



MOX-Report No. 46/2020

**Multipatch Isogeometric Analysis for Electrophysiology:  
Simulation in a Human Heart**

Bucelli, M.; Salvador, M.; Dede', L.; Quarteroni, A.

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

[mox-dmat@polimi.it](mailto:mox-dmat@polimi.it)

<http://mox.polimi.it>

# Multipatch Isogeometric Analysis for Electrophysiology: Simulation in a Human Heart

Michele Bucelli<sup>1,\*</sup>, Matteo Salvador<sup>1</sup>, Luca Dedè<sup>1</sup>, Alfio Quarteroni<sup>1,2</sup>

<sup>1</sup> MOX, Department of Mathematics, Politecnico di Milano, P.zza Leonardo da Vinci 32, 20133 Milan, Italy

<sup>2</sup> Mathematics Institute, EPFL, Av. Piccard, CH-1015 Lausanne, Switzerland (Professor Emeritus)

July 6, 2020

## Abstract

In the framework of cardiac electrophysiology for the human heart, we apply multipatch NURBS-based Isogeometric Analysis for the space discretization of the Monodomain model. Isogeometric Analysis (IGA) is a technique for the solution of Partial Differential Equations (PDEs) that facilitates encapsulating the exact representation of the computational geometry by using basis functions with high-order continuity. IGA features very small numerical dissipation and dispersion if compared to other methods for the solution of PDEs. The use of multiple patches allows to overcome the conventional limitations of single patch IGA, thanks to the gained flexibility in the design of the computational domain, especially when its representation is quite involved as in bioengineering applications. We propose two algorithms for the preprocessing of CAD models of complex surface and volumetric NURBS geometries with cavities, such as atria and ventricles: our purpose is to obtain geometrically and parametrically conforming NURBS multipatch models starting from CAD models. We employ those algorithms for the construction of an IGA realistic representation of a human heart. We apply IGA for the discretization of the Monodomain equation that describes the evolution in space and time of the cardiac action potential. This PDE is coupled with suitable microscopic models to define the behavior at cellular scale: Courtemanche-Ramirez-Nattel model for the atrial simulation, and Luo-Rudy model for the ventricular one. Numerical simulations on realistic human atria and ventricles geometries are carried out, obtaining accurate and smooth excitation fronts by combining IGA with the multipatch approach for the geometrical representation of the computational domains, either surfaces for the atria or solids for the ventricles.

## 1 Introduction

The heart is a muscular organ whose contraction pumps blood into the circulatory system, delivering oxygen and nutrients and allowing all the cells to perform their physiological activities [40, 48, 49]. The mechanical contraction of the heart is triggered by the electrical excitation of the cardiac tissue, starting from the right atrium at the sinoatrial node (SAN) and propagating to the left atrium and to the ventricles [71, 74]. Since cardiovascular diseases are a very common cause of death worldwide, understanding heart function in both physiological and pathological conditions

---

\* Corresponding author

*Email addresses:* `michele.bucelli@polimi.it` (Michele Bucelli), `matteo1.salvador@polimi.it` (Matteo Salvador), `luca.dede@polimi.it` (Luca Dedè), `alfio.quarteroni@polimi.it` (Alfio Quarteroni)

is crucial. In computational medicine, mathematical modeling and numerical simulations are used to provide a quantitative representation of the heart function, thus giving a very valuable contribution in this field [22, 66, 76].

Indeed, the electrical activation of the heart can be accurately described with mathematical models expressed in terms of partial differential equations (PDEs). A very common model for cardiac electrophysiology is the Monodomain equation [14, 22, 57, 65], which describes the evolution in time and space of the transmembrane potential in a continuous domain. This PDE needs to be coupled with an ionic model describing the transmembrane ionic currents that determine the electrical activation of cardiac cells. Among the different ionic models proposed in literature [5, 15, 42, 68], we consider the Luo-Rudy (LR) model [53] for the ventricles and the Courtemanche-Ramirez-Nattel (CRN) [29] model for the atria.

The complexity of electrophysiological models poses a significant challenge for their mathematical analysis and numerical approximation [21, 23, 24, 66]. A fine space resolution is indeed required to capture the sharp propagating excitation fronts and suitable methods are needed to control numerical dissipation and dispersion. The use of the Finite Element Method (FEM) in the space discretization of either Bidomain model [13, 75] or Monodomain equation requires highly refined meshes to capture the propagating front without introducing spurious oscillations. This entails a very large number of degrees of freedom (DOFs) and a consequent large computational cost [51].

In this paper, numerical simulations of the cardiac electrophysiology of the human heart are performed by means of multipatch Isogeometric Analysis (IGA) in the framework of the Galerkin method [10, 17, 26, 27, 33, 45]. IGA is a methodology for the space discretization of PDEs that in many circumstances yields an exact representation of the computational geometry. Thanks to the high-order continuity of NURBS basis functions [60], it allows to solve PDEs accurately and efficiently, without necessarily requiring fine meshes while featuring limited numerical dispersion and dissipation [27, 32, 37, 44]. These aspects make IGA better suited than FEM for wave propagation problems and PDEs featuring solutions with coherent structures, as is the case of cardiac electrophysiology models.

IGA has been applied so far to cardiac electrophysiology in [20, 57, 59] using the single patch approach, i.e. describing the computational domain by means of a single NURBS-parametrized geometry, i.e. the so called patch. However, the accuracy of the solution obtained by IGA is affected by a parametrization of the domain featuring significant distortions [12, 27, 30]. If we consider geometries with multiple cavities or complex topologies, such as those in a realistic representation of the heart, it is crucial to represent the domain by means of multiple patches. This approach is known as multipatch technique [25, 27] and it allows reproducing a complex geometry by means of multiple patches with mesh elements with much better quality than in the single patch case, besides allowing for more flexibility in the definition of the computational domain.

Although CAD software can natively handle NURBS-parametrized geometries, CAD models may not be directly applicable to IGA. Indeed, IGA requires adjacent patches to be conforming, both geometrically and parametrically. For this reason, we propose two algorithms to convert a multipatch surface representation of the computational domain into an IGA one. The first algorithm addresses patch non-conformity issues in a surface geometry, while the second one addresses the issue of constructing a volumetric parametrization starting from a description of the domain boundary. These algorithms are applied to the Zygote Solid 3D heart model [3], resulting in an IGA reconstruction of a realistic human heart. In this way we obtain a domain description ready for the application of multipatch IGA to the space discretization of differential problems, including but not limited to the electrophysiology model considered in this paper.

The paper is structured as follows. In Section 2, we provide a brief description of the electro-

physiology of the human heart. In Section 3 we present NURBS-based IGA and the multipatch. In Section 4 we describe a preprocessing pipeline to convert complex CAD models with cavities into suitable multipatch computational models, and apply it to a realistic representation of both atria and ventricles. In Section 5 we introduce the Monodomain equation along with the mathematical method used for its numerical discretization. In Section 6 we discuss the numerical results. Finally, in Section 7 we draw some conclusions.

## 2 Cardiac electrophysiology

The human heart is a hollow muscular organ that pumps blood through the circulatory system, providing the body with oxygen and nutrients and assists with the removal of the metabolic wastes produced by cellular reactions [40, 48, 49]. The main driver of blood flow through the different chambers of the heart and through the circulatory system is the active contraction of the heart muscle. Such contraction is initiated by an electrical excitation of cardiac cells [22, 73], which are able to activate themselves independently of nervous stimuli, unlike skeletal muscle cells.

The electrical activation is due to an action potential (AP): the difference of potential across the cellular membrane (referred to as transmembrane potential) of excitable cardiac cells is quickly (in about 2 ms) going from its resting value (about  $-84$  mV) to slightly positive values by a depolarizing membrane current, followed by a repolarizing current that brings the potential back to the resting value. This is made possible by several ion channels opening and closing and resulting in ionic currents through the cellular membrane. Such currents cause variations in the concentration of calcium ions in the intracellular space, that in turn initiate the contraction of the myocytes [49], so that the electrical activation drives the mechanical one.

Action potential propagates from one cell to its neighbors through gap junctions, intercellular low-resistance ionic channels. This allows the stimulus to travel from one cell to another one through the whole tissue. Cardiomyocytes are arranged in sheets of fibers. The gap junctions are located for the most part, although not exclusively, in the longitudinal direction of the fibers, resulting in a preferential propagation of the electric signal along the fibers [22].

The electric signal originates at the SAN, a portion of tissue located at the junction between the superior vena cava and the right atrium. The SAN is made of self-exciting cells and serves as the natural pacemaker of the heart. The action potential propagates from the SAN through the right atrium and to the left atrium through a few preferential conduction sites (Bachmann's bundle first, then the anterior and posterior septum and the coronary sinus musculature) [71]. The activation front reaches the atrioventricular node (AV), located at the junction between the interventricular septum and the interatrial septum. The cells composing this node have a slow conduction velocity, that causes a delay between the atrial and ventricular activation. The AV node conducts the electric signal through the atrioventricular septum and it activates the fibers of the bundle of His and of the Purkinje network, composed of cells specialized for the conduction. These fibers transmit then the action potential to the ventricular walls.

## 3 Multipatch IGA

IGA [10, 16, 17, 18, 26, 27, 33, 56] is a framework for the spatial approximation of PDEs based on the isogeometric paradigm: that is using as basis functions the same functions that are used for the parametrization of the computational domain. The choice of such functions is driven by the design of the geometry.

Let  $\Omega \subset \mathbb{R}^d$  be an open, bounded computational domain, defined as the union of  $N_p$  disjoint subdomains  $\Omega_k, k = 1, \dots, N_p$ , referred to as patches. More formally:

$$\overline{\Omega} = \bigcup_{k=1}^{N_p} \overline{\Omega}_k \quad \text{with} \quad \Omega_i \cap \Omega_j = \emptyset \quad i \neq j.$$

Each patch  $\Omega_k$  is the image of a parametric domain  $\widehat{\Omega} \subset \mathbb{R}^{\widehat{d}}$ , with  $\widehat{d} \leq d$  (typically  $\widehat{\Omega} = [0, 1]^{\widehat{d}}$ ) through a mapping in the form

$$\varphi_k(\boldsymbol{\xi}) = \sum_{i=1}^{n_k} \widehat{R}_i^k(\boldsymbol{\xi}) \mathbf{B}_i^k \quad i = 1, \dots, n_k, \quad k = 1, \dots, N_p, \quad (1)$$

where  $\widehat{R}_i^k(\boldsymbol{\xi})$  are suitable basis functions defined in the parametric domain and  $\mathbf{B}_i^k \in \mathbb{R}^d$  are points in the physical space known as control points.

In NURBS-based IGA, functions  $\widehat{R}_i^k$  are  $\widehat{d}$ -dimensional NURBS basis functions. They are tensor-product piecewise rational functions, whose definition involves the partition of each dimension of the parametric domain in sub-intervals, referred to as knot spans. The extremes of knot spans are known as knots, and the images of knot spans through the mapping (1) are referred to as elements. For more details on the definition of NURBS functions and their use for the parametrization of domains, we refer to [28, 60].

The choice of NURBS functions allows an easy integration with CAD software used in the design of the computational domain, since it usually handles NURBS geometries natively. Moreover, NURBS functions allow several refinement procedures that enrich the functional space used for the parametrization while preserving the exact geometry. Finally, NURBS feature useful approximation properties that make them suitable for the discretization of PDE problems [12, 18, 27, 30], in particular with respect to numerical dissipation [28] and dispersion [19, 32]. Specifically, controlling numerical dispersion is an important feature of NURBS-based IGA when approximating electrophysiological problems [57, 59].

For the space approximation of PDEs, the Galerkin method is employed using in each patch  $\Omega_k$  as basis functions the push-forward through  $\varphi_k$  of the basis used in the domain parametrization:

$$R_i^k(\mathbf{x}) = \widehat{R}_i^k(\varphi_k(\mathbf{x})).$$

Following [18, 50], we assume that adjacent patches are conforming. This means that, if  $\Gamma_{kl} = \partial\Omega_k \cap \partial\Omega_l \neq \emptyset$  is the interface of patches  $\Omega_k$  and  $\Omega_l$  for some  $k \neq l$ , then:

- $\Gamma_{kl}$  is either a single point or the image of an entire edge of the parametric domain, or else the image of an entire face (for volumetric patches, i.e. for  $\widehat{d} = 3$ ) of the parametric domain for both patches;
- if a basis function  $R_i^k$  in  $\Omega_k$  has support on  $\Gamma_{kl}$ , then there exists a basis function  $R_j^l$  in  $\Omega_l$  that coincides with  $R_i^k$  at the interface:  $R_i^k|_{\Gamma_{kl}} = R_j^l|_{\Gamma_{kl}}$ .

These assumptions imply that adjacent patches are geometrically and parametrically coincident at their interface. Define the function spaces

$$\begin{aligned} V_h^k &= \text{span}\{R_i^k, i = 1, \dots, n_k\} \quad k = 1, \dots, N_p, \\ V_h &= \{f \in C^0(\overline{\Omega}) : f|_{\Omega_k} \in V_h^k \quad \forall k = 1, \dots, N_p\}. \end{aligned}$$

Under the previous assumptions, it is possible to define a basis over the space  $V_h$ . For each basis function  $R_i^k$  define an index  $G(i, k) \in \mathbb{N}$  such that

$$\begin{aligned} G(i, k) &= G(j, l) && \text{if } \text{supp}(R_i^k \cap \Gamma_{kl}) \neq \emptyset, \text{supp}(R_j^l \cap \Gamma_{kl}) \neq \emptyset, R_i^k|_{\Gamma_{kl}} = R_j^l|_{\Gamma_{kl}}, \\ G(i, k) &\neq G(j, l) && \text{otherwise,} \end{aligned}$$

and let

$$\mathcal{I} = \{G(i, k), i = 1, \dots, n_k, k = 1, \dots, N_p\},$$

be the set of all indices  $G(i, k)$ . In this way, we assign the same index to two corresponding basis functions at the interface between two patches. Then, a basis for  $V_h$  is given by the functions

$$R_l(\mathbf{x}) = \begin{cases} R_i^k(\mathbf{x}) & \text{if } \mathbf{x} \in \Omega_k \text{ and } l = G(i, k) \\ 0 & \text{otherwise} \end{cases} \quad l \in \mathcal{I}.$$

The space  $V_h$  is used as trial and test space for the approximation of the differential models for electrophysiology that we will introduce in Section 5.2. The method achieves optimal order of convergence with respect to the mesh size parameter  $h$ , maximum diameter of all elements of all patches [12, 18]. More details on the approximation properties of the spaces  $V_h^k$  and  $V_h$  can be found in [12, 18, 30].

