configuration in which the heart muscle is subject to the pressure of the blood it contains and of the organs that surround it. To recover the stress-free configuration Ω, we use a reference configuration recovery algorithm described in [9, 52].

16

The model has the following unknowns:

$$v : \Omega_{\mathrm{myo}} \times (0, T) \to \mathbb{R} \qquad\qquad\qquad\qquad \text{transmembrane potential} ,$$

$$\mathbf{w}_{\mathrm{TTP}} : \Omega_{\mathrm{V}} \times (0, T) \to \mathbb{R}^{N_{\mathrm{TTP}}} \qquad\qquad \text{state variables of the TTP06 model} ,$$

$$\mathbf{w}_{\mathrm{CRN}} : (\Omega_{\mathrm{LA}} \cup \Omega_{\mathrm{RA}}) \times (0, T) \to \mathbb{R}^{N_{\mathrm{CRN}}} \qquad \text{state variables of the CRN model} ,$$

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

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

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

We reconstruct the orientation of cardiac muscle fibers through a Laplace-Dirichlet rule based method (LDRBM) [26, 46], defining an orthonormal triplet $\{\mathbf{f}_0, \mathbf{s}_0, \mathbf{n}_0\}$ at every point of Ω.

The evolution of the ionic variables is prescribed by the TTP06 and CRN ionic models.

In the myocardial subdomain Ω_{myo} we solve the monodomain equation with geometry-mediated mechano-electric feedback effects [18, 57]:

$$\begin{cases} J\chi C_{\mathrm{m}} \dfrac{\partial v}{\partial t} - \boldsymbol{\nabla} \cdot \left(J\mathbf{F}^{-1}\mathbf{D}_{\mathrm{m}}\mathbf{F}^{-T}\boldsymbol{\nabla}v \right) + J\chi I_{\mathrm{ion}}(v, \mathbf{w}_{\mathrm{TTP}}, \mathbf{w}_{\mathrm{CRN}}) = J\chi I_{\mathrm{app}} & \text{in } \Omega_{\mathrm{myo}} \times (0, T) , \\ J\mathbf{F}^{-1}\mathbf{D}_{\mathrm{m}}\mathbf{F}^{-T}\boldsymbol{\nabla}v \cdot \mathbf{n} = 0 & \text{on } \partial\Omega_{\mathrm{myo}} \times (0, T) , \\ v = v_0 & \text{in } \Omega_{\mathrm{myo}} \times \{0\} , \end{cases} \quad (A.1)$$

wherein χ is the membrane area-to-surface ratio, C_{m} is the membrane capacitance, and $\mathbf{D}_{\mathrm{m}}$ is the conductivity tensor, defined through

$$\mathbf{D}_{\mathrm{m}} = \sigma_{\mathrm{f}} \frac{\mathbf{F}\mathbf{f}_0 \otimes \mathbf{F}\mathbf{f}_0}{\|\mathbf{F}\mathbf{f}_0\|^2} + \sigma_{\mathrm{s}} \frac{\mathbf{F}\mathbf{s}_0 \otimes \mathbf{F}\mathbf{s}_0}{\|\mathbf{F}\mathbf{s}_0\|^2} + \sigma_{\mathrm{n}} \frac{\mathbf{F}\mathbf{n}_0 \otimes \mathbf{F}\mathbf{n}_0}{\|\mathbf{F}\mathbf{n}_0\|^2} , \quad (A.2)$$

with σ_{f}, σ_{s} and σ_{n} positive coefficient with different constant values in Ω_{V}, Ω_{LA} and Ω_{RA}.

