

POLITECNICO DI MILANO

COURSE OF NUMERICAL ANALYSIS FOR PARTIAL DIFFERENTIAL  
EQUATIONS

---

DISCONTINUOUS GALERKIN METHOD FOR THE BIDOMAIN  
PROBLEM OF CARDIAC ELECTROPHYSIOLOGY

---

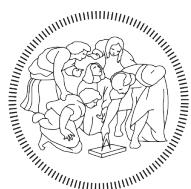
*Authors:*

Viola BRENZONE  
Federico LANTERI  
Marco RAPETTI

*Supervisors:*

prof. C. VERGARA  
prof. P. F. ANTONIETTI

Academic Year 2021 / 2022



**POLITECNICO**  
**MILANO 1863**

# CONTENTS

1	Introduction	2
1.1	Physical problem	2
1.2	Mathematical modeling	3
1.2.1	Conductivity Tensor and Fiber Direction	6
1.3	Starting Point and Goals	7
2	Numerical problem and formulation	8
2.1	Semi-discrete Discontinuous Galerkin weak formulation	8
2.2	Semi-discrete Discontinuous Galerkin algebraic formulation	9
2.3	Dubiner basis	10
2.4	Temporal discretization	11
2.5	Lumping	13
3	Numerical Results	15
3.1	Single Cell Simulations	15
3.2	Error analysis	18
3.2.1	Dominant reaction case	18
3.3	Realistic Simulations	22
3.3.1	Conductivity values	22
3.3.2	1 <sup>st</sup> test-case ( <i>overflow vs underflow</i> case)	23
3.3.3	2 <sup>nd</sup> test-case: impulse from the angle	26
3.3.4	3 <sup>rd</sup> test-case: realistic fibers directions	27
3.3.5	4 <sup>th</sup> test-case: high value of $\alpha$ , stability and lumping	29
4	Conclusions	31
A	The code	32
A.1	Single cell simulations	32
A.2	Fibers Directions	32
A.3	Lumping	33

# 1

## INTRODUCTION

The aim of this project is to study and implement a suitable numerical scheme for the approximate solutions of the Bidomain Problem, a well-known system of non-linear partial differential equations that has been developed in the context of the modeling of electrophysiology of the myocardial muscle. This work is basically the continuation of a three-year-long study carried out by four past course projects (the last ones are [1], [3], [2]). In particular, the broad goal of this project is to check and improve the results obtained by Botta and Calafà in [1].

We provide in what follows a brief introduction to the physical problem and its modelling through the Bidomain equations. For a more complete explanation, we instead refer to [4], [5], [6].

### 1.1 PHYSICAL PROBLEM

The mechanical contractions of the myocardium muscle have their origin in the electrical activation of the cardiac cells, which are excitable. At each heartbeat, myocytes are activated and deactivated following a characteristic electrical cycle.

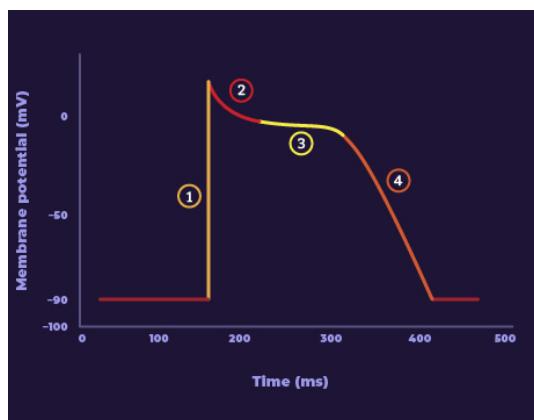


Figure 1.1: Cardiac Action Potential

The excitation of a cardiac cell causes a rapid variation of its potential difference across the cell membrane, the so-called *transmembrane potential*. If the stimulus is below a certain threshold value, the transmembrane potential quickly returns to its resting value after the stimulus ends. On the other hand, if the stimulus is above threshold, from an initial status where the cell is at rest (-90 mV), its potential increases rapidly in circa 2 ms and reaches the value of +20 mV (phase 1): the cell is now activated. Later, a plateau near 0 mV is observed (phase 2-3) and then a slow repolarization to the initial potential (phase 4). This event is called an *action potential* (Figure 1.1).

From a microscopical point of view, it's possible to study the dynamics acting in each single cell as a consequence of the passage of chemical ions through specific channels, e.g.  $Ca^{2+}$ ,  $Na^+$ ,  $K^+$ ,  $Cl^-$ . In particular:

- during the **Phase 1**, myocytes undergo a rapid depolarization due to the opening of the fast  $Na^+$  channels. This causes a rapid increase in the membrane conductance and thus a rapid influx of  $Na^+$  ions into the cell, the  $I_{Na}$  current;
- **Phase 2** occurs with the inactivation of the fast  $Na^+$  channels. The transient net outward current causing the small downward deflection of the action potential is due to  $K^+$  and  $Cl^-$  ions;
- **Phase 3** is the plateau phase sustained by a balance between inward movement of  $Ca^{2+}$  and outward movement of potassium ions  $K^+$ ;
- **Phase 4** is the “rapid repolarization” phase of the action potential, when the  $Ca^{2+}$  channels close while  $K^+$  channels are still open. This ensures a net outward current, corresponding to negative change in membrane potential. This net outward, positive current causes the cell to repolarize.

From a macroscopical point of view, instead, one can describe this phenomenon as a continuous electrical diffusion over the entire cardiac surface. Indeed, although the dynamics of the cardiac action potential are driven at the level of an individual cell by voltage-gated transmembrane currents, the spatial spread of excitation throughout the tissue is driven by the essentially passive flow of intracellular, intercellular, and extracellular currents. Cardiac conduction can therefore be described as a reaction-diffusion process.

Even though cardiac arrhythmias are among the major causes of death and disability, a noninvasive imaging technique yielding an accurate and reliable diagnosis of the electrophysiological state of the heart is not yet available. In the past few decades, experimental electrophysiology has been increasingly supported by the mathematical and numerical models of computational electrocardiology. The formulation of models at both cellular and tissue levels provide essential tools in order to integrate the increasing knowledge of the bioelectrochemical phenomena occurring through cardiac cellular membranes.

## 1.2 MATHEMATICAL MODELING

The single membrane cell can be modelled as a capacitor separating charges that accumulate at its intracellular and extracellular surfaces. Moreover ionic currents cross the membrane through channels which open and close during excitation. A suitable model can therefore be expressed via the simple electric circuit in figure 1.2: for each cell, it consists of a capacitor and a series of resistances, one for each ionic current (here only sodium and potassium ionic channels are depicted). In the intracellular region, two adjacent cells are connected by a resistance representing a gap junction. However, the latter is not explicitly modelled at the macroscopic scales: instead its effect is hidden in the conductivity tensor.