Following this approach, solutions on adjacent patches are coupled by prescribing the continuity of the solution across the interface, which can be achieved provided the adjacent patches match completely both geometrically and parametrically [27]. This requires adjacent patches to have the same knot vectors along their interfaces and to have coincident control points. The strong imposition of continuity across patch interfaces can be obtained also if the two patches are geometrically coincident at the interface, and the interface parametrization of one patch is obtained via h-refinement from that of the other [27]. A discontinuous Galerkin technique for the coupling of non-conforming adjacent patches is discussed in [52], while the possibility of imposing higher order continuity across patch interfaces is investigated in [25, 45, 46]. Moreover, in [6, 7, 47, 55] techniques for the management of trimmed geometries and geometries obtained via boolean operations are presented.

The accuracy of the solution is hindered by the distortion of the geometrical map  $\varphi$  [12], quantified in terms of the determinant of its gradient  $J = \det(\nabla\varphi)$ . Because of the tensor product structure of the parametric space, representing multiple connected domains with a single patch causes large deformations in the geometrical map. Indeed, each patch can be seen as a continuous deformation of the parametric space (i.e. a cube). Therefore, the single patch representation of complex domains, for example domains with multiple cavities such as those involved in cardiac modeling, requires significant mesh deformations. In such cases, multiple patches allow to construct regular parametrizations, resulting in a gain in accuracy [27].

A detailed description of the methods and properties of NURBS-based IGA can be found in [26, 27].

## 4 Generating multipatch NURBS geometries for heart modeling

In order to solve PDEs in NURBS multipatch geometries in the standard Galerkin framework, the patches need to be conforming (or, more in general, adjacent patches need to differ in their parametrization solely by the insertion of one or more knots). Moreover, if the computational domain is a volume, the NURBS parametrization should cover the whole volume, and not only

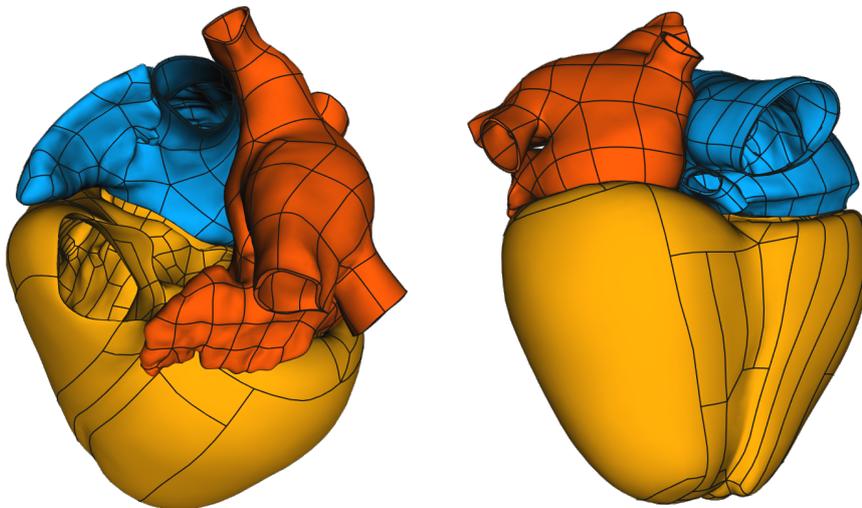


Figure 1: Zygote heart model [3]. Black lines are the boundaries of the patches in the original geometry. Notice in particular that patches are not conforming, as evident from the figure on the right.

its boundary surface. For these reasons, geometries coming from CAD modeling may not be ready to be used in numerical simulations directly.

Therefore, we propose two algorithms for the preprocessing of NURBS geometries. The first algorithm addresses non-conformity issues in a NURBS multipatch surface geometry. The second algorithm deals with the problem of constructing a NURBS volume starting from a surface representation of its boundary. In both cases, we will assume the original multipatch geometries to be continuous at the geometric level, i.e. to be composed by a set of  $G^0$  surface patches. The two algorithms provide a continuous parametrization for the same geometry.

The manipulation of NURBS geometries is performed by means of the MATLAB/Octave NURBS package [1, 31].

Both for atria and ventricles, the source geometry is a 3D model of the heart of a healthy, 21 year-old, 50th percentile U.S. male, created by the Zygote Media Group [3], reconstructed from magnetic resonance images. The original geometry is shown in Figure 1.

#### 4.1 Preprocessing of a realistic human atrial geometry

We consider a realistic human atrial geometry modeled as a NURBS multipatch surface. All patches are geometrically watertight at their interfaces. However some pairs of adjacent patches do not have the same parametrization at the interface, which is required to impose strong continuity of the solution. We face these issues in the mesh generation, producing a geometry composed of conforming patches. We distinguish two types of non-conformities:

- patch-level non-conformities: two adjacent patches only share a part of one side (Figure 2a);
- knot-level non-conformities: two adjacent patches share an entire side, but the interface is differently parametrized on the two patches (due to different knot vectors in the corresponding parametric directions; Figure 4a).

The following sections describe the algorithms to address non-conformities of the two types.

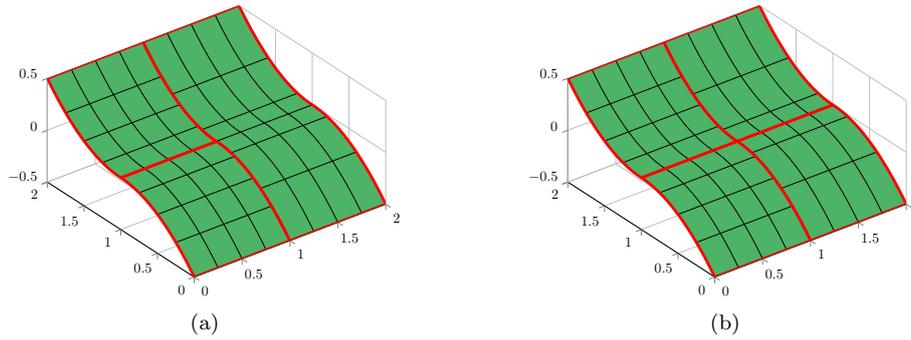


Figure 2: Solution of a patch-level non-conformity; patch boundaries are shown in red. (a): initially, one side of the rightmost patch is shared with both the other patches; (b): a  $C^0$  knot is inserted in the rightmost patch in correspondence of the boundary between the other patches, and then such patch is split into two patches at that position.

Such algorithms manipulate the patches' knot vectors and control points in order to eliminate all non-conformities and convert the geometry into a conforming multipatch NURBS surface.

#### 4.1.1 Detection of patch-level non-conformities

For a NURBS surface patch, we refer to the four points corresponding to the parametric coordinates  $(0, 0)$ ,  $(0, 1)$ ,  $(1, 0)$  and  $(1, 1)$  as vertices, and to the four NURBS curves corresponding to the parametric equations  $\xi = 0$ ,  $\xi = 1$ ,  $\eta = 0$  and  $\eta = 1$  as sides. The non-conformity between two patches can be checked by counting the vertices that are shared by two patches, according to the following algorithm.

Algorithm 1: Detection of adjacent patches that are not non-conforming at the patch level

```

1: procedure DETECTPATCHLEVELNONCONF(p1, p2)
   Input p1, p2 - the two patches
   Output result - one of CONFORMING, NON_CONFORMING, NOT_ADJACENT

2:   extract vertices of both patches
3:   if the patches have two common vertices then
4:     result = CONFORMING
5:   else if the patches have one common vertex then
6:     P = common vertex
7:     if p2 has a vertex on one side of p1 then
8:       result = NON_CONFORMING
9:     else if p1 has a vertex on one side of p2 then
10:      result = NON_CONFORMING
11:    else
12:      result = NOT_ADJACENT
13:    end if
14:  else
15:    result = NOT_ADJACENT
16:  end if
17: end procedure

```

To check if one vertex lies on one side of a patch (Algorithm 1, lines 7 and 9), the NURBS curve that parametrizes that side must be inverted. This requires the solution of a non-linear equation with Newton's method [60] and can be computationally demanding.

Algorithm 2 takes as input two sets of patches  $p1$  and  $p2$  and returns all pairs of patches, one from  $p1$  and one from  $p2$ , that are non-conforming at the patch level.

Algorithm 2: List all non-conforming pairs of patches in a set

```

1: procedure LISTPATCHLEVELNONCONF(p1, p2)
   Input p1, p2 - two sets of patches
   Output pairs - a set of pairs [i,j] such that p1[i] and p2[j]
   are patch-level non-conforming
2:   n1 = length(p1)
3:   n2 = length(p2)
4:   pairs = []
5:   for i = 1, ..., n1 do
6:     for j = 1, ..., n2 do
7:       if detectPatchLevelNonConf(p1[i],p2[j]) == NON_CONFORMING then
8:         insert [i,j] into pairs
9:       end if
10:    end for
11:  end for
12: end procedure

```

#### 4.1.2 Solution of patch-level non-conformities

Once a patch-level non-conformity is detected, it can be removed by inserting a  $C^0$  knot in correspondence of the hanging node, thus splitting one of the two patches into two smaller patches. This operation, shown in Figure 2, does not introduce any geometrical change in the geometry.

Assume to have defined a procedure `splitPatch(patch,direction)` that returns the patches obtained by splitting `patch` in correspondence of the  $C^0$  knots along the parametric direction `direction`.

For simplicity, the following algorithm deals only with the case of the two patches sharing a vertex and `patch1` having a vertex on the side of `patch2`. The symmetric case is handled in the same way. More general cases in which the patches are adjacent without sharing vertices can be handled similarly and have been omitted for brevity. In the pseudocode that follows, we denote with `p.knots[i]` the knot vector of patch `p` along the `i`-th parametric direction (with `i` either 1 or 2).

Algorithm 3: Solve a patch-level non-conformity between two patches

```

1: procedure SOLVEPATCHLEVELNONCONF(patch1, patch2)
   Input patch1, patch2 - the two patches
   Output patches - set of resulting patches
   Require patch1 and patch2 have a common vertex
   Require patch1 has a vertex on one side of patch2 (the symmetric
   case is the same)
2:   p = polynomial degree of patch2
3:   direction2 = parametric direction of the interface in patch2

```

```

4:   A = common vertex to patch1 and patch2
5:   B = hanging vertex
6:
7:   xi = coordinate of B on the side of patch2
8:   insert xi into patch2.knots[direction2] with multiplicity p+1
9:
10:  [patch2a, patch2b] = splitPatch(patch2,direction2)
11:  patches = [patch1, patch2a, patch2b]
12: end procedure

```

To solve all patch-level non-conformities, we detect all pairs of non-conforming adjacent patches. Then, we solve non-conformities by the procedure described above and update the list of non-conforming pairs since the modification might have introduced further issues, repeating until all non-conformities have been solved. At each step we check only the modified patches for new non-conformities (Algorithm 4, lines 8–14).

Algorithm 4: Detect and solve patch-level non-conformities

```

1: procedure PATCHLEVELNONCONFORMITIES(P)
   Input P - the set of patches
   Output modifies P solving patch-level non-conformities

2:   NC = listPatchLevelNonConf(P, P)
3:   while length(NC) > 0 do
4:     Pnew = solvePatchLevelNonConf(P[NC[1][1]],P[NC[1][2]])
5:     remove patches P[NC[1][1]] and P[NC[1][2]] from P
6:     insert patches Pnew into P
7:
8:     remove all pairs containing either NC[1][1] or NC[1][2]
       from NC
9:
10:    NCnew1 = listPatchLevelNonConf(P, Pnew)
11:    NCnew2 = listPatchLevelNonConf(Pnew,Pnew)
12:
13:    append NCnew1 to NC
14:    append NCnew2 to NC
15:  end while
16: end procedure

```

### 4.1.3 Solution of knot-level non-conformities

A knot-level non-conformity between two adjacent patches can be detected by comparing the knot vectors at the interface of the two patches.

Knot-level non-conformities are removed simultaneously for a group of patches that should share the same knot vector (see for example Figure 3). This requires significantly less steps than removing non-conformities for two patches at a time and repeating. A first naïve approach is to take the union of the knot vectors of all patches, denoted by  $\bar{\Xi}$ . This procedure, shown in Figure 4b, does not introduce any geometrical change in the patches. However, in case there are two very close but distinct knots in two different patches, this results in extremely thin elements. In general, taking the union of knot vectors can lead to unevenly spaced knots, which is undesirable. For this reason, if two knots in  $\bar{\Xi}$  are too close according to a prescribed tolerance,

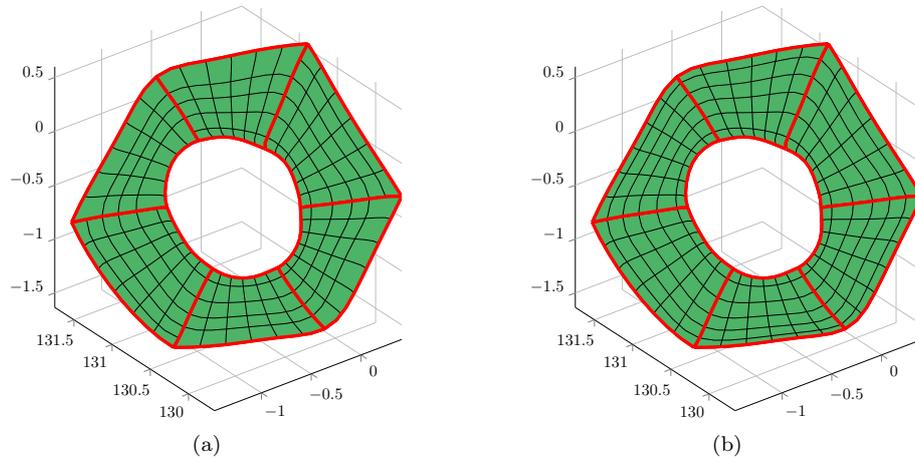


Figure 3: A group of patches before (a) and after (b) the removal of knot-level non-conformities. After the removal of non-conformities, all the patches have the same knot vector along the radial direction.