The ionic current term $I_{\mathrm{ion}}(v, \mathbf{w}_{\mathrm{TTP}}, \mathbf{w}_{\mathrm{CRN}})$ is defined according to a different ionic model, depending on the subdomain:

$$I_{\mathrm{ion}}(v, \mathbf{w}_{\mathrm{TTP}}, \mathbf{w}_{\mathrm{CRN}}) = \begin{cases} I_{\mathrm{ion}}^{\mathrm{TTP}}(v, \mathbf{w}_{\mathrm{TTP}}) & \text{in } \Omega_{\mathrm{V}} , \\ I_{\mathrm{ion}}^{\mathrm{CRN}}(v, \mathbf{w}_{\mathrm{CRN}}) & \text{in } \Omega_{\mathrm{LA}} \cup \Omega_{\mathrm{RA}} . \end{cases} \quad (A.3)$$

The evolution of the ionic variables $\mathbf{w}_{\mathrm{TTP}}$ and $\mathbf{w}_{\mathrm{CRN}}$ is prescribed by a system of ODEs at every point in the corresponding subdomain:

$$\begin{cases} \dfrac{\partial \mathbf{w}_{\mathrm{TTP}}}{\partial t} = \mathbf{F}_{\mathrm{ion}}^{\mathrm{TTP}}(v, \mathbf{w}_{\mathrm{TTP}}) & \text{in } \Omega_{\mathrm{V}} \times (0, T) , \\ \mathbf{w}_{\mathrm{TTP}} = \mathbf{w}_{\mathrm{TTP},0} & \text{in } \Omega_{\mathrm{V}} \times \{0\} , \end{cases} \quad (A.4)$$

$$\begin{cases} \dfrac{\partial \mathbf{w}_{\mathrm{CRN}}}{\partial t} = \mathbf{F}_{\mathrm{ion}}^{\mathrm{CRN}}(v, \mathbf{w}_{\mathrm{CRN}}) & \text{in } (\Omega_{\mathrm{LA}} \cup \Omega_{\mathrm{RA}}) \times (0, T) , \\ \mathbf{w}_{\mathrm{CRN}} = \mathbf{w}_{\mathrm{CRN},0} & \text{in } (\Omega_{\mathrm{LA}} \cup \Omega_{\mathrm{RA}}) \times \{0\} . \end{cases} \quad (A.5)$$

The definitions of $I_{\mathrm{ion}}^{\mathrm{TTP}}$, $I_{\mathrm{ion}}^{\mathrm{CRN}}$, $\mathbf{F}_{\mathrm{ion}}^{\mathrm{TTP}}$ and $\mathbf{F}_{\mathrm{ion}}^{\mathrm{CRN}}$ are given in the original works [19, 62]. The initial conditions for v, $\mathbf{w}_{\mathrm{TTP}}$ and $\mathbf{w}_{\mathrm{CRN}}$ are determined as the limit cycle of a zero-dimensional simulation, as in [26, 47].

Both ionic models include a variable representing the intracellular calcium concentration, denoted by $w_{\mathrm{TTP}}^{\mathrm{Ca}}$ and $w_{\mathrm{CRN}}^{\mathrm{Ca}}$. For convenience, we define

$$[\mathrm{Ca}^{2+}]_{\mathrm{i}} = \begin{cases} w_{\mathrm{TTP}}^{\mathrm{Ca}} & \text{in } \Omega_{\mathrm{V}} , \\ w_{\mathrm{CRN}}^{\mathrm{Ca}} & \text{in } \Omega_{\mathrm{LA}} \cup \Omega_{\mathrm{RA}} . \end{cases} \quad (A.6)$$

The generation of contractile force is modeled through the RDQ20-MF model [51], which can be expressed as a system of ODEs:

$$\begin{cases} \dfrac{\partial \mathbf{s}}{\partial t} = \mathbf{F}_{\mathrm{act}}\left(\mathbf{s}, [\mathrm{Ca}^{2+}]_{\mathrm{i}}, \mathrm{SL}, \dfrac{\partial \mathrm{SL}}{\partial t} \right) & \text{in } \Omega_{\mathrm{myo}} \times (0, T) , \\ \mathbf{s} = \mathbf{s}_0 & \text{in } \Omega_{\mathrm{myo}} \times \{0\} , \end{cases} \quad (A.7)$$

where SL denotes the sarcomere length, defined as $\mathrm{SL} = \mathrm{SL}_0 \|\mathbf{F}\mathbf{f}_0\|$. The force generation model also defines the active stress tensor $\mathbf{P}_{\mathrm{act}} = \mathbf{P}_{\mathrm{act}}(\mathbf{s}, \mathbf{d})$. The initial condition for $\mathbf{s}$ is the steady state of a zero-dimensional simulation [26, 47].