By applying to this circuit general electromagnetism laws, the so-called Bidomain equations are obtained. These equations are defined on the heart wall, which is made up of three layers: the internal thin *endocardium*, the thick muscular *myocardium* and the external thin *epicardium* 1.3.

In addition to initial and boundary conditions for these equations, a suitable model for the ionic currents  $I_{ion}$  must be prescribed in order to close the system. In particular, a scaled version of the Fitzhugh-Nagumo has been chosen as ionic model:

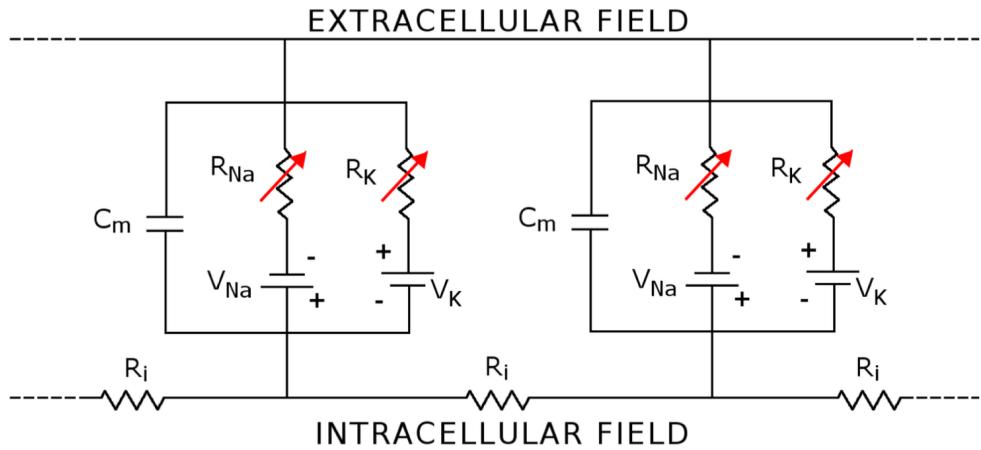


Figure 1.2: Electrical circuit for the sequence of two cardiac cells

**Problem 1.1 (Bidomain model)** Find  $\varphi_i$  and  $\varphi_e$  such that:

$$\begin{cases} \chi_m C_m \frac{\partial V_m}{\partial t} - \nabla \cdot (\Sigma_i \nabla \varphi_i) + \chi_m I_{ion}(V_m, w) = I_i^{ext}, & \text{in } \Omega_{mus} \times (0, T], \\ -\chi_m C_m \frac{\partial V_m}{\partial t} - \nabla \cdot (\Sigma_e \nabla \varphi_e) + \chi_m I_{ion}(V_m, w) = -I_e^{ext}, & \text{in } \Omega_{mus} \times (0, T], \\ I_{ion}(V_m, w) = \alpha k V_m (V_m - a)(V_m - 1) + \alpha b w, & \text{in } \Omega_{mus} \times (0, T], \\ \frac{\partial w}{\partial t} = \alpha \varepsilon (V_m - \gamma w), & \text{in } \Omega_{mus} \times (0, T], \\ \Sigma_i \nabla \varphi_i \cdot \mathbf{n} = 0, & \text{on } \partial \Omega_{mus} \times (0, T], \\ \Sigma_e \nabla \varphi_e \cdot \mathbf{n} = 0 & \text{on } \partial \Omega_{mus} \times (0, T], \end{cases}$$

where:

- $\varphi_i, \varphi_e$  are the intracellular and extracellular potentials (*unknowns*),
- $V_m = \varphi_i - \varphi_e$  is the transmembrane potential,
- $w$  is the gating variable (*unknown*) which represents the percentage of open channels per unit area of the membrane,
- $\mathbf{n}$  is the outward normal vector,
- $\chi_m, C_m$  are known positive constants:  $\chi_m$  represents the surface area-to-volume ratio of cardiomyocytes,  $C_m$  represents the transmembrane capacitance per unit area,
- $\Sigma_i, \Sigma_e$  are known positive definite intracellular and extracellular conductivity tensors,
- $I_i^{ext}, I_e^{ext}$  are the applied currents,
- $I_{ion}$  is the Ionic Current,
- $\Omega_{mus}$  is the cardiac domain (endocardium + myocardium + epicardium 1.3),
- $k, a, \varepsilon, b$  are known constants,
- $\alpha$  is the scaling factor for the Fitzhugh-Nagumo model.

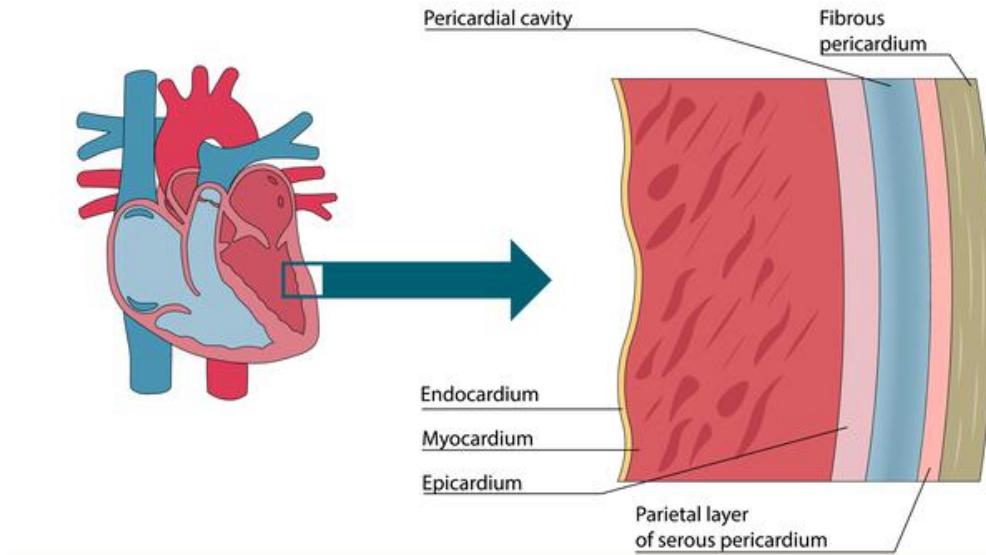


Figure 1.3: Domain  $\Omega_{mus} = \text{endocardium} + \text{myocardium} + \text{epicardium}$

The novelty with respect to the classic Fitzhugh-Nagumo model, which is used in the previous works (for example in [1]), is the introduction of the scaling parameter  $\alpha$ , a positive constant. The advantage of this model is that, changing  $\alpha$  we can control the excitation rate and decay constants of the ionic model: if  $\alpha \leq 1$  the dynamic gets slower and slower for decreasing values of  $\alpha$ , while, if  $\alpha \geq 1$ , a more rapid dynamic of our system is expected as the value of  $\alpha$  increases.