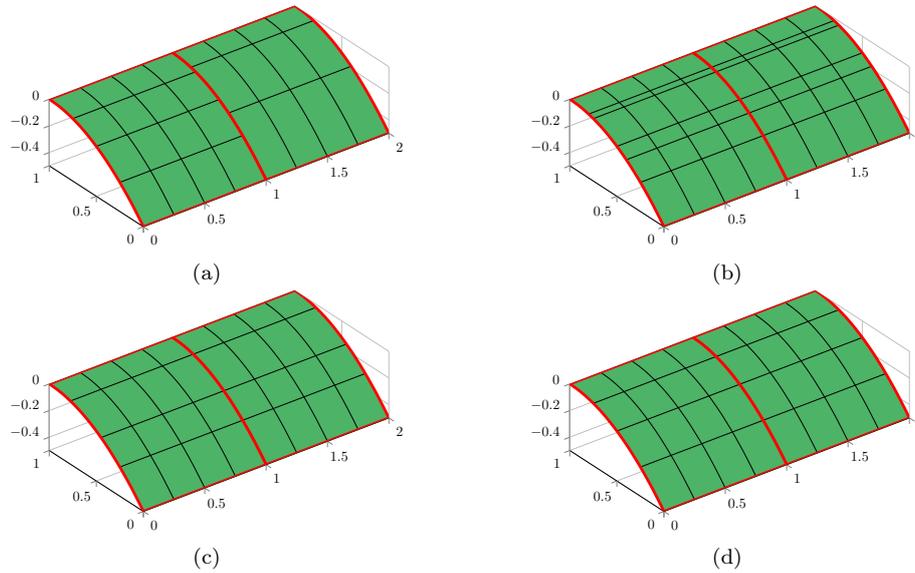


Figure 4: Steps for the solution of a knot-level non-conformity between two patches. Patch boundaries are shown in red. (a): initial non-conforming patches; (b): the knot vectors are joined, introducing very stretched elements; (c): close knots are replaced by their midpoint, improving the element shape; (d): optionally, the knot vectors are forced to be uniform.

they are “merged” into their midpoint (Figure 4c), that is their midpoint is inserted and the two knots are removed using the algorithm described in Section 4.1.4. To decide whether two knots should be merged, their distance  $d$  in the physical space is checked against a threshold  $d_{\text{th}} = \varepsilon_{\text{th}}h$ : if  $d < d_{\text{th}}$ , the knots are removed and their midpoint is inserted, otherwise they are kept as they are. In general, operations involving the removal of a knot introduce geometrical changes in the geometry, and the entity of such changes should be manually assessed to determine whether they are acceptable or not. A large value of  $d_{\text{th}}$  leads to more evenly spaced elements, at the price of more significant geometrical modifications.

Algorithm 5: Removal of a knot-level non-conformity in a group of patches

```

1: procedure SOLVEKNOTLEVELNONCONF(patches,sides,epsilon,h)
   Input patches - the set of patches containing knot-level non-
   conformities
   Input sides - for each patch the parametric side corresponding to
   the interface
   Input epsilon - the threshold parameter for joining knots
   Input h - the desired element size
   Output modifies patches solving non-conformities

2:   n = length(patches)
3:   Xi = []                                     ▷ union of all knot vectors
4:   for i = 1, 2, ..., n do
5:     Xi = union of Xi and patches[i].knots[sides[i]]
6:   end for
7:
8:   for each knot span [xi1,xi2] in Xi do
9:     if |xi1 - xi2| < epsilon * h then
10:      xiNew = 0.5 * (xi1 + xi2)
11:      remove xi1 and xi2 from Xi
12:      insert xiNew into Xi
13:     end if
14:   end for
15:
16:   for i = 1, 2, ..., n do
17:     patches[i] = nrbkntset(patches[i],sides[i],Xi)
18:   end for
19: end procedure

```

Additionally, it is sometimes desirable to obtain a mesh with evenly spaced elements. Indeed, the whole knot vector of the patch can be potentially replaced by a uniform knot vector of the same length (Figure 4d). Again, this operation involves knot removal and insertion. Therefore, it can alter the geometry, and it is useful only if the introduced geometrical changes are not too relevant. This should be checked manually, and this step should be skipped if the geometry is changed in an unacceptable way.

This operation is repeated on all non-conforming patches, until all knot-level non-conformities are solved.

#### 4.1.4 Utility routines for knot vectors manipulation

It may be necessary to remove a knot from a patch, for instance if that knot gives rise to excessively thin elements that would hinder the quality of the numerical solution. In general,

the removal of a knot cannot be carried out without altering the geometry of the patch [60]. However, in case there are multiple knots that are relatively close, the geometrical changes are small and can be acceptable.

Assume to have defined the following routines:

- `hRefineMatrix(knots1, knots2, p)`, to construct the h-refinement matrix  $T$ , that maps control points before knot insertion onto control points after the insertion of one or more knots [26];
- `pseudoInverse(A)`, to compute the least-squares inverse of a matrix.

Then, the removal of a knot can be carried out using the following algorithm:

Algorithm 6: Removal of a knot from a NURBS surface patch

```

1: procedure NRBKNTREM(patch, side, i)
   Input patch - the NURBS patch
   Input side (1 or 2) - the parametric direction to remove the knot
   from
   Input i - the index of the knot to be removed
   Output modifies patch removing the knot

2:   p = polynomial degree of patch
3:   newKnots = patch.knots[side]
4:   remove i-th element from newKnots
5:   T = hRefineMatrix ( newKnots, patch.knots[side], p )
6:
7:   patch.knots[side] = newKnots
8:   patch.ctrlPoints = pseudoInverse(T) * patch.ctrlPoints
9: end procedure

```

The following algorithm can be applied to simultaneously insert and remove multiple knots from a NURBS surface patch.

Algorithm 7: Multiple knot insertion and removal for a NURBS surface patch

```

1: procedure NRBKNTSET(patch, side, newKnots)
   Input patch - the patch to be modified
   Input side (1 or 2) - the parametric side to be modified
   Input newKnots - the desired knot vector
   Output modifies patch imposing the desired knot vector

2:   for each knot xi in newKnots not in patch.knots[side] do
3:     insert xi into patch.knots[side]
4:   end for
5:
6:   for each knot xi in patch.knots[side] not in newKnots do
7:     patch = nrbkntrem(patch, side, xi)
8:   end for
9: end procedure

```

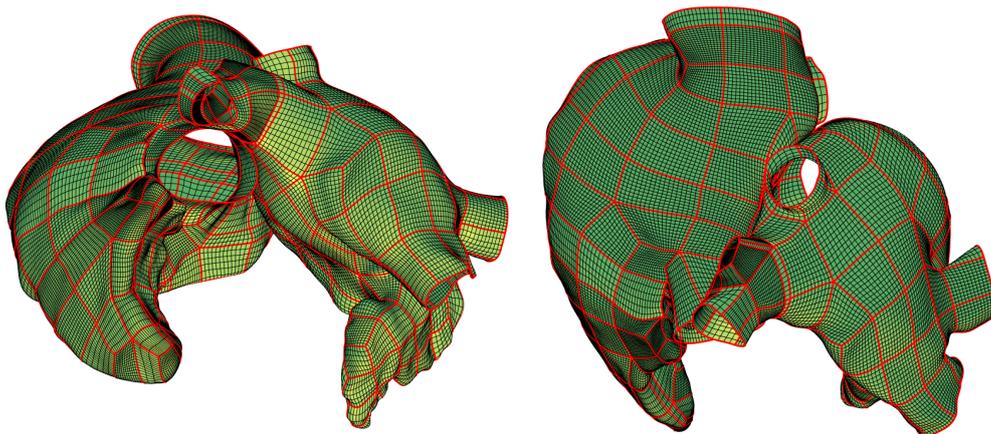


Figure 5: The meshes for the two atria, at a coarse refinement level. Patch boundaries are shown in red.

#### 4.1.5 Application to the human atrial geometries

We implement the above described algorithm in MATLAB relying on the NURBS package [1] for what concerns the treatment of the geometries. The resulting meshes after preprocessing are shown in Figure 5. Our right atrium is comprised of 172 patches for a total of 162'234 elements, while our left atrium is comprised of 145 patches for a total of 121'671 elements. All the patches are parametrized by NURBS functions of degree 3.

Moreover, for both atria, the internal and external surfaces were available, while the simulation was performed only on the external surface. The internal surfaces were manually eliminated using the CAD software SolidWorks [2].

## 4.2 Preprocessing of realistic human left and right ventricle geometries

We also preprocess a geometry representing human left and right ventricles. The source geometry is represented only in terms of its boundary (this is referred to as boundary representation, or B-rep), described by a set of non-conforming NURBS surface patches. Given the non-negligible thickness of the ventricles, we aim at representing them by one or more NURBS solid volume patches, obtained by filling the volume between the epicardial and endocardial surfaces.

The problem of moving from a surface representation of the boundary of a volume to a trivariate parametrization for that same volume is not trivial, and several automatic approaches have been proposed [4, 41, 43, 56, 77, 78].

Our procedure is based on reconstructing in a semi-automatic way the source geometry, up to a small approximation error, with a parametrization independent of the original one. It is possible to impose patch conformity constraints to the resulting parametrization. In particular, we can impose a one-to-one correspondence of patches between the epicardium and the endocardium, that allows to construct in a trivial way the volume patches, and conformity between adjacent patches.

The method is outlined as follows. First, a structured grid of points laying on the surface of interest is obtained by a raycasting method. Then, NURBS surfaces are least-squares fitted on the grid of points. This is done both on the epicardium and on the endocardium. Finally, the epicardial and endocardial patches are connected to obtain the volume patches. The steps are shown for the left ventricle patch in Figure 6.

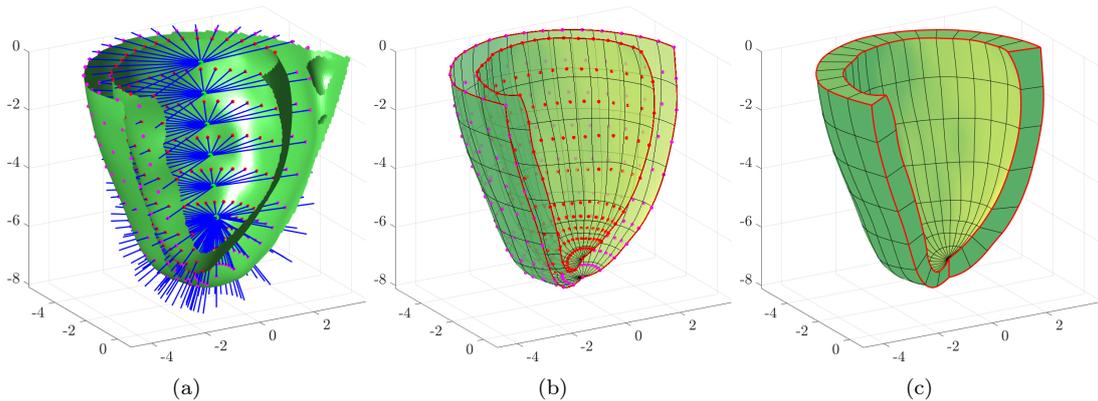


Figure 6: Reconstruction of the left ventricle patch. (a): half-lines and intersection points; green dots are the half-line origin points, blue lines are the half-lines, red and magenta dots are the intersection points between the half-lines and the endocardium and epicardium respectively; the green surface is the source geometry, sectioned for visualization. (b): fitted NURBS surface patches for endocardium and epicardium, alongside the points on which they were fitted; (c): NURBS volume patch for the left ventricle, obtained by linear interpolation of the endocardial and epicardial surface patches.

This procedure still requires to choose manually the subdivision in patches of the geometry, and careful manual tuning for the placement of origin points and directions for raycasting, for the number of points to be used for fitting and for the knot vectors of the fitted surfaces. On the other hand, the grid of points could be provided by other sources, such as experimental measures, so that this approach can be used to construct analysis-suitable geometries in more general settings.

The following sections detail the individual steps of the above-mentioned procedure.

#### 4.2.1 Computation of fitted points

The points at which surfaces are fitted are computed by a raycasting procedure: each of the needed points is obtained by intersecting a half-line, typically starting from the approximate centerline of the chamber being reconstructed and in radial direction, with the source geometry.

The intersection of a half-line with a NURBS surface patch requires the solution of the non-linear system of equations:

$$\sum_{i=1}^n N_{i,p}(\xi) \mathbf{P}_i = \mathbf{o} + \eta \mathbf{r} \quad \xi \in [0, 1], \eta > 0, \quad (2)$$

where  $\mathbf{o}$  and  $\mathbf{r}$  are respectively the origin and direction of the half-line. Efficient iterative methods for the solution of (2) have been proposed [54, 63], but the problem can become expensive, especially as the number of grid points and amount of patches increase.

A set of points is obtained for each patch we want to construct. For each patch, the origin points and directions of the half-lines are manually constructed so that the resulting intersection points form a grid without intersecting edges. To do so it is possible to exploit symmetries in the geometry to be reconstructed (for example, the approximate cylindrical symmetry of the left ventricle in our case). To enforce conformity between adjacent patches, the grids of intersection points used to fit those patches must share an interface of coincident points. This can be obtained by using the same half-line origins and directions for those points when computing the two grids.

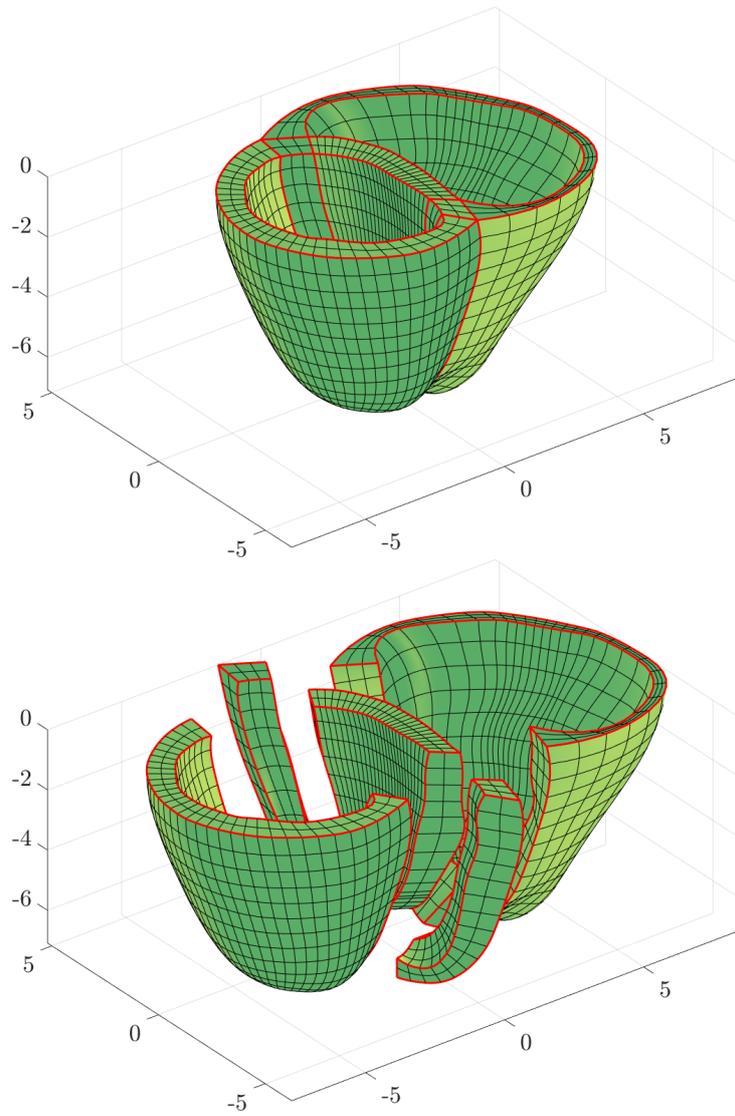


Figure 7: Left and right ventricle geometry, at a coarse refinement level. Patch boundaries are shown in red.