The deformation of the muscle is governed by the elastodynamics equation, endowed with suitable boundary conditions [26]:

$$
\begin{cases}
\dfrac{\partial^2 \mathbf{d}}{\partial t^2} - \boldsymbol{\nabla} \cdot \mathbf{P}(\mathbf{s}, \mathbf{d}) = \mathbf{0} & \text{in } \Omega \times (0, T) , \\[2mm]
\mathbf{P}(\mathbf{s}, \mathbf{d})\mathbf{n} + (\mathbf{n} \otimes \mathbf{n}) \left(\mathbf{K}_{\mathrm{epi}}\mathbf{d} + \mathbf{C}_{\mathrm{epi}} \dfrac{\partial \mathbf{d}}{\partial t} \right) = \mathbf{0} & \text{on } \left(\Gamma_{\mathrm{epi}}^{PF} \cup \Gamma_{\mathrm{epi}}^{\mathrm{EAT}} \cup \Gamma_{\mathrm{epi}}^{\mathrm{art}} \right) \times (0, T) , \\[2mm]
\mathbf{P}(\mathbf{s}, \mathbf{d})\mathbf{n} = -p_{\mathrm{i}}(t)\mathbf{n} & \text{on } \Gamma_{\mathrm{i}} \times (0, T) \quad \text{for i} \in \{\mathrm{LA, LV, RA, RV, Ao, PT}\} , \\[2mm]
\mathbf{d} = \mathbf{0} & \text{on } \Gamma_{\mathrm{rings}} \times (0, T) , \\[2mm]
\mathbf{d} = \mathbf{d}_0 & \text{in } \Omega \times \{0\} , \\[2mm]
\dfrac{\partial \mathbf{d}}{\partial t} = \mathbf{0} & \text{in } \Omega \times \{0\} .
\end{cases}
\tag{A.8}
$$

The initial displacement is determined through the solution of a quasi-static problem imposing suitable initial endocardial pressures [26, 52].

Finally, the evolution of the endocardial pressures p_{i} is regulated by a lumped-parameter model of the circulatory system [11, 26, 35, 47, 52], expressed by a system of ODEs:

$$
\begin{cases}
\dfrac{\mathrm{d}\mathbf{c}}{\mathrm{d}t} = \mathbf{F}_{\mathrm{circ}}(\mathbf{c}, t) & \text{in } (0, T) , \\[2mm]
\mathbf{c}(0) = \mathbf{c}_0 .
\end{cases}
\tag{A.9}
$$

The circulation and mechanics models are further coupled by imposing the volume constraints

$$
V_{\mathrm{i}}^{\mathrm{3D}}(t) = V_{\mathrm{i}}^{\mathrm{0D}}(t) \quad \text{for i} \in \{\mathrm{LA, LV, RA, RV}\} ,
\tag{A.10}
$$

where $V_{\mathrm{i}}^{\mathrm{3D}}$ and $V_{\mathrm{i}}^{\mathrm{0D}}$ are the volumes of chamber i computed through the 3D mechanics model and the circulation model respectively [26].

Acknowledgements

Bibliography

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

[2] P. C. Africa, I. Fumagalli, M. Bucelli, A. Zingaro, M. Fedele, L. Dede', and A. Quarteroni. lifex-cfd: An open-source computational fluid dynamics solver for cardiovascular applications. *Computer Physics Communications*, 296:109039, 2024.

[3] P. C. Africa, R. Piersanti, M. Fedele, L. Dede', and A. Quarteroni. lifex-fiber: an open tool for myofibers generation in cardiac computational models. *BMC Bioinformatics*, 24:143, 2023.

[4] P. C. Africa, R. Piersanti, F. Regazzoni, M. Bucelli, M. Salvador, M. Fedele, S. Pagani, L. Dede', and A. Quarteroni. lifex-ep: a robust and efficient software for cardiac electrophysiology simulations. *BMC bioinformatics*, 24(1):389, 2023.