From now on, the system of equations given by 1.1 will be the reference analytical problem for the development of the forthcoming numerical schemes.

### 1.2.1 Conductivity Tensor and Fiber Direction

The ventricular myocardium exhibits a complex three-dimensional spatial organization where myocytes are connected mostly end-to-end to form cardiac fibers with varying orientation. This ventricular fiber orientation rotates smoothly between endocardium and epicardium. Cardiac fibers are also known to have an additional laminar organization consisting of muscle sheets, typically 4-6 myocytes thick, running radially from epicardium to endocardium, with surface orientation varying throughout the ventricles (figure 1.4). These sheets are separated by gaps called cleavage planes and by layers of connective tissue [5].

Anisotropy refers to the property of a material or system to have different physical properties in different directions, and in our case, in order to run realistic simulation, we need to consider also that the heart is composed of many muscle fibers that are oriented in different directions. Indeed, the fiber orientation determines how the electric signal propagates within the muscle. As a matter of fact, in the literature it's described how the fibers' orientation determine the anisotropic repolarization's spread ([15]).

For this reason, in numerical heart electrophysiology, a critical issue is that of modeling the myocardial fibers direction that characterizes the cardiac tissue, and this motivates the need to accurately include fiber orientations in order to obtain physically meaningful results [4], [7], [16].

In our formulation, the conductivity tensors are defined for intracellular ( $\Sigma_i$ ) and extracellular ( $\Sigma_e$ ) domains. Due to the anisotropy of the cardiac tissue induced by the presence of fibres, each conductivity tensor is in general expressed in terms of two scalar quantities  $\sigma_l$  and  $\sigma_t$  representing the conductivities along the fibre direction  $a_l(x)$  and the direction  $a_t(x)$  orthogonal to  $a_l(x)$ , that is:

$$\Sigma = \sigma_l \mathbf{a}_l \cdot \mathbf{a}_l^T + \sigma_t \mathbf{a}_t \cdot \mathbf{a}_t^T$$

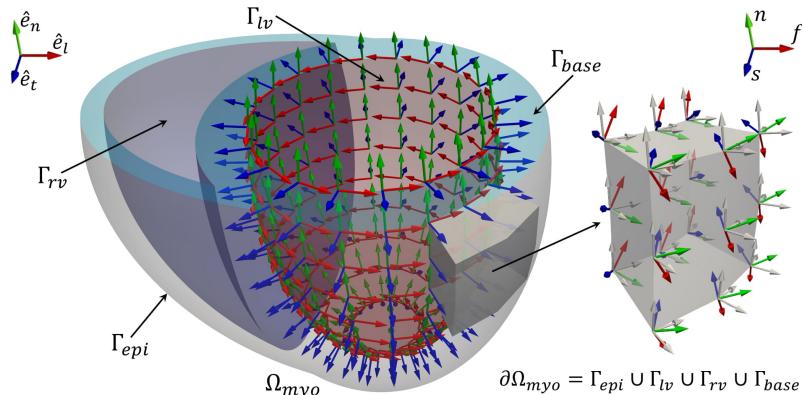


Figure 1.4: Organization of cardiac muscle tissue in sheets and fibers

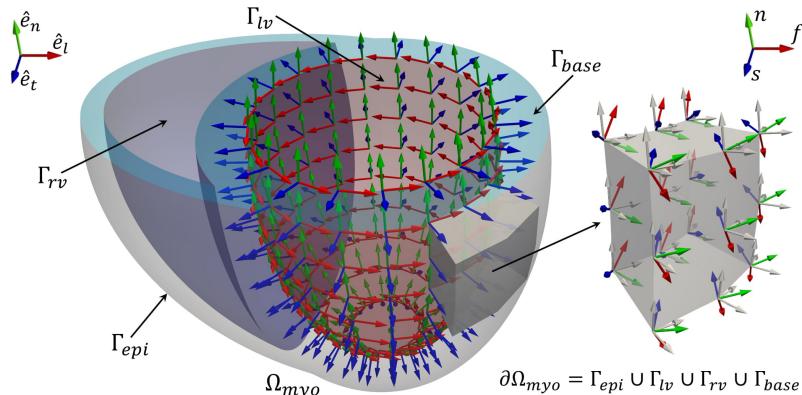


Figure 1.5: Fibers directions

It is important to note that the exact fiber orientations and the conductivity values are difficult to measure, so the fiber orientations used in numerical models are approximations that can vary along the different experiments. Cardiac propagation characteristics such as anisotropy ratio and conduction velocities are often determined experimentally from epicardial measurements: for example, magnetic resonance techniques can be used to measure fiber orientation as the principal eigenvector of the diffusion tensor. Alternatively, fibers' orientations can be generated through numerical strategies. For example in figure 1.5, taken from [7], fibers are obtained using a rule-based method known as Laplace-Dirichlet-Rule-Based-Methods, which rely on the solution of a Laplace boundary-value problem: the directions, which are obtained by considering the solution of a Laplace problem, are then properly rotated in order to match the histological observations.

To better understand how the propagation of the electrical potential spreads into the ventricles, we observe that cardiomyocytes are of cylindrical type. This allows us to define the fibre direction of the cell, resulting in a macroscopic fibre direction intended as the average cell orientation in a sufficiently small control volume.

### 1.3 STARTING POINT AND GOALS

As we already introduced, our project aims to continue and improve the work of a previous project. Botta and Calafà obtained satisfactory results with their code, but they were not able to capture the repolarization phase described in section 1.1 and this did not allow them to obtain realistic results [1]. Thus, the main purposes of the project are:

- Understand why repolarization was not captured in previous work and solve this issue;
- Perform pseudo-realistic simulations, focusing also on a more realistic treatment of the fiber directions;
- Study the possibility to use Mass-Lumping technique to stabilize reaction dominated equation.

# 2

---

## NUMERICAL PROBLEM AND FORMULATION

Because of the fast upstroke of the action potential 1.1, a numerically robust calculation of the propagation of the wave across the tissue is well known to be computationally challenging. This rapid increase of the transmembrane potential in one cell over a few milliseconds results also in a steep wave front in space, requiring high resolution temporal and spatial discretizations: for these reasons standard methods are prone to not correctly capture the wave propagation unless the grid is very refined.

The original intuition for the previous works [1], [3], [2], was the great suitability of discontinuous methods to solve this kind of problems where solutions are represented by steep and fast waves: Discontinuous Galerkin method is considered in order to assess the hypothesis that a reduced number of degree of freedom are needed to have a good accuracy with respect to the standard Finite Elements case. Moreover DG method allow more ease in the mesh choice, since it is suitable for non conformal meshes.