Figure 6a shows as an example the half-lines and intersection points that are obtained for the left ventricle endocardium.

Assuming to have defined a routine `castRayNurbs(nurbs, origin, direction)` computing the intersection, if any, between a half-line and a NURBS surface patch, the following algorithm formalizes this step.

Algorithm 8: Find intersection points between a multipatch NURBS surface and a set of half-lines.

```

1: procedure CASTRAYS(patchess, origins, directions)
   Input patches - collection of patches representing the geometry to be reconstructed
   Input origins - n-by-m-by-3 matrix of half-line origin points
   Input directions - n-by-m-by-3 matrix of half-line direction vectors
   Output points - n-by-m-by-3 matrix of the intersection points

2:   for i = 1, 2, ..., n do
3:     for j = 1, 2, ..., m do
4:       for each patch P in patches do
5:         pt = castRayNurbs(P, origins[i][j], directions[i][j])
6:         if an intersection was found then
7:           points[i][j] = pt
8:         end if
9:       end for
10:    end for
11:  end for
12: end procedure

```

#### 4.2.2 Fitting of NURBS surfaces

The algorithm for the least-squares fit of a NURBS surface to a grid of points is taken from [60]. However, such algorithm chooses knot vectors according to an optimality criterion based on the points to be fitted that aims at obtaining the least distorted parametrization. This would result in different fits at the interface for adjacent patches, that would end up being neither parametrically nor geometrically conforming.

For this reason, we choose to use uniform knot vectors for all patches, even if this is not optimal in the sense of [60], since the obtained result is sufficiently good in terms of distortion.

An example of surface fitted on the points obtained from raycasting is shown in Figure 6b.

#### 4.2.3 Construction of NURBS volumes

Once NURBS surfaces for the epicardium and endocardium are obtained, they can be connected with a volume (see Figure 6c), with a straightforward generalization of the ruled surface construction algorithm described in [60], resulting in a linear interpolation between the two surfaces to be connected.

Since the surface patches are constructed to be geometrically and parametrically conforming, also the volume patches will be. The resulting geometry will then be ready as computational domain for the numerical simulation.

#### 4.2.4 Application to the ventricles' geometry

The ventricles' geometry is reconstructed by five patches: left ventricle, right ventricle, septum and two junctions that connect the other three (see Figure 7). This kind of subdivision allows us to have conforming patches while maintaining a small distortion of the elements.

In both ventricles, the half-line origin points are placed approximately along the centerline, and half-lines directions are chosen radially. This allows to obtain a regular parametrization of both the endocardium and the epicardium, especially for the left ventricle, whose cross section is approximately circular. Due to the more complex shape of the right ventricle, its parametrization is slightly more irregular, in the sense that it leads to unevenly spaced elements, and it is more sensitive to the choice of half-line origins and directions. The polynomial degree used to parametrize the patches is equal to 2.

## 5 Mathematical models for cardiac electrophysiology

In this section we introduce the Monodomain equation and the ionic models, and we discuss their numerical discretization both in space and time.

### 5.1 Monodomain equation and ionic model

The evolution in space and time of the transmembrane potential in the cardiac tissue is described by means of the Monodomain model [22]. Let  $\Omega \subset \mathbb{R}^3$  be the computational domain,  $u(\mathbf{x}, t)$  the transmembrane potential at the space coordinate  $\mathbf{x} \in \Omega$  and at time  $t \in (0, T)$ ,  $C_m$  the total membrane capacitance and  $\chi$  the membrane surface to volume ratio. The Monodomain equation coupled with a generic ionic model reads:

$$\begin{cases} \chi C_m \frac{\partial u}{\partial t} - \nabla \cdot (D \nabla u) + \chi I_{\text{ion}}(u, \mathbf{w}, \mathbf{c}) = I_{\text{app}} & \text{in } \Omega \times (0, T), \\ \frac{\partial \mathbf{w}}{\partial t} = \mathbf{F}(u, \mathbf{w}) & \text{in } \Omega \times (0, T), \\ \frac{\partial \mathbf{c}}{\partial t} = \mathbf{G}(u, \mathbf{w}, \mathbf{c}) & \text{in } \Omega \times (0, T), \\ D \nabla u \cdot \mathbf{n} = 0 & \text{on } \partial\Omega \times (0, T), \\ u = u_0 & \text{in } \Omega \times \{0\}, \\ \mathbf{w} = \mathbf{w}_0 & \text{in } \Omega \times \{0\}, \\ \mathbf{c} = \mathbf{c}_0 & \text{in } \Omega \times \{0\}, \end{cases} \quad (3)$$

where  $\mathbf{n}$  is the outward directed unit vector normal to  $\partial\Omega$ ,  $C_m = 0.001 \text{ mF/cm}^2$ ,  $\chi = 1000 \text{ cm}^{-1}$ .  $u_0$ ,  $\mathbf{w}_0$ ,  $\mathbf{c}_0$  and  $I_{\text{app}}$  are the given initial conditions and the applied current respectively. System (3) defines the propagation in space and time of the transmembrane potential  $u(\mathbf{x}, t)$  at the tissue level along with the ionic behavior, at the cellular level, by means of  $\mathbf{w}$  and  $\mathbf{c}$  [22]. We refer to  $\mathbf{w}$  as gating variables, which represent fractions of open ionic channels, and to  $\mathbf{c}$  as intracellular ionic concentrations. A variety of ionic models have been proposed in literature [5, 15, 29, 53, 68]. We consider here the LR model for the ventricles [53] and the CRN for the atria [29]. Both models reproduce the evolution of the intracellular calcium concentration, which is a crucial ionic species that triggers the mechanical activation of the cardiac tissue [49]. A comprehensive list of the existing ionic models can be found in [22]. The problem is closed with homogeneous Neumann boundary conditions for the transmembrane potential, to keep into account the assumption of electrically isolated domain, and initial conditions for both transmembrane potential and ionic model variables.

$D$  expresses the conductivity of the tissue and can be used to represent its electric anisotropy, accounting for the presence of cardiac fibers. Assuming axisymmetric conductivity and denoting by  $\mathbf{a}_1(\mathbf{x})$  the local direction of the fibers, the tensor can be computed as

$$D(\mathbf{x}) = \sigma_t(\mathbf{x})I + (\sigma_l(\mathbf{x}) - \sigma_t(\mathbf{x}))\mathbf{a}_1(\mathbf{x})\mathbf{a}_1(\mathbf{x})^T,$$

where  $\sigma_l = 0.012\,042\text{ S cm}^{-1}$  and  $\sigma_t = 0.001\,761\text{ S cm}^{-1}$  are the conductivities in the fibers direction and in the direction orthogonal to fibers respectively.

The Monodomain equation is a simplified version of the Bidomain model [20, 22], derived under the assumption that the intracellular and extracellular conductivities of the Bidomain model have equal anisotropy ratio. Nonetheless, the results obtained from the Monodomain equation are accurate in physiological conditions and in many pathological cases, and this model is widely used in literature [22, 34, 35, 39, 62]. The advantage of the Monodomain equation resides in its reduced complexity in comparison with the Bidomain model, which can be expressed with both a parabolic-elliptic and a parabolic-parabolic PDE system. As a consequence, the Monodomain equation is much less demanding from the computational viewpoint than the Bidomain model. Moreover, the system to be solved for the Monodomain equation is significantly better conditioned than the one of the Bidomain model. For these reasons, we used the Monodomain equation in our numerical simulations. A comparison of the two different models is carried out for example in [14, 22, 24].

## 5.2 Space discretization

The weak formulation of the Monodomain equation, see e.g. [65], is here discretized in space using multipatch IGA. We look for an approximation  $u_h$  of  $u \in V = H^1(\Omega)$  in (3) in the finite dimensional space  $V_h \subset V$ , so that

$$u_h(\mathbf{x}, t) = \sum_{j=1}^{N_h} U_j(t) R_j(\mathbf{x}),$$

where  $U_j(t)$ ,  $j = 1, \dots, N_h$  are suitable time-dependent control variables and  $R_j(\mathbf{x})$ ,  $j = 1, \dots, N_h$  are NURBS-based basis functions, as illustrated in Section 3. Then we approximate the first equation of (3) by:

$$\chi C_m \sum_{j=1}^{N_h} \frac{\partial U_j}{\partial t} (R_j, R_i) + \sum_{j=1}^{N_h} U_j a(R_j, R_i) + \chi (I_{\text{ion}}(u_h, \mathbf{w}, \mathbf{c}), R_i) = (I_{\text{app}}, R_i) \quad \forall i = 1, \dots, N_h, \quad (4)$$

where  $(\cdot, \cdot)$  denotes the  $L^2$  scalar product and

$$a : V \times V \rightarrow \mathbb{R}, \quad a(\varphi, \psi) = \int_{\Omega} D \nabla \varphi \cdot \nabla \psi \, d\Omega.$$

Equation 4 can be rewritten as a system of ODEs:

$$\chi C_m M \frac{d}{dt} \mathbf{U}(t) + K \mathbf{U}(t) + \chi \mathbf{I}_{\text{ion}}(\mathbf{U}(t), \mathbf{w}(t), \mathbf{c}(t)) = \mathbf{I}_{\text{app}}(t), \quad (5)$$

where  $\mathbf{U}(t)$  denotes the vector of entries  $U_j$ ,  $M$  is the mass matrix of entries  $M_{ij} = (R_j, R_i)$ ,  $K$  is the stiffness matrix of entries  $K_{ij} = a(R_j, R_i)$  and  $\mathbf{I}_{\text{ion}}(\mathbf{U}(t), \mathbf{w}(t), \mathbf{c}(t))$  is a vector of entries  $(\mathbf{I}_{\text{ion}})_i = (I_{\text{ion}}(u_h(t), \mathbf{w}(t), \mathbf{c}(t)), R_i)$ .

For the computation of the gating variables  $\mathbf{w}$  and ionic concentrations  $\mathbf{c}$  we use Gaussian integration [38, 51, 57]. We aim at computing

$$(\mathbf{I}_{\text{ion}}(t))_i = (I_{\text{ion}}(u_h(t), \mathbf{w}(t), \mathbf{c}(t)), R_i) = \int_{\Omega} I_{\text{ion}}(u_h(t), \mathbf{w}(t), \mathbf{c}(t)) R_i \, d\Omega.$$

We employ a Gaussian quadrature rule of order  $p+1$ , where  $p$  is the polynomial degree of the basis functions, to approximate the integral above. We denote by  $N_e$  the number of elements in

$\Omega$ .  $\{K_j\}_{j=1}^{N_e}$  are the elements,  $\{\mathbf{x}_m^j\}_{m=1}^s$  are the Gaussian quadrature points on the  $j$ -th element, with associated weights  $\{\omega_m\}_{m=1}^s$ , being  $s$  the number of quadrature points per element. Using the introduced notation we get:

$$\begin{aligned} (\mathbf{I}_{\text{ion}}(t))_i &= \int_{\Omega} I_{\text{ion}}(u_h(t), \mathbf{c}(t), \mathbf{w}(t)) R_i d\Omega = \sum_{j=1}^{N_e} \int_{K_j} I_{\text{ion}}(u_h(t), \mathbf{w}(t), \mathbf{c}(t)) R_i d\Omega \approx \\ &\approx \sum_{j=1}^{N_e} \sum_{m=1}^s I_{\text{ion}}(u_h(\mathbf{x}_m^j, t), \mathbf{w}(\mathbf{x}_m^j, t), \mathbf{c}(\mathbf{x}_m^j, t)) R_i(\mathbf{x}_m^j) \omega_m. \end{aligned} \quad (6)$$

Therefore, for the purpose of computing the ionic current term in the discrete weak formulation (5), it is sufficient to compute the values of  $\mathbf{w}$  and  $\mathbf{c}$  at the Gauss points. This can be done by solving at each quadrature point the following ODE problem:

$$\begin{cases} \frac{d}{dt} \mathbf{w}(\mathbf{x}_m^j, t) = \mathbf{F}(u_h(\mathbf{x}_m^j, t), \mathbf{w}(\mathbf{x}_m^j, t)) & t \in (0, T), \\ \frac{d}{dt} \mathbf{c}(\mathbf{x}_m^j, t) = \mathbf{G}(u_h(\mathbf{x}_m^j, t), \mathbf{w}(\mathbf{x}_m^j, t), \mathbf{c}(\mathbf{x}_m^j, t)) & t \in (0, T), \\ \mathbf{w}(\mathbf{x}_m^j, 0) = \mathbf{w}_0(\mathbf{x}_m^j), \\ \mathbf{c}(\mathbf{x}_m^j, 0) = \mathbf{c}_0(\mathbf{x}_m^j). \end{cases}$$

The solution of the ionic model reduces to the solution of a set of independent ODEs, one for each quadrature node of the mesh.

Alternative approaches to the discretization of the ionic model are based on applying Galerkin method to the ionic model equations as well, and are known as nodal interpolation methods [51, 57, 58, 64]. With respect to such methods, Gaussian integration saves computational resources by avoiding the interpolation of the ionic variables, at the price of a higher memory storage due to the need to save ionic variables at each quadrature node.

IGA has been applied to the discretization of the Monodomain or Bidomain model in [20, 57, 59], highlighting the good properties of the method in terms of accuracy of the action potential propagation [57] and numerical dispersion [59]. However, the use of a single patch in those cases limits the applicability of the method to complex, realistic geometries. Such limit can be overcome by using multiple patches.