[5] D. Arndt, W. Bangerth, D. Davydov, T. Heister, L. Heltai, M. Kronbichler, M. Maier, J. Pelteret, B. Turcksin, and D. Wells. The `deal.II` finite element library: design, features, and insights. *Computers & Mathematics with Applications*, 81:407–422, 2021.

[6] D. Arndt, W. Bangerth, M. Feder, M. Fehling, R. Gassmöller, T. Heister, L. Heltai, M. Kronbichler, M. Maier, P. Munch, J.-P. Pelteret, S. Sticko, B. Turcksin, and D. Wells. The `deal.II` library, version 9.4. *Journal of Numerical Mathematics*, 30(3):231–246, 2022.

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

[8] C. M. Augustin, M. A. Gsell, E. Karabelas, E. Willemen, F. W. Prinzen, J. Lumens, E. J. Vigmond, and G. Plank. A computationally efficient physiologically comprehensive 3D–0D closed-loop model of the heart and circulation. *Computer methods in applied mechanics and engineering*, 386:114092, 2021.

[9] N. Barnafi, F. Regazzoni, and D. Riccobelli. Reconstructing relaxed configurations in elastic bodies: Mathematical formulations and numerical methods for cardiac modeling. *Computer Methods in Applied Mechanics and Engineering*, 423:116845, 2024.

[10] R. K. Beatson, J. B. Cherrie, and C. T. Mouat. Fast fitting of radial basis functions: Methods based on preconditioned GMRES iteration. *Advances in Computational Mathematics*, 11:253–270, 1999.

[11] P. J. Blanco and R. A. Feijóo. A 3D-1D-0D computational model for the entire cardiovascular system. *Mecánica Computacional*, 29(59):5887–5911, 2010.

[12] Official boost website. https://www.boost.org/ (last accessed: 6 November 2023).

[13] D. Brown, L. Ling, E. Kansa, and J. Levesley. On approximate cardinal preconditioning methods for solving PDEs with radial basis functions. *Engineering Analysis with Boundary Elements*, 29(4):343–353, 2005.

[14] M. Bucelli, F. Regazzoni, L. Dede', and A. Quarteroni. Preserving the positivity of the deformation gradient determinant in intergrid interpolation by combining RBFs and SVD: application to cardiac electromechanics. *Computer Methods in Applied Mechanics and Engineering*, page 116292, 2023.

[15] G. Chourdakis, K. Davis, B. Rodenberg, M. Schulte, F. Simonis, B. Uekermann, G. Abrams, H.-J. Bungartz, L. C. Yau, I. Desai, et al. preCICE v2: A sustainable and user-friendly coupling library. *Open Research Europe*, 2, 2022.

[16] P. Ciaccia, M. Patella, F. Rabitti, and P. Zezula. Indexing metric spaces with m-tree. In *SEBD*, volume 97, pages 67–86, 1997.

[17] T. C. Clevenger, T. Heister, G. Kanschat, and M. Kronbichler. A flexible, parallel, adaptive geometric multigrid method for FEM. *ACM Transactions on Mathematical Software (TOMS)*, 47(1):1–27, 2020.

[18] P. Colli Franzone, L. F. Pavarino, and S. Scacchi. *Mathematical Cardiac Electrophysiology*, volume 13. Springer, 2014.

[19] M. Courtemanche, R. J. Ramirez, and S. Nattel. Ionic mechanisms underlying human atrial action potential properties: insights from a mathematical model. *American Journal of Physiology-Heart and Circulatory Physiology*, 275(1):H301–H321, 1998.

[20] S. De Marchi and H. Wendland. On the convergence of the rescaled localized radial basis function method. *Applied Mathematics Letters*, 99:105996, 2020.

[21] Official `deal.ii` website. https://www.dealii.org/ (last accessed: 6 November 2023).

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

[23] S. Deparis, D. Forti, and A. Quarteroni. A rescaled localized radial basis function interpolation on non-cartesian and nonconforming grids. *SIAM Journal on Scientific Computing*, 36(6):A2745–A2762, 2014.