### 2.1 SEMI-DISCRETE DISCONTINUOUS GALERKIN WEAK FORMULATION

Introducing a triangulation  $\mathcal{T}_h$  over  $\Omega$ , where  $F_h = F_h^I \cup F_h^B$  are the set of the faces of the partition, where  $F_h^I$  are the internal faces and  $F_h^B$  are the boundary ones. Let the DG space be defined as  $V_h^P = \{v_h \in L^2(\Omega) \mid v_h|_K \in \mathbb{P}^p(K) \forall K \in \mathcal{T}_h\}$ , where  $p$  is the degree of the piecewise continuous polynomial.

**Problem 2.1** (*DG weak formulation*). For any  $t \in [0, T]$  find  $\varphi_h(t) = \begin{bmatrix} \varphi_i^h(t) \\ \varphi_e^h(t) \end{bmatrix} \in [V_h^P]^2$ ,  $V_m^h(t) = \varphi_i^h(t) - \varphi_e^h(t)$  and  $w_h(t) \in V_h^P$  such that:

$$\left\{ \begin{array}{l} \sum_{K \in \mathcal{T}_h} \int \chi_m C_m \frac{\partial V_m^h}{\partial t} v_h dw + a_i(\varphi_i^h, v_h) + \sum_{K \in \mathcal{T}_h} \int \chi_m \alpha k (V_m^h - 1)(V_m^h - a) V_m^h v_h dw + \\ - \sum_{K \in \mathcal{T}_h} \int \chi_m \alpha b w_h v_h dw = (I_i^{ext}, v_h) \quad \forall v_h \in V_h^P \\ \\ - \sum_{K \in \mathcal{T}_h} \int \chi_m C_m \frac{\partial V_m^h}{\partial t} v_h dw + a_e(\varphi_e^h, v_h) - \sum_{K \in \mathcal{T}_h} \int \chi_m \alpha k (V_m^h - 1)(V_m^h - a) V_m^h v_h dw + \\ - \sum_{K \in \mathcal{T}_h} \int \chi_m \alpha b w_h v_h dw = (-I_e^{ext}, v_h) \quad \forall v_h \in V_h^P \\ \\ \sum_{K \in \mathcal{T}_h} \int \frac{\partial w_h}{\partial t} v_h dw = \sum_{K \in \mathcal{T}_h} \int \alpha \varepsilon (V_m^h - \gamma w_h) v_h dw \quad \forall v_h \in V_h^P \end{array} \right.$$

where:

- $\nabla_h$  is the element pointwise gradient
- $a_j(\varphi_j^h, v_h) = \sum_{K \in \mathcal{T}_h} \int (\Sigma_j \nabla_h \varphi_j^h) \cdot \nabla_h v_h dw - \sum_{F \in F_h^I} \int \{\{\Sigma_j \nabla_h \varphi_j^h\}\} \cdot [v_h] d\sigma +$

- $$-\delta \sum_{F \in F_h^I} \int_F \{\{\Sigma_j \nabla_h v_h\}\} \cdot [\varphi_j^h] d\sigma + \sum_{F \in F_h^I} \int_F \Gamma [\varphi_j^h] \cdot [v_h] d\sigma \quad \text{with } j = i, e$$
- $(I_i^{ext}, v_h) = \sum_{K \in \mathcal{T}_h} \int_K I_i^{ext} v_h dw$
  - $(-I_e^{ext}, v_h) = - \sum_{K \in \mathcal{T}_h} \int_K I_e^{ext} v_h dw$

According to the choice of the parameter  $\delta$ , we can use 3 different penalty methods:

- $\delta = 1$ : Symmetric Interior Penalty Method (SIP)
- $\delta = 0$ : Incomplete Interior Penalty Method (IIP)
- $\delta = -1$ : Non-Symmetric Interior Penalty Method (NIP)

## 2.2 SEMI-DISCRETE DISCONTINUOUS GALERKIN ALGEBRAIC FORMULATION

Taking  $\{\psi_j\}_{j=1}^{N_h}$  as a basis of  $V_h^p$ , so that we can write:

$$\begin{aligned} \varphi_h(t) &= \begin{bmatrix} \varphi_i^h(t) \\ \varphi_e^h(t) \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^{N_h} \varphi_{i,j}(t) \psi_j \\ \sum_{j=1}^{N_h} \varphi_{e,j}(t) \psi_j \end{bmatrix} \\ w_h(t) &= \sum_{j=1}^{N_h} w_j(t) \psi_j \\ V_m^h(t) &= \sum_{j=1}^{N_h} V_{m,j}(t) \psi_j = \sum_{j=1}^{N_h} (\varphi_{i,j}(t) - \varphi_{e,j}(t)) \psi_j \end{aligned}$$

Where  $\varphi_{i,j}$ ,  $\varphi_{e,j}$  and  $w_j \in \mathbb{R}$  are the unknowns  $\forall j = 1, \dots, N_h$ .

In order to define the algebraic formulation we define the following matrices and vectors (with  $l = i, e$ ):

$$\begin{cases} (V_l)_{i,j} = \sum_{K \in \mathcal{T}_h} \int_K \nabla_h \psi_j \cdot \Sigma_l \nabla_h \psi_i \\ (I_l^T)_{i,j} = \sum_{F \in F_h^I} \int_F \{\{\Sigma_l \nabla_h \psi_j\}\} \cdot [\psi_i] \\ (I_l)_{i,j} = \sum_{F \in F_h^I} \int_F [\psi_j] \cdot \{\{\Sigma_l \nabla_h \psi_i\}\} \quad \forall i, j = 1, \dots, N_h \\ (S_l)_{i,j} = \sum_{F \in F_h^I} \int_F \Gamma_l [\psi_j] \cdot [\psi_i] \end{cases} \Rightarrow A_l = (V_l - I_l^T - \vartheta I_l + S_l)$$

$$M_{r,j} = \sum_{K \in \mathcal{T}_h} \int_K \psi_j \psi_r \quad \text{Mass matrix}$$

$$V_m^h(t) = \varphi_i^h(t) - \varphi_e^h(t) \quad \text{Vector of the coefficients of the solution}$$

$$C(V_m^h(t))_{r,j} = \sum_{K \in \mathcal{T}_h} \int_K \chi_m \alpha k (V_m^h(t)_j - 1) (V_m^h(t)_j - a) \psi_j \psi_r \quad \text{Non-Linear matrix}$$

$$F_{i,j}^h = \sum_{K \in \mathcal{T}_h} \int_K I_i^{ext} \psi_j \quad \text{Intra-cellular forcing term}$$

$$F_{e,j}^h = \sum_{K \in \mathcal{T}_h} \int_K I_e^{ext} \psi_j \quad \text{Extra-cellular forcing term}$$

Thanks to these definitions we are able to formulate the algebraic formulation of our problem as follows:

**Problem 2.2 (DG algebraic formulation).** Find  $\varphi_h(t) = \begin{bmatrix} \varphi_i^h(t) \\ \varphi_e^h(t) \end{bmatrix} \in \mathbb{R}^{2N_h}$  and  $w_h(t) \in \mathbb{R}^{N_h}$  for any  $t \in (0, T]$  such that:

$$\left\{ \begin{array}{l} \chi_m C_m \begin{bmatrix} M & -M \\ -M & M \end{bmatrix} \begin{bmatrix} \dot{\varphi}_i^h(t) \\ \dot{\varphi}_e^h(t) \end{bmatrix} + \begin{bmatrix} A_i & 0 \\ 0 & A_e \end{bmatrix} \begin{bmatrix} \varphi_i^h(t) \\ \varphi_e^h(t) \end{bmatrix} + \begin{bmatrix} C(V_m^h) & -C(V_m^h) \\ -C(V_m^h) & C(V_m^h) \end{bmatrix} \begin{bmatrix} \varphi_i^h(t) \\ \varphi_e^h(t) \end{bmatrix} + \\ + \chi_m \alpha b \begin{bmatrix} M & 0 \\ 0 & -M \end{bmatrix} \begin{bmatrix} w_h(t) \\ w_h(t) \end{bmatrix} = \begin{bmatrix} F_i^h \\ F_e^h \end{bmatrix} \\ M \dot{w}_h(t) = \alpha \varepsilon M (V_m^h(t) - \gamma w_h(t)) \end{array} \right.$$

## 2.3 DUBINER BASIS

So far, we have described a general semi-discrete discontinuous formulation without examining which basis to use to generate the  $V_h^P$  space. Usually, the common choice consists in the classical hat functions from FEM. However, the very novelty of the previous works is the adoption of a new kind of basis, completely different from the previous and commonly known as *Dubiner basis*, which is well suited to high-order approximations. For a more complete treatment of this topic we refer to [1] and [12].

The peculiarity of this family of functions is that it consists of orthogonal polynomials defined on the reference triangle  $\hat{K} = \{(\zeta, \eta) \mid \zeta, \eta \geq 0, \zeta + \eta \leq 1\}$  and not on the reference square  $\hat{Q} = \{(a, b) \mid -1 \leq a \leq 1, -1 \leq b \leq 1\}$ . The Dubiner basis correspond to a suitable basis initially defined on the reference square that is later transformed on the reference triangle using a proper transformation.

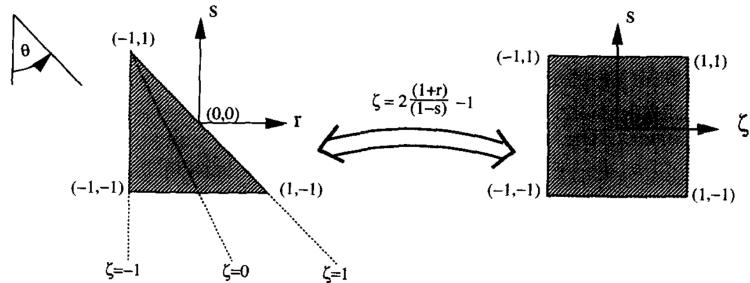


Figure 2.1: Reference square and triangle

The main properties of this choice of basis is that Dubiner basis are orthogonal in  $L^2(\hat{K})$ . Thanks to this property the mass matrix of DG space turns out to be diagonal. In general, the orthogonality property implies some good numerical qualities, not only the diagonalization of the mass matrix. For instance, interesting bounds for the condition number have been proved.

For this reason, the adoption of this choice aims to improve the results, at least from the space discretization side.

## 2.4 TEMPORAL DISCRETIZATION

So far, we have described the space discretization while a temporal discretization is still needed to totally discretize the Bidomain time-dependent problem. Thus, the interval  $(0, T]$  is divided into  $N$  subintervals  $(t^n, t^{n+1}]$  of length  $\Delta t$  such that  $t^n = n\Delta t$ ,  $n = 0, 1, \dots, N - 1$ . We then assume our fully discretized solution  $V_m^n \approx V_m^h(t^n)$ . In the code there are implemented 3 different temporal discretization strategies:

- **The semi-implicit method.** One of the critical points of this problem is that neither fully explicit nor fully implicit time discretization methods can be adopted. Indeed, in the former case we would have a conditionally stable scheme: the condition  $\Delta t \lesssim h^2$  would be necessary to achieve stability, but this would require a very fine temporal discretization; on the contrary, the latter case is unconditionally stable, but it would require to solve at each time step a huge non linear problem. Therefore, for difference reasons, both of these methods are not relevant in this case because they lead to huge and not affordable computational cost.

In order to overcome this issue, the basic idea is to treat most of the terms implicitly while treating the non-linear term semi-implicitly. Since the non-linear term is cubic, the best choice is to treat only one of the  $V_m$  terms implicitly, i.e.:

$$I_{ion}^{n+1} = \alpha k (V_m^n - a) (V_m^n - 1) V_m^{n+1} + w^{n+1}$$

at each time step. Moreover, the gating variable ODE is treated implicitly with the exception of the term  $V_m$

$$M \frac{w^{n+1} - w^n}{\Delta t} = \alpha \varepsilon M (V_m^n - \gamma w^{n+1})$$

Introducing this temporal discretization method, the following algebraic formulation is obtained:

$$\begin{cases} \chi_m C_m \begin{bmatrix} M & -M \\ -M & M \end{bmatrix} \begin{bmatrix} \frac{\varphi_i^{n+1} - \varphi_i^n}{\Delta t} \\ \frac{\varphi_e^{n+1} - \varphi_e^n}{\Delta t} \end{bmatrix} + \begin{bmatrix} A_i & 0 \\ 0 & A_e \end{bmatrix} \begin{bmatrix} \varphi_i^{n+1} \\ \varphi_e^{n+1} \end{bmatrix} + \\ \begin{bmatrix} C(V_m^h) & -C(V_m^h) \\ -C(V_m^h) & C(V_m^h) \end{bmatrix} \begin{bmatrix} \varphi_i^{n+1} \\ \varphi_e^{n+1} \end{bmatrix} + \chi_m \alpha b \begin{bmatrix} M & 0 \\ 0 & -M \end{bmatrix} \begin{bmatrix} w^{n+1} \\ w^{n+1} \end{bmatrix} = \begin{bmatrix} F_i^{n+1} \\ F_e^{n+1} \end{bmatrix} \\ M \frac{w^{n+1} - w^n}{\Delta t} = \alpha \varepsilon M (V_m^n - \gamma w^{n+1}) \end{cases}$$