### 5.3 Time discretization

For the discretization of time derivatives, both for the Monodomain and ionic model equations, we use Backward Differentiation Formulas (BDF) of order 2, suitable for the solution of stiff differential equations [64]. We treat the ionic current  $\mathbf{I}_{\text{ion}}$ , i.e. the only non linear term in the Monodomain equation, using a semi-implicit scheme. We also use suitable second order extrapolations (denoted in the following with the subscript ext). The resulting, fully discrete problem is the following: at each timestep,

1. Solve  $\forall m = 1, \dots, s$  and  $\forall j = 1, \dots, N_e$ :

$$\begin{aligned} \alpha \frac{\mathbf{w}_{h,m}^{j,k+1} - \mathbf{w}_{\text{BDF},m}^{j,k+1}}{\Delta t} &= \mathbf{F}(u_{m,\text{ext}}^{j,k+1}, \mathbf{w}_{\text{ext}}^{j,k+1}), \\ \alpha \frac{\mathbf{c}_m^{j,k+1} - \mathbf{c}_{\text{BDF},m}^{j,k+1}}{\Delta t} &= \mathbf{G}(u_{m,\text{ext}}^{j,k+1}, \mathbf{w}_{\text{ext}}^{j,k+1}, \mathbf{c}_{\text{ext}}^{j,k+1}), \end{aligned}$$

in which  $u_{m,\text{ext}}^{j,k+1}$  denotes the evaluation of the extrapolated numerical solution for the transmembrane potential at the quadrature point  $\mathbf{x}_m^j$ ;

2. compute the ionic current as in (6):

$$I_{\text{ion}}^{k+1} = \sum_{j=1}^{N_e} \sum_{m=1}^s I_{\text{ion}}(u_{m,\text{ext}}^{j,k+1}, \mathbf{w}_{h,m}^{j,k+1}, \mathbf{c}_{h,m}^{j,k+1}) R_i(\mathbf{x}_m^j) \omega_m;$$

3. solve the fully discrete Monodomain equation:

$$\chi C_m M \alpha \frac{\mathbf{U}^{k+1} - \mathbf{U}_{\text{BDF}}^{k+1}}{\Delta t} + K \mathbf{U}^{k+1} = \mathbf{I}_{\text{app}} - \chi \mathbf{I}_{\text{ion}}.$$

Although we do not expect the time discretization to be of order 2, due to the splitting of ionic and monodomain equations, the use of BDF of order 2 increases the accuracy in time. This allows to obtain accurate results with a larger timestep than the one required by lower order time discretization schemes, such as implicit Euler, without additional computational costs [61].

## 6 Numerical results

We report hereby the numerical results of our electrophysiological simulations on both atria and ventricles.

All the numerical simulations require the reconstruction of fibers direction. This can be achieved by interpolating diffusion tensor imaging (DTI) data on the computational mesh [22, 39]. However, in case DTI data are not available or too noisy, it is necessary to reconstruct the fibers orientation through suitable rule-based algorithms [11, 36, 69]. For each geometry, we discuss the used algorithm for the development of suitable fiber fields.

We have developed our code in the framework of the C++ IGA library Isoglib. We use the HPC infrastructure made available at MOX, Department of Mathematics, Politecnico di Milano.

### 6.1 Atria

#### 6.1.1 Fiber generation

The fibers field on the atria is reconstructed following a rule-based method [57, 59, 61]. We solve the following Laplace-Beltrami problem:

$$\begin{cases} -\Delta_{\Omega} \varphi = f, & \text{in } \Omega, \\ \nabla_{\Omega} \varphi \cdot \mathbf{n} = 0, & \text{on } \partial\Omega, \end{cases} \quad (7)$$

where  $\nabla_{\Omega}$  and  $\Delta_{\Omega}$  indicate the surface gradient operator and the Laplace-Beltrami operator respectively, whereas  $f$  is a scalar field satisfying the following compatibility condition:

$$\int_{\Omega} f \, d\Omega = 0.$$

Given the homogeneous Neumann boundary condition, the solution  $\varphi$  to (7) is defined up to a constant. Therefore, we add the constraint  $\int_{\Omega} \varphi \, d\Omega = 0$ , which is imposed with a Lagrange

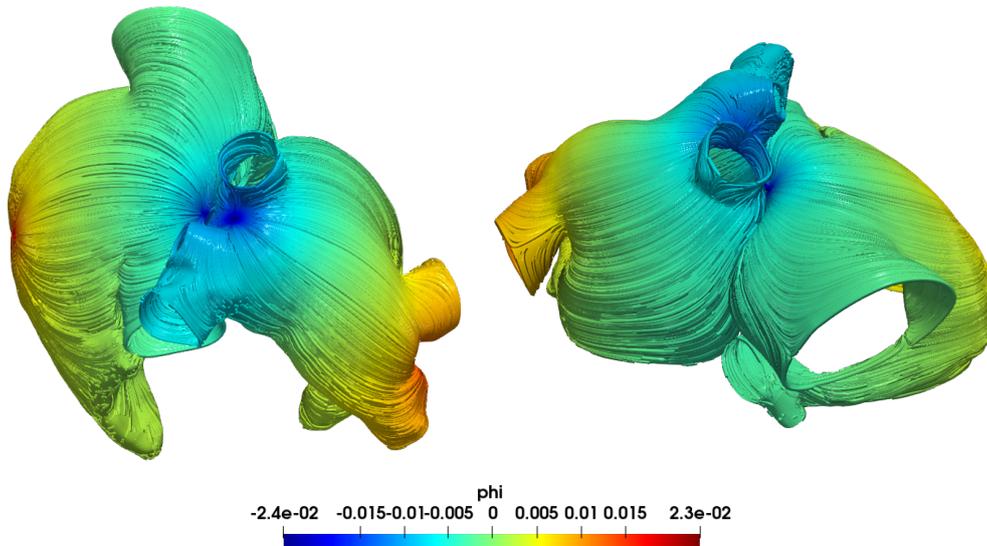


Figure 8: The computed fiber field for the atria; the color is the solution of the Laplace-Beltrami problem (7).

multiplier from the numerical viewpoint. Starting from  $\varphi$ , we can compute the direction of the fibers as the normalized gradient of the solution:

$$\mathbf{f}_0 = \frac{\nabla_{\Omega} \varphi}{|\nabla_{\Omega} \varphi|}.$$

Thanks to the homogeneous Neumann conditions, the resulting field is tangent to the boundary of the atria (corresponding to the cutting sections of the incoming and outgoing vessels and valves). Function  $f$  is defined as follows:

$$f(\mathbf{x}) = \begin{cases} 1 & \text{if } |\mathbf{x} - \mathbf{x}_1| < \delta, \\ -1 & \text{if } |\mathbf{x} - \mathbf{x}_2| < \delta, \\ 0 & \text{otherwise,} \end{cases}$$

where the two points  $\mathbf{x}_1$  and  $\mathbf{x}_2$  are defined on each of the atria following [59]: on the left atrium, one point is located between the left pulmonary veins, the other is between the right pulmonary veins. On the right atrium, one point is located on the auricle, the other is on the top part, opposite to the tricuspid valve. The resulting fibers field is shown in Figure 8.

### 6.1.2 Electrophysiology

Atria electrophysiology is simulated using the Monodomain equation coupled with the CRN ionic model. The simulation ran for 1 s, with a timestep  $\Delta t = 10 \mu\text{s}$ . The mesh for the atria is comprised of 283'905 elements (162'234 for the right atrium, 121'671 for the left), corresponding to a mesh size of about  $h = 0.0005$  m. For the atrial simulation, we use two cluster nodes endowed with 20 cores each (Intel Xeon ES-2640 CPUs) and 128GB of available RAM.

The linear system deriving from the discretization of the Monodomain equation is solved using GMRES method preconditioned with ILUT factorization [67, 70], with a tolerance of  $10^{-6}$  on the normalized residual.

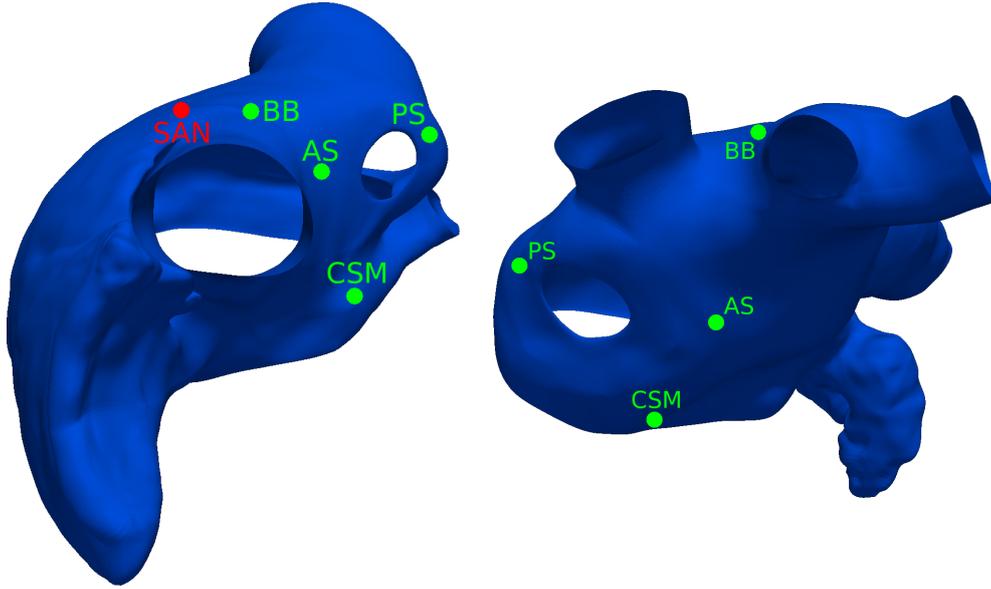


Figure 9: Stimulation point corresponding to the SAN on the right atrium and position of the muscular bundles connecting the two atria: Bachmann's bundle (BB), anterior septum (AS), posterior septum (PS), coronary sinus musculature (CSM).

The right atrium is stimulated in correspondence of the SAN by an applied current of 100 mA for 1 ms. This is meant to mimic its pacemaker role. The stimulation of the left atrium, on the other hand, is triggered according to the electric pattern of the right atrium, to simulate the muscular bundles that connect electrically the two atria (Bachmann's bundle, anterior and posterior septum, coronary sinus musculature [71]). Following [59], for each bundle, we check when the transmembrane potential reaches a suitable threshold ( $u > 0$  mV) at the junction between the bundle and the right atrium. When it happens, we apply a stimulus to the corresponding point on the left atrium. As for the left and right ventricles, stimuli, both on the right and on the left atrium, are applied as external currents of 100 mA with a duration of 1 ms. The locations of bundles and stimulation points are shown in Figure 9.

In Figure 10 we report the evolution of the transmembrane potential and intracellular calcium concentration. From Figure 11a we can observe how the front propagates preferentially in the direction of the fibers, in accordance with the model. The wave front is captured smoothly and is not hindered by patch interfaces or collisions (Figure 11b).

In Figure 12 the evolution of the transmembrane potential at three points on the atria is reported. It can be seen how the action potential behaves in the same way at all the points, up to a time shift due to different activation times, reproducing correctly the atrial hump before the plateau phase.

## 6.2 Left and right ventricles

### 6.2.1 Fiber generation

The fibers on the left and right ventricles are generated according to the rule-based algorithm by Bayer et al. [11], and are reported in Figure 13a. The algorithm is based on solving four Laplace problems on the ventricular geometry, with different boundary conditions, whose solutions are

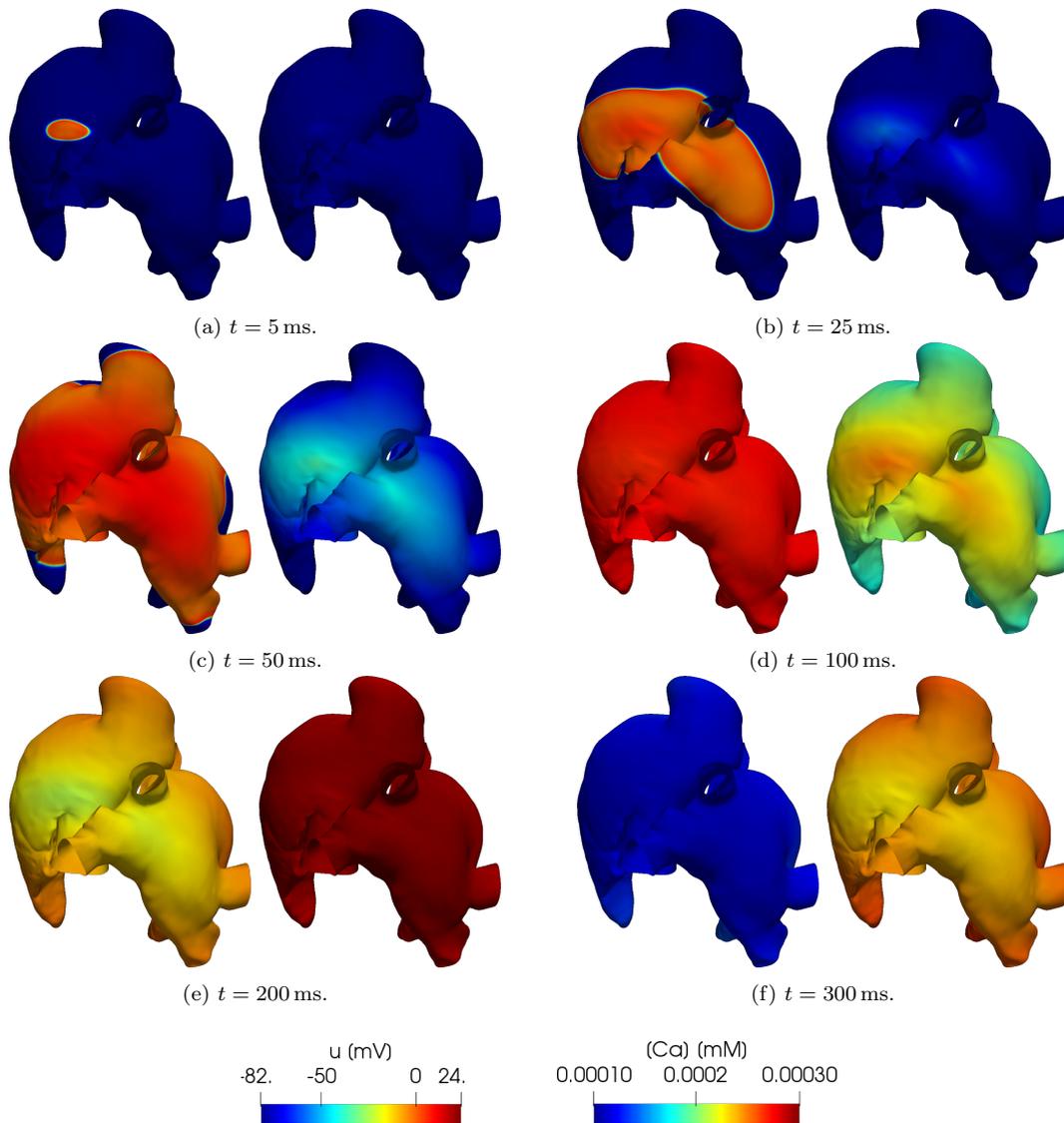


Figure 10: Transmembrane potential (left) and intracellular calcium concentration (right) on the atria, computed using Monodomain equation coupled with the CRN ionic model.