[24] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, 1959.

[25] P. Farah, A.-T. Vuong, W. Wall, and A. Popp. Volumetric coupling approaches for multiphysics simulations on non-matching meshes. *International Journal for Numerical Methods in Engineering*, 108(12):1550–1576, 2016.

[26] M. Fedele, R. Piersanti, F. Regazzoni, M. Salvador, P. C. Africa, M. Bucelli, A. Zingaro, L. Dede', and A. Quarteroni. A comprehensive and biophysically detailed computational model of the whole human heart electromechanics. *Computer Methods in Applied Mechanics and Engineering*, 410:115983, 2023.

[27] M. Fedele and A. Quarteroni. Polygonal surface processing and mesh generation tools for the numerical simulation of the cardiac function. *International Journal for Numerical Methods in Biomedical Engineering*, 37(4):e3435, 2021.

[28] C. Franke and R. Schaback. Solving partial differential equations by collocation using radial basis functions. *Applied Mathematics and Computation*, 93(1):73–82, 1998.

[29] T. Gerach and A. Loewe. Differential effects of mechano-electric feedback mechanisms on whole-heart activation, repolarization, and tension. *The Journal of Physiology*, 2024.

[30] T. Gerach, S. Schuler, J. Fröhlich, L. Lindner, E. Kovacheva, R. Moss, E. M. Wülfers, G. Seemann, C. Wieners, and A. Loewe. Electro-mechanical whole-heart digital twins: a fully coupled multi-physics approach. *Mathematics*, 9(11):1247, 2021.

[31] 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, and G. Plank. A framework for the generation of digital twins of cardiac electrophysiology from clinical 12-leads ECGs. *Medical Image Analysis*, 71:102080, 2021.

[32] V. Gurev, T. Lee, J. Constantino, H. Arevalo, and N. A. Trayanova. Models of cardiac electromechanics based on individual hearts imaging data: Image-based electromechanical models of the heart. *Biomechanics and Modeling in Mechanobiology*, 10(3):295–306, 2011.

[33] A. Guttman. R-trees: A dynamic index structure for spatial searching. In *Proceedings of the 1984 ACM SIGMOD international conference on Management of data*, pages 47–57, 1984.

[34] J. Heijman, H. Sutanto, H. J. Crijns, S. Nattel, and N. A. Trayanova. Computational models of atrial fibrillation: Achievements, challenges, and perspectives for improving clinical care. *Cardiovascular Research*, 117(7):1682–1699, 2021.

[35] M. Hirschvogel, M. Bassilious, L. Jagschies, S. M. Wildhirt, and 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. *International Journal for Numerical Methods in Biomedical Engineering*, 33(8):e2842, 2017.

[36] E. Karabelas, M. A. Gsell, G. Haase, G. Plank, and C. M. Augustin. An accurate, robust, and efficient finite element framework with applications to anisotropic, nearly and fully incompressible elasticity. *Computer Methods in Applied Mechanics and Engineering*, 394:114887, 2022.

[37] M. Kronbichler and K. Ljungkvist. Multigrid for matrix-free high-order finite element computations on graphics processors. *ACM Transactions on Parallel Computing (TOPC)*, 6(1):1–32, 2019.

[38] M. Lanthier, A. Maheshwari, and J. R. Sack. Approximating shortest paths on weighted polyhedral surfaces. *Algorithmica*, 30:527–562, 2001.

[39] M. Lanthier, A. Maheshwari, and J.-R. Sack. Approximating weighted shortest paths on polyhedral surfaces. In *Proceedings of the thirteenth annual symposium on Computational geometry*, pages 274–283, 1997.

[40] 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, and B. Rodriguez. Sensitivity analysis of a strongly-coupled human-based electromechanical cardiac model: Effect of mechanical parameters on physiologically relevant biomarkers. *Computer Methods in Applied Mechanics and Engineering*, 361:112762, 2020.

[41] Official life$^\text{x}$ website. https://lifex.gitlab.io/ (last accessed: 6 November 2023).