We know that  $V_m^n = \varphi_i^n - \varphi_e^n$ ; so we obtain that:

$$\begin{cases} \left( \frac{\chi_m C_m}{\Delta t} \begin{bmatrix} M & -M \\ -M & M \end{bmatrix} + \begin{bmatrix} A_i & 0 \\ 0 & A_e \end{bmatrix} + \begin{bmatrix} C(V_m^h) & -C(V_m^h) \\ -C(V_m^h) & C(V_m^*) \end{bmatrix} \right) \begin{bmatrix} \varphi_i^{n+1} \\ \varphi_e^{n+1} \end{bmatrix} = \\ = \begin{bmatrix} F_i^{n+1} \\ F_e^{n+1} \end{bmatrix} - \chi_m \alpha b \begin{bmatrix} M & 0 \\ 0 & -M \end{bmatrix} \begin{bmatrix} w^{n+1} \\ w^{n+1} \end{bmatrix} + \frac{\chi_m C_m}{\Delta t} \begin{bmatrix} M & 0 \\ 0 & -M \end{bmatrix} \begin{bmatrix} V_m^n \\ V_m^n \end{bmatrix}, \\ \left( \frac{1}{\Delta t} + \alpha \varepsilon \gamma \right) M w^{n+1} = \alpha \varepsilon M V_m^n + \frac{M}{\Delta t} w^n \end{cases}$$

Defining:

$$\begin{aligned} - B &= \frac{\chi_m C_m}{\Delta t} \begin{bmatrix} M & -M \\ -M & M \end{bmatrix} + \begin{bmatrix} A_i & 0 \\ 0 & A_e \end{bmatrix} \\ - C_{nl}(V_m^n) &= \begin{bmatrix} C(V_m^n) & -C(V_m^n) \\ -C(V_m^n) & C(V_m^n) \end{bmatrix} \\ - r^{n+1} &= \begin{bmatrix} F_i^{n+1} \\ F_e^{n+1} \end{bmatrix} - \chi_m \alpha b \begin{bmatrix} M & 0 \\ 0 & -M \end{bmatrix} \begin{bmatrix} w^{n+1} \\ w^{n+1} \end{bmatrix} + \frac{\chi_m C_m}{\Delta t} \begin{bmatrix} M & 0 \\ 0 & -M \end{bmatrix} \begin{bmatrix} V_m^n \\ V_m^n \end{bmatrix} \end{aligned}$$

We obtain the final form of our algebraic problem:

**Problem 2.3** (*Semi-implicit discretized system*). Find  $\varphi^{n+1} = \begin{bmatrix} \varphi_i^{n+1} \\ \varphi_e^{n+1} \end{bmatrix}$  and  $w^{n+1} \forall n = 0, \dots, N-1$  such that:

$$\begin{cases} \left( \frac{1}{\Delta t} + \alpha \varepsilon \gamma \right) M w^{n+1} = \alpha \varepsilon M V_m^n + \frac{M}{\Delta t} w^n, \\ (B + C_{nl}(V_m^n)) \varphi^{n+1} = r^{n+1} \end{cases}$$

We expect this scheme to be conditionally stable with a burden of the type  $\Delta t \lesssim h$ , which should not be too hard (and costly) to achieve.

- **The Godunov operator-splitting method.**

The main feature of a general operator-splitting method is the sub-division of the problem into two different problems to be solved sequentially. This is possible and justified when the original functional operator  $L$  is split into 2 different operators such that  $L(u) = L_1(u) + L_2(u)$ . The basic difference from the semi-implicit method is that the first two equations of 1.1 are split into two differential equations for  $V_m$ : a linear PDE without the  $I_{ion}$  term, and a non-linear PDE involving only the  $I_{ion}$  term and no spatial derivatives. For more details we refer to [13] In this case the final form of our algebraic problem is:

**Problem 2.4** (*Godunov operator-splitting discretized system*). Find  $\varphi^{n+1} = \begin{bmatrix} \varphi_i^{n+1} \\ \varphi_e^{n+1} \end{bmatrix}$  and  $w^{n+1} \forall n = 0, \dots, N-1$  such that:

$$\begin{cases} \left( \frac{\chi_m C_m}{\Delta t} \begin{bmatrix} M & -M \\ -M & M \end{bmatrix} + \begin{bmatrix} A_i & 0 \\ 0 & -A_e \end{bmatrix} \right) \begin{bmatrix} \varphi_i^{n+1} \\ \varphi_e^{n+1} \end{bmatrix} = \\ = \begin{bmatrix} F_i^{n+1} \\ F_e^{n+1} \end{bmatrix} - \chi_m \alpha b \begin{bmatrix} M & 0 \\ 0 & M \end{bmatrix} \begin{bmatrix} w^n \\ w^n \end{bmatrix} + \left( \frac{\chi_m C_m}{\Delta t} \begin{bmatrix} M & 0 \\ 0 & M \end{bmatrix} - \begin{bmatrix} C(V_m^n) & 0 \\ 0 & C(V_m^n) \end{bmatrix} \right) \begin{bmatrix} V_m^n \\ V_m^n \end{bmatrix} \\ w^{n+1} = (1 - \alpha \varepsilon \gamma \Delta t) w^n + \alpha \varepsilon \Delta t V_m^n \end{cases}$$

- **The quasi-implicit operator-splitting method.** The aim of a quasi-implicit operator splitting is to treat implicitly all the terms except the cubic one. Even if it cannot be defined as a fully implicit method, the hope is to achieve more stability if compared to the previous Godunov-kind scheme, where the term  $I_{ion}(V_m, w)$  is treated fully explicitly.

In this case the final form of our algebraic problem is:

**Problem 2.5** (*quasi-implicit operator-splitting discretized system*). Find  $\varphi^{n+1} = \begin{bmatrix} \varphi_i^{n+1} \\ \varphi_e^{n+1} \end{bmatrix}$  and  $w^{n+1} \forall n = 0, \dots, N-1$  such that:

$$\begin{cases} \left( \begin{bmatrix} Q_n & -Q_n \\ Q_n & -Q_n \end{bmatrix} + \begin{bmatrix} A_i & 0 \\ 0 & -A_e \end{bmatrix} \right) \begin{bmatrix} \varphi_i^{n+1} \\ \varphi_e^{n+1} \end{bmatrix} = \begin{bmatrix} R_n \\ R_n \end{bmatrix} + \begin{bmatrix} F_i^{n+1} \\ F_e^{n+1} \end{bmatrix} \\ w^{n+1} = \frac{w^n + \alpha \varepsilon \Delta t (\varphi_i^{n+1} - \varphi_e^{n+1})}{1 + \alpha \varepsilon \gamma \Delta t} \end{cases}$$

Where:

$$\begin{aligned} - Q_n &= \frac{\chi_m C_m}{\Delta t} M + C(V_m^n) + \alpha b \frac{\alpha \varepsilon \chi_m \Delta t}{1 + \alpha \varepsilon \gamma \Delta t} M \\ - R_n &= \frac{\chi_m C_m}{\Delta t} M V_m^n - \alpha b \frac{\chi_m}{1 + \alpha \varepsilon \gamma \Delta t} M w^n \end{aligned}$$