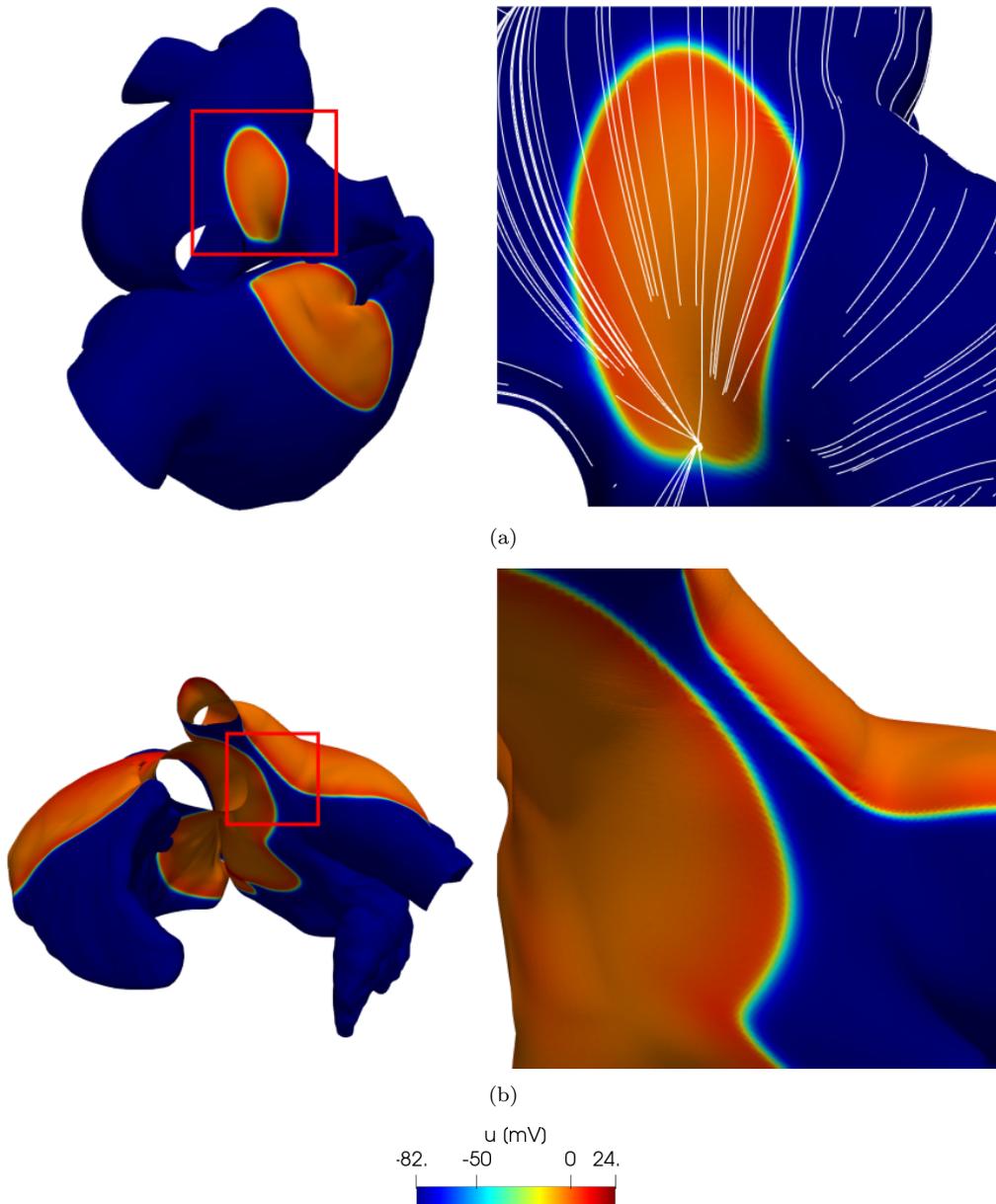


Figure 11: (a): close-up of the electrophysiological simulation in the left atrium, near the location of Bachmann's bundle; the color is the transmembrane potential, whereas white lines are the fibers; (b): close-up of the electrophysiological simulation in the left atrium, highlighting the collision between different fronts. We can also appreciate the smoothness of the transmembrane potential wave propagation.

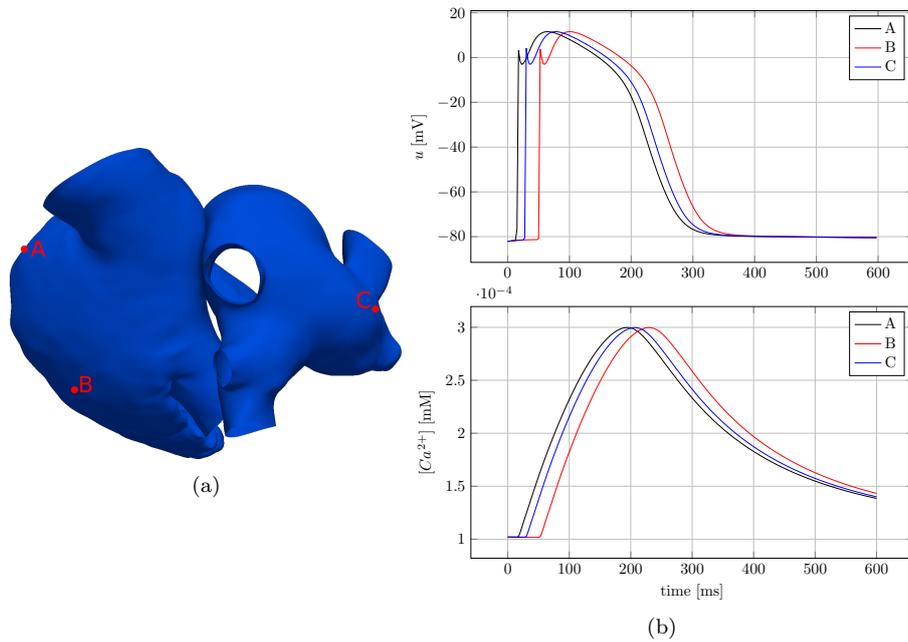


Figure 12: Evolution of both transmembrane potential and intracellular calcium concentration at three points on the atria, according to the Monodomain and CRN models. The three points, shown in (a), were chosen sufficiently far from the SAN and from the interatrial bundles on the left atrium, to minimize the effect of the applied currents.

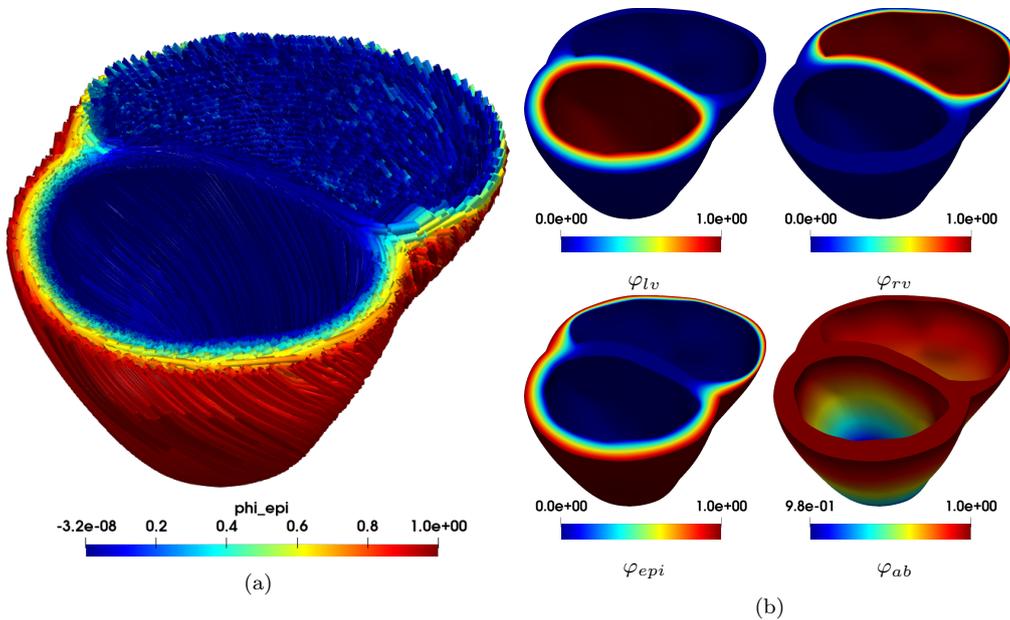


Figure 13: (a): computed fiber field for the left and right ventricles; (b): solutions of the Laplace problems involved in the definition of the fibers field.

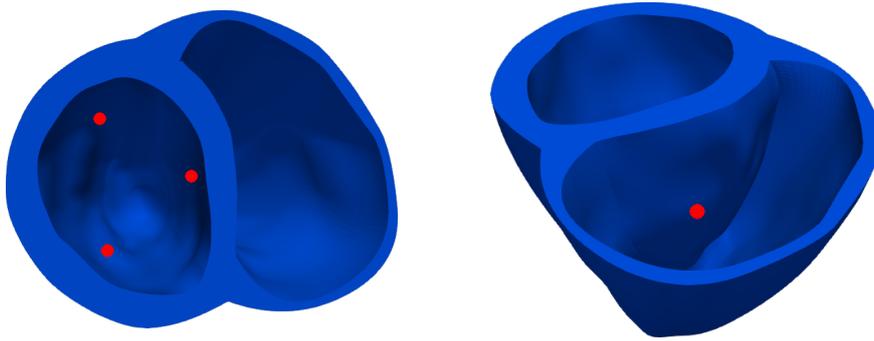


Figure 14: Stimulation points on the left and right ventricles.

depicted in Figure 13b.

### 6.2.2 Electrophysiology

Electrophysiological simulations on the left and right ventricles is simulated by the Monodomain equation coupled with the LR ionic model. The simulation ran for 800 ms, with a timestep  $\Delta t = 10 \mu\text{s}$ . The mesh representing left and right ventricles is composed of 801'216 elements, corresponding to a mesh size of about  $h = 0.0014\text{m}$ . We use 48 cores on a single cluster node, endowed with Intel Xeon Platinum 8160 processors and a total amount of 1792GB of available RAM.

The linear system resulting from the discretization is solved by the GMRES method and ILUT preconditioner [67, 70], with a tolerance of  $10^{-6}$ .

For the application of the stimuli, we qualitatively replicate the effect of the Purkinje network by applying a current at three points of the left ventricle endocardium and in one point of the right ventricle endocardium, as shown in Figure 14. Each stimulus consists of an applied current of 100 mA for 1 ms.

In Figure 15 we report the space evolution of transmembrane potential and calcium concentration on the left and right ventricles for different timesteps of the numerical simulation. In Figure 16 we depict the evolution in time of the same two electrophysiological variables in two different points of the mesh. We can appreciate the detailed representation of all the action potential phases, including both the upstroke and the downstroke just after depolarization. Moreover, the calcium behavior in time follows the LR ventricular physiological dynamics, which is significantly different from the atrial ones, detected by the CRN ionic model.

## 6.3 Simulating the full heart

Computational models for the electrophysiology of the whole heart can be found in [8, 9, 61, 72]. We show here the results obtained by combining the simulation of the atria with the one of the ventricles. For the sake of simplicity, we do not consider the outflow tracts, nor the papillary muscles in the geometrical reconstruction of the ventricles, as done in Section 6.2. Indeed, these cardiac structures account for minor effects on electrophysiological simulations without rhythm disorders.