[42] Y. Manolopoulos, A. N. Papadopoulos, and Y. Theodoridis. *R-Trees: Theory and Applications*. Springer Science & Business Media, 2006.

[43] B. S. Morse, T. S. Yoo, P. Rheingans, D. T. Chen, and K. R. Subramanian. Interpolating implicit surfaces from scattered surface data using compactly supported radial basis functions. In *ACM SIGGRAPH 2005 Courses*, pages 78–es. 2005.

[44] S. Pagani, L. Dede', A. Frontera, M. Salvador, L. Limite, A. Manzoni, F. Lipartiti, G. Tsitsinakis, A. Hadjis, P. Della Bella, and A. Quarteroni. A computational study of the electrophysiological substrate in patients suffering from atrial fibrillation. *Frontiers in Physiology*, 12:673612, 2021.

[45] M. Peirlinck, F. S. Costabal, J. Yao, J. Guccione, S. Tripathy, Y. Wang, D. Ozturk, P. Segars, T. Morrison, S. Levine, and E. Kuhl. Precision medicine in human heart modeling. *Biomechanics and Modeling in Mechanobiology*, 20(3):803–831, 2021.

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

[47] R. Piersanti, F. Regazzoni, M. Salvador, A. F. Corno, C. Vergara, and A. Quarteroni. 3D–0D closed-loop model for the simulation of cardiac biventricular electromechanics. *Computer Methods in Applied Mechanics and Engineering*, 391:114607, 2022.

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

[49] A. Quarteroni, L. Dede', and F. Regazzoni. Modeling the cardiac electromechanical function: A mathematical journey. *Bulletin of the American Mathematical Society*, 59(3):371–403, 2022.

[50] A. Quarteroni, L. Dede', F. Regazzoni, and C. Vergara. A mathematical model of the human heart suitable to address clinical problems. *Japan Journal of Industrial and Applied Mathematics*, pages 1–21, 2023.

[51] F. Regazzoni, L. Dede', and A. Quarteroni. Biophysically detailed mathematical models of multiscale cardiac active mechanics. *PLoS Computational Biology*, 16(10):e1008294, 2020.

[52] F. Regazzoni, M. Salvador, P. C. Africa, M. Fedele, L. Dede', and A. Quarteroni. A cardiac electromechanical model coupled with a lumped-parameter model for closed-loop blood circulation. *Journal of Computational Physics*, 457:111083, 2022.

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

[54] M. Sala and M. A. Heroux. Robust algebraic preconditioners using IFPACK 3.0. Technical report, Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), 2005.

[55] M. Salvador, L. Dede', and A. Quarteroni. An intergrid transfer operator using radial basis functions with application to cardiac electromechanics. *Computational Mechanics*, 66:491–511, 2020.

[56] M. Salvador, M. Fedele, P. C. Africa, E. Sung, L. Dede', A. Prakosa, J. Chrispin, N. A. Trayanova, and A. Quarteroni. Electromechanical modeling of human ventricles with ischemic cardiomyopathy: numerical simulations in sinus rhythm and under arrhythmia. *Computers in Biology and Medicine*, 136:104674, 2021.

[57] M. Salvador, F. Regazzoni, S. Pagani, N. A. Trayanova, and A. Quarteroni. The role of mechano-electric feedbacks and hemodynamic coupling in scar-related ventricular tachycardia. *Computers in Biology and Medicine*, 142:105203, 2022.

[58] S. Stella, F. Regazzoni, C. Vergara, L. Dede', and A. Quarteroni. A fast cardiac electromechanics model coupling the eikonal and the nonlinear mechanics equations. *Mathematical Models and Methods in Applied Sciences*, 32(08):1531–1556, 2022.

[59] 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, and 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. *Journal of Biomechanics*, 101:109645, 2020.

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

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

[62] K. H. Ten Tusscher and A. V. Panfilov. Alternans and spiral breakup in a human ventricular tissue model. *American Journal of Physiology-Heart and Circulatory Physiology*, 291(3):H1088–H1100, 2006.