## 2.5 LUMPING

In the case of one dimensional reaction-diffusion problems, we can obtain the same result as with finite differences by using linear finite elements, provided that we resort to the so-called mass-lumping technique, thanks to which the mass matrix

$$M = (m_{ij}), m_{ij} = \int_0^1 \psi_j \psi_i dx$$

which is tridiagonal, is approximated using a diagonal matrix  $M_L$ , called condensed or lumped matrix. To this end we use the following trapezoidal quadrature formula on each interval  $(x_i, x_{i+1})$ , for each  $i = 0, \dots, M-1$

$$\int_{x_i}^{x_{i+1}} f(x) dx \approx \frac{h}{2} (f(x_i) + f(x_{i+1}))$$

Thanks to the properties of finite element basis functions, we then find:

$$\begin{aligned} \int_{x_{i-1}}^{x_i} \psi_{i-1} \psi_i dx &\approx \frac{h}{2} (\psi_{i-1}(x_{i-1}) \psi_i(x_{i-1}) + \psi_{i-1}(x_i) \psi_i(x_i)) = 0 \\ \int_{x_{i-1}}^{x_{i+1}} \psi_i^2 dx &\approx 2 \int_{x_{i-1}}^{x_i} \psi_i^2 dx \approx 2 \frac{h}{2} (\psi_i^2(x_{i-1}) + \psi_i^2(x_i)) = h \\ \int_{x_i}^{x_{i+1}} \psi_i \psi_{i+1} dx &\approx \frac{h}{2} (\psi_i(x_i) \psi_{i+1}(x_i) + \psi_i(x_{i+1}) \psi_{i+1}(x_{i+1})) = 0 \end{aligned}$$

Using the previous formulae to approximate the mass matrix coefficients, we get to the following diagonal matrix  $M_L$  whose elements are the sums of the elements of each row of  $M$ , i.e.

$$M_L = \text{diag}(\tilde{m}_{ii}), \tilde{m}_{ii} = \sum_{j=i-1}^{i+1} m_{ij} \quad (1)$$

Replacing  $M$  with  $M_L$  does not reduce the order of accuracy of the method.

The process of mass lumping can be generalized to the two-dimensional case when linear elements are used. For quadratic finite elements, instead, the above-mentioned procedure consisting in summing by rows would generate a singular mass matrix  $M_L$ . An alternative diagonalization strategy consists in using the matrix  $\hat{M} = \text{diag}(\hat{m}_{ii})$  with elements

$$\hat{m}_{ii} = \frac{m_{ii}}{\sum_j m_{ij}}$$

Mass Lumping Technique can be useful because it allows to stabilize diffusion-reaction equations when the reaction term is dominant. Indeed when this happens numerical oscillations may occur. If in this case we replace the mass matrix with its lumped version in the Galerkin approximation, we reobtain the finite difference scheme, thus an unconditionally stable solution [8].

In the Bidomain problem with the scaled Fitzhugh-Nagumo model, we are multiplying by  $\alpha$  the non linear reaction term  $\chi_m k V_m (V_m - a)(V_m - 1)$ . As explained in section 1.2, in order to obtain a faster, we need to adopt values of  $\alpha$  greater than 1. We can then expect that, as  $\alpha$  grows, the non-linear term tends to become dominant and stability problems may arise.

We verified that also  $C(V_m^n)$  presents the same tridiagonal pattern as  $M$ . Therefore, in those cases we expect that we can replicate the same procedure described in 2.5 for the mass matrix, but in this case on  $C(V_m^n)$ . For this reason we expect that in case of numerical instabilities arising when  $\alpha$  is too big, the usage of a lumped version of  $C(V_m^n)$  obtained in the same way as described in equation 1, could stabilize the scheme and lead to better results.

# 3 | NUMERICAL RESULTS

For all the simulations in this chapter these parameters have been used:

Initial condition for $V_M$	0
Initial condition for $w$	0
$\chi_m (m^{-1})$	$10^5$
$C_m (F m^{-2})$	$10^{-2}$

For what concerns the *Fitz-Hugh Nagumo* parameters, we consider these values:

$k$	$19.5 \cdot \alpha$
$\varepsilon$	$1.2 \cdot \alpha$
$\gamma$	0.1
$a$	$13 \cdot 10^{-3}$
$b$	$\alpha$

## 3.1 SINGLE CELL SIMULATIONS

As explained in section 1.1, the repolarization phenomenon occurs at microscopic scale and it is due to the passage of ions through the cell membrane. Thus in the bidomain mathematical model this aspect is ruled by the ionic model, which is responsible of the modeling of the behaviour of the action potential shown in figure 1.1 and therefore of the model at microscopic scale. Also Botta and Calafà, analizing their results, pointed out that the issue causing the non-repolarization was probably related to the ionic model.

In order to "isolate" the behaviour of the ionic model, we conducted this study performing a simplified simulation, the so called *single cell simulation* which consists in the bidomain problem with the diffusion term dropped and coupled with the Fitzhugh-Nagumo model: in this way the solution represents the action potential on one single excited cell. This problem reads as follow:

### Problem 3.1 (Bidomain Single Cell Simulation)

$$\begin{cases} \frac{dV_m}{dt} = \frac{I^{ext} - \chi_m \alpha k V_m (V_m - a) (V_m - 1) - \chi_m w}{\chi_m C_m} = a(V_m, w) \\ \frac{dw}{dt} = \varepsilon (V_m - \gamma w) = b(V_m, w) \end{cases}$$

Regarding the initial conditions, Botta and Calafà adopted  $(v_0, w_0) = (0, 0)$ , and we found in literature support to this choice. Indeed, according to [5], a technique to generate a physiological action potential consists in starting from resting initial conditions and applying an appropriate current pulse  $I^{ext}$  for a sufficiently long time interval  $T_{stim}$ . For these simulations we used  $I^{ext} = 200 \cdot 10^3 A m^{-3}$ ,  $T_{stim} = 10^{-3} s$ . In the FitzHugh-Nagumo model, the application of such current pulse shifts the  $V_m$  nullcline upward if  $I^{ext} > 0$ . This shift changes temporarily the equilibrium point for the duration of the stimulation interval  $T_{stim}$  so that  $(v_0, w_0)$  is

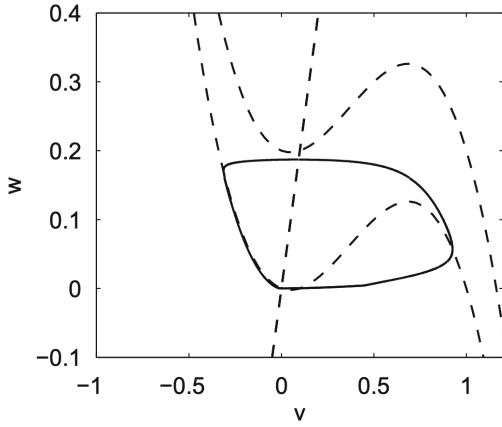


Figure 3.1: Single cell simulation

no longer an equilibrium point and the variables  $(v_0, w_0)$  start a trajectory toward the closest nullcline branch. When the pulse is turned off at time  $T_{stim}$ , the  $V_m$ -nullcline returns to the original curve. By this time,  $v$  and  $w$  have reached a point in phase-space that, for proper choices of pulse amplitude and interval duration, belongs to the excitable region, so that an action potential is generated (see figure 3.1).