The simulations on the atria and the ventricles are run separately, by properly timing the stimuli to replicate the delay between the activation of the atria and of the ventricles (due to the effect of the atrioventricular node and of the Purkinje network, which are not modeled explicitly). This is achieved by applying the stimuli on the ventricles approximately 200 ms after

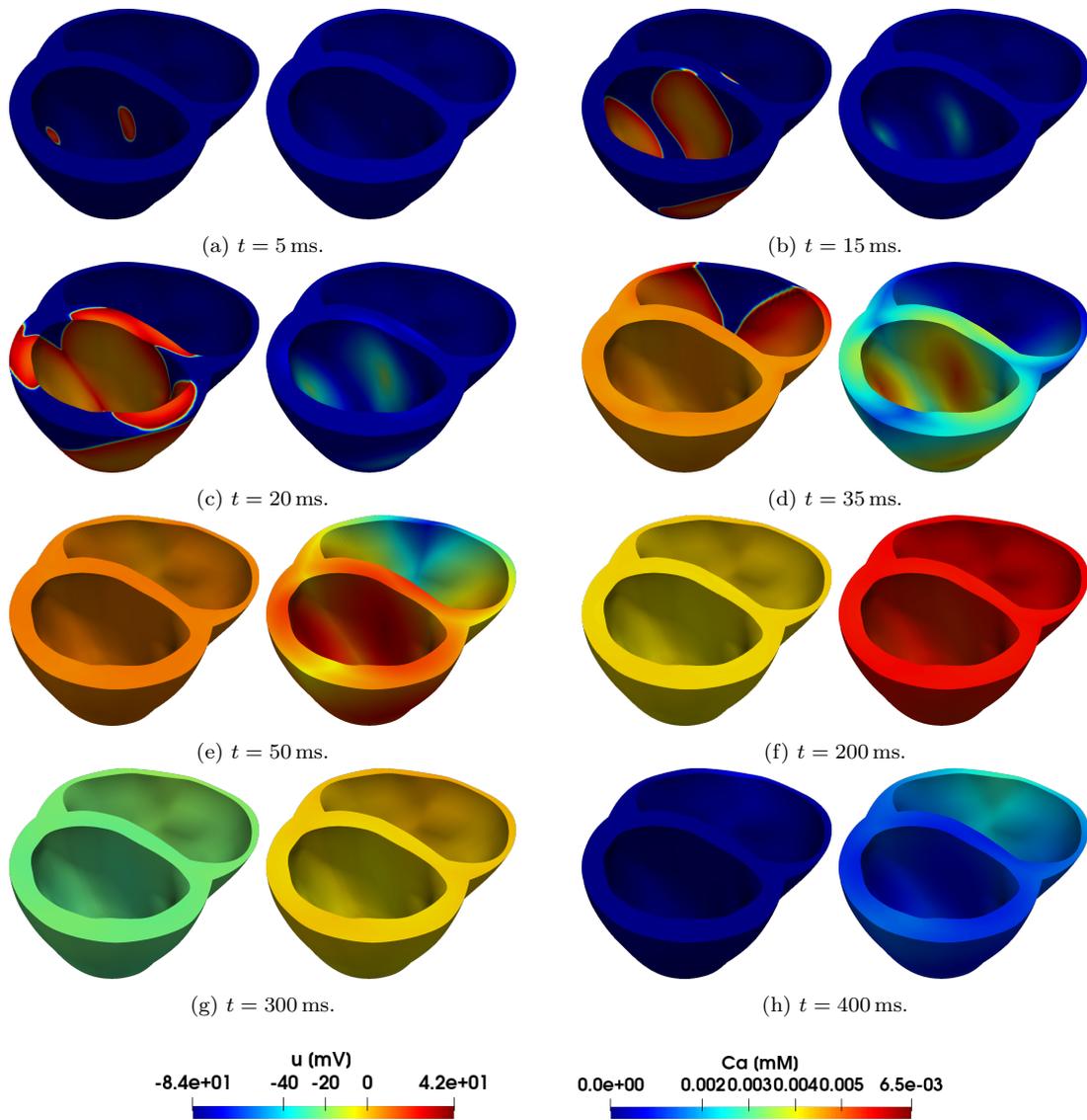


Figure 15: Transmembrane potential (left) and intracellular calcium concentration (right) on the left and right ventricles, computed using Monodomain equation coupled with the LR ionic model.

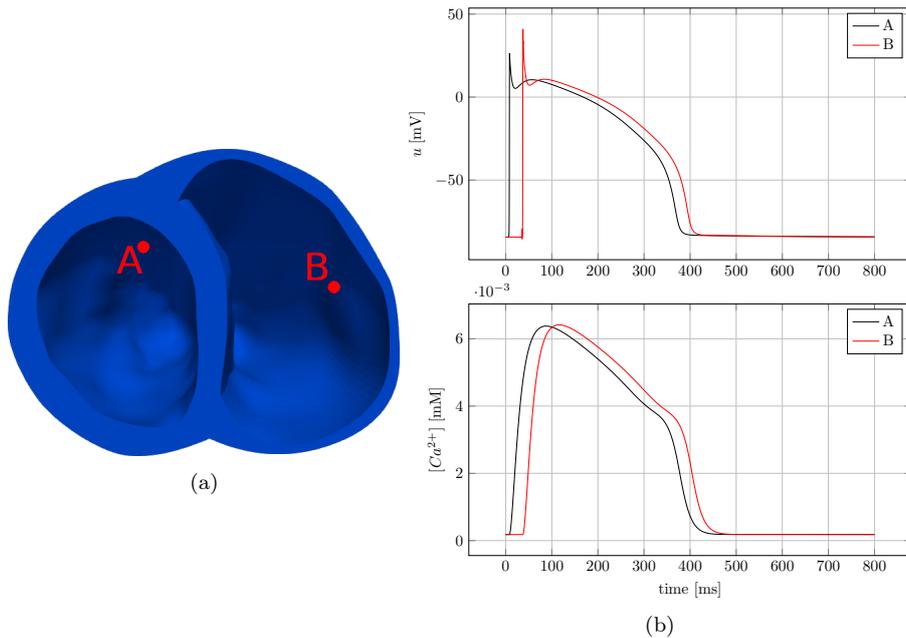


Figure 16: Evolution of both transmembrane potential and intracellular calcium concentration at two points on the left and right ventricles, according to the Monodomain and LR models. The two points, shown in (a), were chosen sufficiently far from the stimulation points, to minimize the effect of the applied currents.

applying them on the atria. Results are reported in Figure 17, where we display the evolution of both the transmembrane potential and the calcium concentration over a realistic four chambers representation of the human heart, considering one full cardiac cycle. Gating and ionic variables are described using a different ionic model according to the specific chamber at hand, which leads to a time development of all electrophysiological variables specialized on the different parts of the heart, as shown in both Figure 12 and Figure 16.

## 7 Conclusions

In this work, we apply multipatch IGA to the spatial approximation of the Monodomain model on realistic computational models of human atria and ventricles. We propose different techniques for the preprocessing of CAD geometries that we employ to construct multipatch parametrizations of realistic cardiac geometries. These techniques result in IGA conforming multipatch surface and volumetric representations of the computational domain, that can then be used for the discretization of differential problems modeling cardiac physiology. Then, we perform numerical simulations of cardiac electrophysiology both with atria and ventricles, using the Courtemanche-Ramirez-Nattel and the Luo-Rudy ionic models respectively to reproduce the physiological electrical behavior both at a macroscopic scale and at the level of single cardiomyocytes. In both cases we use suitable rule-based algorithms to reconstruct the cardiac muscular fibers. By inspecting our numerical results, we find that the high mesh quality and regularity that can be obtained by the multipatch IGA technique, together with the high-order continuity featured by NURBS basis functions, result in smooth excitation fronts and an accurate solution of the

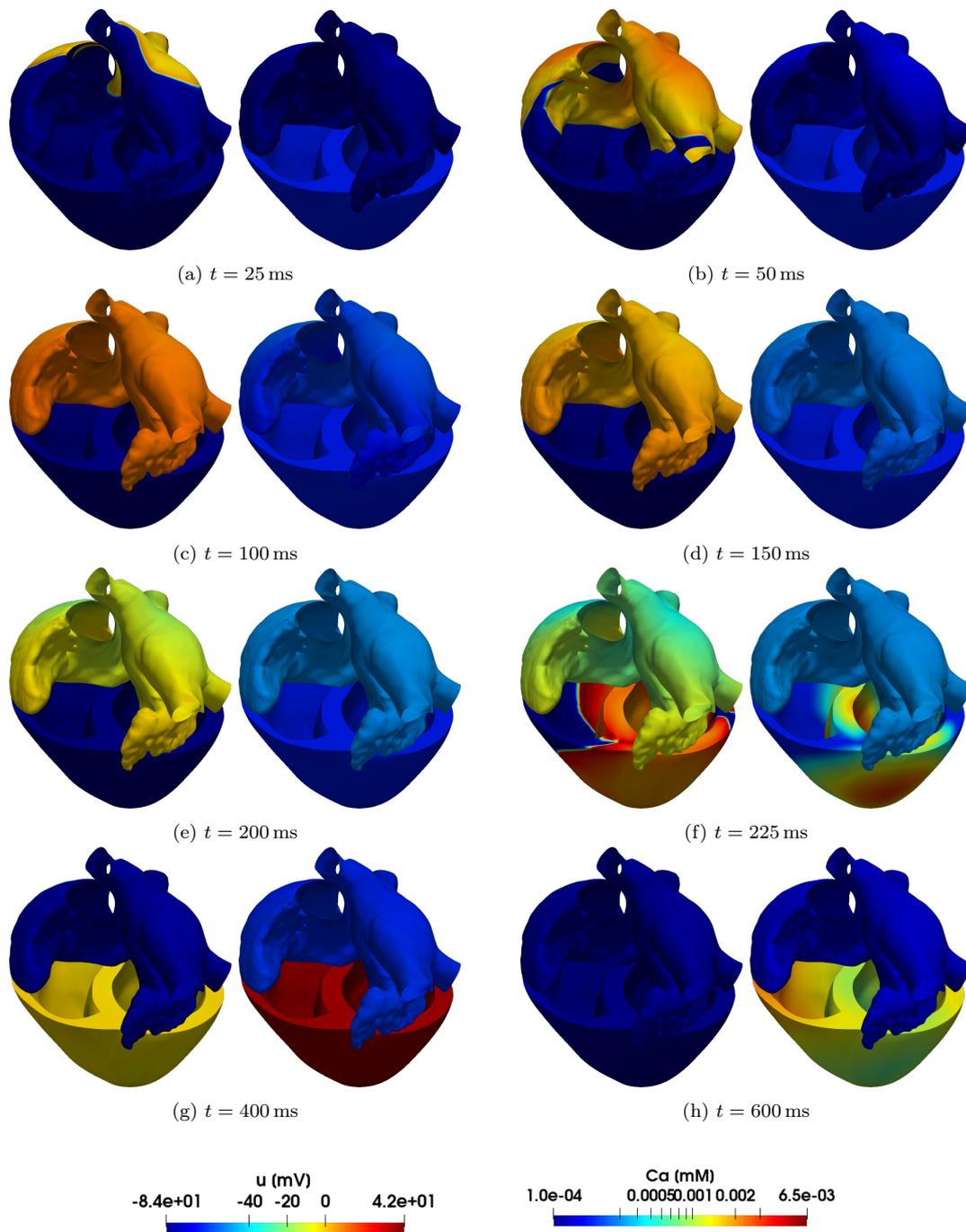


Figure 17: Transmembrane potential (left) and calcium concentration (right) for the whole heart. The color scale for the calcium concentration is logarithmic, in order to accommodate the different orders of magnitude predicted by the CRN and the LR ionic models.

electrophysiology problem, thus improving the control of numerical dispersion enabled by IGA (which was formerly highlighted in [57, 59] for the single patch case). For this reason multipatch IGA on complex domains proves to be more suitable than FEM for wave propagation problems involving multiple time and space scales.

## 8 Acknowledgements

This project has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement No 740132, iHEART - An Integrated Heart Model for the simulation of the cardiac function, P.I. Prof. A. Quarteroni).



## References

- [1] NURBS package on Octave Forge. <https://octave.sourceforge.io/nurbs/index.html>.
- [2] SolidWorks - 3D CAD design software. <https://www.solidworks.com/>.
- [3] Zygote 3D Human Anatomy Models. <https://www.zygote.com>.
- [4] H. Al Akhras, T. Elguedj, A. Gravouil, and M. Rochette. Isogeometric analysis suitable trivariate NURBS models from standard B-Rep models. *Computer Methods in Applied Mechanics and Engineering*, 307:256–274, 2016.
- [5] R. R. Aliev and A. V. Panfilov. A simple two-variable model of cardiac excitation. *Chaos, Solitons & Fractals*, 7:293–301, 1996.
- [6] P. Antolin, A. Buffa, and M. Martinelli. Isogeometric Analysis on V-reps: first results. *Computer Methods in Applied Mechanics and Engineering*, 355:976–1002, 2019.
- [7] P. Antolin, A. Buffa, R. Puppi, and X. Wei. Overlapping Multi-Patch Isogeometric method with minimal stabilization. *ArXiv:1912.06400*, 2019.
- [8] C. Augustin, A. Neic, M. Liebmann, A. Prassl, S. Niederer, G. Haase, and G. Plank. Anatomically accurate high resolution modeling of human whole heart electromechanics: A strongly scalable algebraic multigrid solver method for nonlinear deformation. *Journal of Computational Physics*, 305:622–646, 2016.
- [9] B. Baillargeon, N. Rebelo, D. Fox, R. Taylor, and E. Kuhl. The Living Heart Project: A robust and integrative simulator for human heart function. *European Journal of Mechanics - A/Solids*, 48:38–47, 2014.
- [10] A. Bartzaghi, L. Dede’, and A. Quarteroni. Isogeometric analysis of high order partial differential equations on surfaces. *Computer Methods in Applied Mechanics and Engineering*, 295:446–469, 2015.
- [11] J. D. Bayer, R. C. Blake, G. Plank, and N. A. Trayanova. A novel rule-based algorithm for assigning myocardial fiber orientation to computational models. *Annals of Biomedical Engineering*, 40:2243–2254, 2012.
- [12] Y. Bazilevs, L. Beirao da Veiga, J. A. Cottrell, T. J. R. Hughes, and G. Sangalli. Isogeometric Analysis: Approximation, stability and error estimates for h-refined meshes. *Mathematical Models and Methods in Applied Sciences*, 16:1031–1090, 2006.

- [13] Y. Bourgault, Y. Coudiere, and C. Pierre. Existence and uniqueness of the solution for the bidomain model used in cardiac electrophysiology. *Nonlinear Analysis: Real World Applications*, 10:458–482, 2009.
- [14] Y. Bourgault and C. Pierre. Comparing the bidomain and monodomain models in electrocardiology through convergence analysis. *HAL Archives-ouvertes*, 2010.
- [15] A. Bueno-Orovio, E. M. Cherry, and F. H. Fenton. Minimal model for human action potentials in tissue. *Journal of Theoretical Biology*, 253:544–560, 2008.
- [16] A. Buffa, J. Rivas, G. Sangalli, and R. Vázquez. Isogeometric discrete differential forms in three dimensions. *SIAM Journal on Numerical Analysis*, 49:814–844, 2011.
- [17] A. Buffa and G. Sangalli. *IsoGeometric Analysis: A New Paradigm in the Numerical Approximation of PDEs*. Springer, 2016.
- [18] A. Buffa, R. Vázquez, G. Sangalli, and L. Beirao da Veiga. Approximation estimates for isogeometric spaces in multipatch geometries. *Numerical Methods for Partial Differential Equations*, 31:422–438, 2014.
- [19] V. Calo, Q. Deng, and V. Puzyrev. Dispersion optimized quadratures for isogeometric analysis. *Journal of Computational and Applied Mathematics*, 355:283–300, 2019.
- [20] L. A. Charawi. Isogeometric overlapping Schwarz preconditioners for the bidomain reaction–diffusion system. *Computer Methods in Applied Mechanics and Engineering*, 319:472–490, 2017.
- [21] P. Colli Franzone and L. F. Pavarino. A parallel solver for reaction–diffusion systems in computational electrocardiology. *Mathematical Models and Methods in Applied Sciences*, 14:883–911, 2004.
- [22] P. Colli Franzone, L. F. Pavarino, and S. Scacchi. *Mathematical Cardiac Electrophysiology*. Springer, 2014.
- [23] P. Colli Franzone, L. F. Pavarino, and S. Scacchi. A numerical study of scalable cardiac electro-mechanical solvers on HPC architectures. *Frontiers in Physiology*, 9:268, 2018.
- [24] P. Colli Franzone, L. F. Pavarino, and B. Taccardi. Simulating patterns of excitation, repolarization and action potential duration with cardiac Bidomain and Monodomain models. *Mathematical Biosciences*, 197:35–66, 2005.
- [25] A. Collin, G. Sangalli, and T. Takacs. Analysis-suitable  $G^1$  multi-patch parametrizations for  $C^1$  isogeometric spaces. *Computer Aided Geometric Design*, 47:93–113, 2016.
- [26] J. A. Cottrell, T. J. R. Hughes, and Y. Bazilevs. Isogeometric Analysis: CAD, finite elements, NURBS, exact geometry, and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194:4135–4195, 2005.
- [27] J. A. Cottrell, T. J. R. Hughes, and Y. Bazilevs. *Isogeometric Analysis: Towards Integration of CAD and FEA*. Wiley, 2009.
- [28] J. A. Cottrell, T. J. R. Hughes, and A. Reali. Studies of refinement and continuity in isogeometric structural analysis. *Computer Methods in Applied Mechanics and Engineering*, 196:4160–4183, 2007.