[63] F. Viola, G. Del Corso, R. De Paulis, and R. Verzicco. Gpu accelerated digital twins of the human heart open new routes for cardiovascular research. *Scientific Reports*, 13(1):8230, 2023.

[64] Y. Voet, G. Anciaux, S. Deparis, and P. Gervasio. The INTERNODES method for applications in contact mechanics and dedicated preconditioning techniques. *Computers & Mathematics with Applications*, 127:48–64, 2022.

[65] T. Washio, J.-i. Okada, A. Takahashi, K. Yoneda, Y. Kadooka, S. Sugiura, and T. Hisada. Multiscale heart simulation with cooperative stochastic cross-bridge dynamics and cellular structures. *Multiscale Modeling & Simulation*, 11(4):965–999, 2013.

[66] H. Wendland. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in computational Mathematics*, 4(1):389–396, 1995.

[67] H. Wendland. Meshless Galerkin methods using radial basis functions. *Mathematics of computation*, 68(228):1521–1531, 1999.

[68] J. W. J. Williams. Algorithm 232 - heapsort. *Communications of the ACM*, 7(6):347–348, 1964.

[69] E. Zappon, A. Manzoni, P. Gervasio, and A. Quarteroni. A reduced order model for domain decompositions with non-conforming interfaces. *arXiv preprint arXiv:2206.09618*, 2022.

[70] E. Zappon, A. Manzoni, and A. Quarteroni. Efficient and certified solution of parametrized one-way coupled problems through DEIM-based data projection across non-conforming interfaces. *Advances in Computational Mathematics*, 49(2):21, 2023.

[71] A. Zingaro, M. Bucelli, R. Piersanti, F. Regazzoni, L. Dede', and A. Quarteroni. An electromechanics-driven fluid dynamics model for the simulation of the whole human heart. *Journal of Computational Physics (in press)*, 2024.

[72] Zygote Media Group Inc. Zygote solid 3D heart generation II developement report. Technical report. 2014.

MOX Technical Reports, last issues

Dipartimento di Matematica

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

25/2024 Enrico Ballini e Luca Formaggia e Alessio Fumagalli e Anna Scotti e Paolo Zunino
Application of Deep Learning Reduced-Order Modeling for Single-Phase Flow in Faulted Porous Media

22/2024 Gatti, F.; de Falco, C.; Fois, M.; Formaggia, L.
A scalable well-balanced numerical scheme for a depth-integrated lava flow model

21/2024 Caldana, M.; Antonietti P. F.; Dede' L.
Discovering Artificial Viscosity Models for Discontinuous Galerkin Approximation of Conservation Laws using Physics-Informed Machine Learning

19/2024 Torzoni, M.; Manzoni, A.; Mariani, S.
A multi-fidelity surrogate model for structural health monitoring exploiting model order reduction and artificial neural networks

20/2024 Torzoni, M.; Manzoni, A.; Mariani, S.
Structural health monitoring of civil structures: A diagnostic framework powered by deep metric learning

17/2024 Fois, M.; de Falco, C.; Formaggia, L.
A semi-conservative depth-averaged Material Point Method for fast flow-like landslides and mudflows

16/2024 Domanin D. A.; Pegoraro M.; Trimarchi S.; Domanin M.; Secchi P.
Persistence diagrams for exploring the shape variability of abdominal aortic aneurysms

15/2024 Vaccaro, F.; Mauri, A.G.; Perotto, S.; Brivio, S.; Spiga, S.
Modeling and simulation of electrochemical and surface diffusion effects in filamentary cation-based resistive memory devices

14/2024 Zappon, E.; Salvador, M.; Piersanti, R.; Regazzoni, F.; Dede', L.; Quarteroni, A.
An integrated heart-torso electromechanical model for the simulation of electrophysiological outputs accounting for myocardial deformation

12/2024 Zingaro, A.; Ahmad, Z.; Kholmovski, E.; Sakata, K.; Dede', L.; Morris, A.K.; Quarteroni, A.; Trayanova, N.A.
A comprehensive stroke risk assessment by combining atrial computational fluid dynamics simulations and functional patient data