- [29] 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:H301–H321, 1998.
- [30] L. Beirao da Veiga, A. Buffa, J. Rivas, and G. Sangalli. Some estimates for h-p-k-refinement in Isogeometric Analysis. *Numerische Mathematik*, 118:271–305, 2010.
- [31] C. de Falco, A. Reali, and R. Vázquez. GeoPDEs: A research tool for Isogeometric Analysis of PDEs. *Advances in Engineering Software (Thomson Reuters)*, 42:1020–1034, 2011.
- [32] L. Dede', C. Jäggli, and A. Quarteroni. Isogeometric numerical dispersion analysis for two-dimensional elastic wave propagation. *Computer Methods in Applied Mechanics and Engineering*, 284:320–348, 2015.
- [33] L. Dede' and A. Quarteroni. Isogeometric analysis for second order partial differential equations on surfaces. *Computer Methods in Applied Mechanics and Engineering*, 284:807–834, 2014.
- [34] D. Deng, H. J. Arevalo, A. Prakosa, D. J. Callans, and N. A. Trayanova. A feasibility study of arrhythmia risk prediction in patients with myocardial infarction and preserved ejection fraction. *Europace*, 18:60–66, 2016.
- [35] D. Deng, A. Prakosa, J. Shade, P. Nikolov, and N. A. Trayanova. Characterizing Conduction Channels in Postinfarction Patients Using a Personalized Virtual Heart. *Biophysical Journal*, 117:2287–2294, 2019.
- [36] R. Doste, D. Soto-Iglesias, G. Bernardino, A. Alcaine, R. Sebastian, S. Giffard-Roisin, M. Sermesant, A. Berruezo, D. Sanchez-Quintana, and O. Camara. A rule-based method to model myocardial fiber orientation in cardiac biventricular geometries with outflow tracts. *International Journal for Numerical Methods in Biomedical Engineering*, page 3185, 2019.
- [37] P. Gervasio, L. Dede', O. Chanon, and A. Quarteroni. Comparing Isogeometric Analysis and Spectral Element Methods: accuracy and spectral properties. *Journal of Scientific Computing*, 83, 2020.
- [38] S. Götepkpe and E. Kuhl. Computational modeling of cardiac electrophysiology: A novel finite element approach. *International Journal for Numerical Methods in Engineering*, 72:156–178, 2009.
- [39] V. Gurev, T. Lee, J. Constantino, H. Arevalo, and N. A. Trayanova. Models of cardiac electromechanics based on individual hearts imaging data. *Biomechanics and Modeling in Mechanobiology*, 10:295–306, 2011.
- [40] R. A. Harrington, J. Narula, and Z. J. Eapen. *Hurst's the Heart*. MacGraw-Hill, 2011.
- [41] C. Heinrich, B. Simeon, and E. Pilgerstofer. Swept Volume Parametrization for Isogeometric Analysis. *Conference: Mathematics of Surfaces XIII*, 2009.
- [42] A. L. Hodgkin and A. F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *Journal of Physiology*, 117:500–544, 1952.
- [43] M. Hsu, C. Wang, A. J. Herrema, D. Schillinger, A. Ghoshal, and Y. Bazilevs. An interactive geometry modeling and parametric design platform for isogeometric analysis. *Computers and Mathematics with Applications*, 70:1481–1500, 2015.

- [44] T. J. R. Hughes, A. Reali, and G. Sangalli. Duality and unified analysis of discrete approximations in structural dynamics and wave propagation: Comparison of p-method finite elements with k-method NURBS. *Computer Methods in Applied Mechanics and Engineering*, 197:4104–4124, 2008.
- [45] T. J. R. Hughes and G. Sangalli. Mathematics of Isogeometric Analysis: A Conspectus. In E. Stein, R. Borst, and T. J. R. Hughes, editors, *Encyclopedia of Computational Mechanics Second Edition*, pages 1–40. American Cancer Society, 2017.
- [46] M. Kapl, G. Sangalli, and T. Takacs. An isogeometric  $C^1$  subspace on unstructured multipatch planar domains. *Computer Aided Geometric Design*, 69:55–75, 2019.
- [47] S. Kargaran, B. Jüttler, S. Kleiss, A. Mantzaflaris, and T. Takacs. Overlapping multipatch structures in isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 356:325–353, 2019.
- [48] A. M. Katz. *Physiology of the Heart*. Wolters Kluwer, 2011.
- [49] R. Klabunde. *Cardiovascular Physiology Concepts*. Lippincott Williams & Wilkins, 2011.
- [50] S. Kleiss, C. Pechstein, B. Jüttler, and S. Tomar. IETI - Isogeometric Tearing and Interconnecting. *Computer Methods in Applied Mechanics and Engineering*, 247-248:201–215, 2012.
- [51] S. Krishnamoorthi, M. Sarkar, and W. S. Klug. Numerical quadrature and operator splitting in finite element methods for cardiac electrophysiology. *International Journal for Numerical Methods in Biomedical Engineering*, 29:1243–1266, 2013.
- [52] U. Langer, A. Mantzaflaris, S. E. Moore, and I. Touloupoulos. Multipatch Discontinuous Galerkin Isogeometric Analysis. *Isogeometric Analysis and Applications 2014. Lecture Notes in Computational Science and Engineering*, 107:1–32, 2015.
- [53] C. H. Luo and Y. Rudy. A model of the ventricular cardiac action potential. Depolarization, repolarization, and their interaction. *Circulation Research*, 68:1501–1526, 1991.
- [54] W. Martin, E. Cohen, R. Fish, and P. Shirley. Practical ray tracing of trimmed NURBS surfaces. *Journal of Graphics Tools*, 5:27–52, 2000.
- [55] A. Nagy and D. Benson. On the numerical integration of trimmed isogeometric elements. *Computer Methods in Applied Mechanics and Engineering*, 284:165–185, 2015.
- [56] V. P. Nguyen, C. Anitescu, S. P. A. Bordas, and T. Rabczuk. Isogeometric Analysis: An overview and computer implementation aspects. *Mathematics and Computers in Simulation*, 117:89–116, 2015.
- [57] A. S. Patelli, L. Dede, T. Lassila, A. Bartzzaghi, and A. Quarteroni. Isogeometric approximation of cardiac electrophysiology models on surfaces: An accuracy study with application to the human left atrium. *Computer Methods in Applied Mechanics and Engineering*, 317:248–273, 2017.
- [58] P. Pathmanathan, G. Mirams, J. Southern, and J. Whiteley. The significant effect of the choice of ionic current integration method in cardiac electro-physiological simulations. *International Journal for Numerical Methods in Biomedical Engineering*, 27:1751–1770, 2011.

- [59] L. Pegolotti, L. Dede', and A. Quarteroni. Isogeometric Analysis of the electrophysiology in the human heart: Numerical simulation of the bidomain equations on the atria. *Computer Methods in Applied Mechanics and Engineering*, 343:52–73, 2019.
- [60] L. A. Piegl and W. Tiller. *The NURBS Book*. Springer, 1997.
- [61] R. Piersanti, P. C. Africa, M. Fedele, C. Vergara, L. Dede', A. F. Corno, and A. Quarteroni. Modelling cardiac muscle fibers in ventricular and atrial electrophysiology simulations. *MOX Report*, 29, 2020.
- [62] G. Plank, L. Zhou, J. L. Greenstein, S. Cortassa, R. L. Winslow, B. O'Rourke, and N. A. Trayanova. From mitochondrial ion channels to arrhythmias in the heart: computational techniques to bridge the spatio-temporal scales. *Philosophical Transactions of The Royal Society*, 366:3381–3409, 2008.
- [63] K. Qin, M. Gong, and G. Tong. Fast ray tracing NURBS surfaces. *Journal of Computer Science and Technology*, 11:17–29, 1996.
- [64] A. Quarteroni. *Numerical Models for Differential Problems*. Springer, 2017.
- [65] A. Quarteroni, L. Dede', A. Manzoni, and C. Vergara. *Mathematical Modeling of the Human Cardiovascular System: Data, Numerical Approximation, Clinical Applications*. Cambridge University Press, 2019.
- [66] A. Quarteroni, T. Lassila, S. Rossi, and R. Ruiz-Baier. Integrated heart-coupling multiscale and multiphysics models for the simulation of the cardiac function. *Computer Methods in Applied Mechanics and Engineering*, 314:345–407, 2017.
- [67] A. Quarteroni, R. Sacco, F. Saleri, and P. Gervasio. *Matematica Numerica*. Springer, 2014.
- [68] J. M. Rogers and A. D. McCulloch. A collocation-Galerkin finite element model of cardiac action potential propagation. *IEEE Transactions on Biomedical Engineering*, 41:743–757, 1994.
- [69] S. Rossi, T. Lassila, R. Ruiz-Baier, A. Sequeira, and A. Quarteroni. Thermodynamically consistent orthotropic activation model capturing ventricular systolic wall thickening in cardiac electromechanics. *European Journal of Mechanics - A/Solids*, 48:129–142, 2014.
- [70] Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2003.
- [71] S. Sakamoto, T. Nitta, Y. Ishii, Y. Miyagi, H. Ohmori, and K. Shimizu. Interatrial electrical connections: the precise location and preferential conduction. *Journal of Cardiovascular Electrophysiology*, 16:1077–1086, 2005.
- [72] A. Santiago, J. Aguado-Sierra, M. Zavala-Ake, R. Doste-Beltran, S. Gomez, R. Aris, J. Cajas, E. Casoni, and M. Vázquez. Fully coupled fluid-electro-mechanical model of the human heart for supercomputers. *International Journal for Numerical Methods in Biomedical Engineering*, 34:e3140, 2018.
- [73] A. M. Scher. The sequence of ventricular excitation. *The American Journal of Cardiology*, pages 287–293, 1964.
- [74] J. Sundnes, G. T. Lines, X. Cai, B. F. Nielsen, K. Mardal, and A. Tveito. *Computing the Electrical Activity in the Heart*, volume 1. Springer, 2007.

- [75] E. Vigmond, R. Weber dos Santos, A. J. Prassl, M. Deo, and G. Plank. Solvers for the cardiac bidomain equations. *Progress in Biophysics & Molecular Biology*, 96:3–18, 2008.
- [76] E. Vigmond, F. Vadakkumpadan, V. Gurev, H. Arevalo, M. Deo, G. Plank, and N. A. Trayanova. Towards predictive modeling of the electrophysiology of the heart. *Experimental Physiology*, 94:563–577, 2009.
- [77] G. Xu, B. Mourrain, R. Duvigneau, and A. Galligo. Analysis-suitable volume parametrization of multi-block computational domain in isogeometric applications. *Computer-Aided Design*, 45:395–404, 2013.
- [78] Y. Zhang, Y. Bazilevs, S. Goswami, C. L. Bajaj, and T. J. R. Hughes. Patient-specific vascular NURBS modeling for isogeometric analysis of blood flow. *Computer Methods in Applied Mechanics and Engineering*, 196:2943–2959, 2007.

## MOX Technical Reports, last issues

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

- 44/2020** Masci, C.; Ieva, F.; Paganoni A.M.  
*EM algorithm for semiparametric multinomial mixed-effects models*
- 45/2020** Gatti, F.; Menafoglio, A.; Togni, N.; Bonaventura, L.; Brambilla, D.; Papini, M; Longoni, L.  
*A novel downscaling procedure for compositional data in the Aitchison geometry with application to soil texture data*
- 42/2020** Miglio, E.; Parolini, N.; Quarteroni, A.; Verani, M.; Zonca, S.  
*A spatio-temporal model with multi-city mobility for COVID-19 epidemic*
- 43/2020** Quarteroni, A.; Vergara, C.  
*Modeling the effect of COVID-19 disease on the cardiac function: A computational study*
- 39/2020** Martinolli, M.; Biasetti, J.; Zonca, S.; Polverelli, L.; Vergara, C.  
*Extended Finite Element Method for Fluid-Structure Interaction in Wave Membrane Blood Pumps*
- 40/2020** Fresca, S.; Manzoni, A.; Dedè, L.; Quarteroni, A.  
*Deep learning-based reduced order models in cardiac electrophysiology*
- 41/2020** Cannistrà, M.; Masci, C.; Ieva, F.; Agasisti, T.; Paganoni, A.M.  
*Not the magic algorithm: modelling and early-predicting students dropout through machine learning and multilevel approach*
- 38/2020** Sollini, M.; Kirienko, M.; Cavinato, L.; Ricci, F.; Biroli, M.; Ieva, F.; Calderoni, L.; Tabacchi, D.  
*Methodological framework for radiomics applications in Hodgkin's lymphoma*
- 34/2020** Antonietti, P.F.; Mazzieri, I.; Nati Poltri, S.  
*A high-order discontinuous Galerkin method for the poro-elasto-acoustic problem on polygonal and polyhedral grids*
- 35/2020** Morbiducci, U.; Mazzi, V.; Domanin, M.; De Nisco, G.; Vergara, C.; Steinman, D.A.; Gallo, D.  
*Wall shear stress topological skeleton independently predicts long-term restenosis after carotid bifurcation endarterectomy*