Performing this simulation with  $\alpha = 1$ , corresponding to the classic Fitzhugh-Nagumo used in [1], using `ode45` function on MATLAB, we obtained the plot in 3.2: the repolarization occurs after  $\sim 3$ s. This was confirmed also performing a simulation using the complete bidomain code. Therefore, the repolarization effectively takes place and it is well captured, but the times are way too dilated and totally non-physiological.

For this reason we adopted the scaled version of the FN model with  $\alpha > 1$  in order to increase the response velocity of the model, as explained in section 1.2. As expected, reperforming the single cell simulation as above with increasing value of  $\alpha$ , the time at which repolarization occurred was significantly reduced, as we can see from figure 3.3.

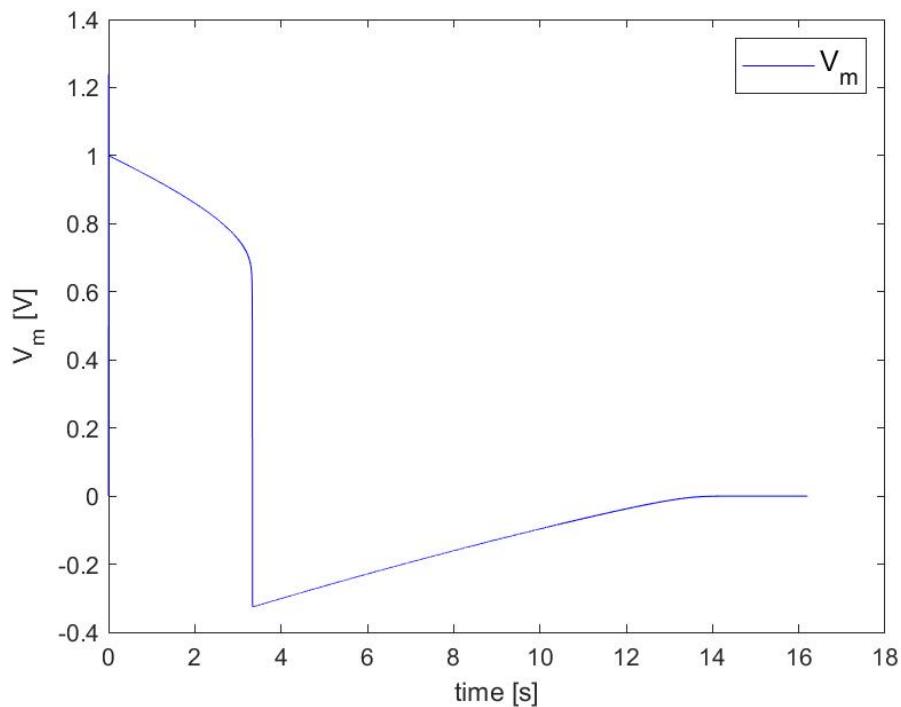


Figure 3.2: Single cell simulation considering  $\alpha = 1$

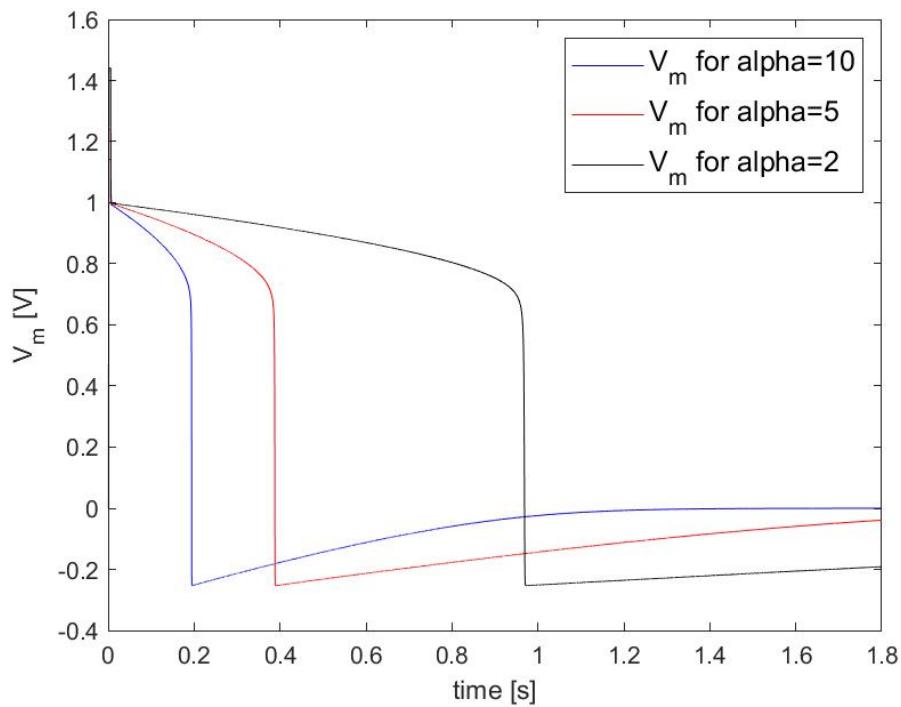


Figure 3.3: Single cell simulation for different  $\alpha$

## 3.2 ERROR ANALYSIS

Considering some example problems, we provide now an experimental error analysis that can show that the modifications introduced in the model do not effect efficiency and goodness of our numerical schemes. For all the simulations, we choose the same parameters used by Bottà and Calafà and we expected to obtain similar results. Values of the parameters are reported in tables above.

We choose as exact solutions:

$$\begin{aligned} V_m &= \sin(2\pi x)\sin(2\pi y)e^{-5t} \\ w &= \frac{\varepsilon}{\varepsilon\gamma - 5}\sin(2\pi x)\sin(2\pi y)e^{-5t} \end{aligned}$$

From these assumptions, we compute the r.h.s., the boundary conditions and initial conditions accordingly.

Choosing as basis the Lagrangian hat basis functions, we performed an error analysis related to different polynomials degrees. We expect to see different error orders for different polynomials orders. As it's possible to verify from Figure 3.4, 3.5, 3.6, we obtained the following results:

- P1: our results are exactly as expected and show a first order for  $V_m$  errors in  $H^1$  and DG norms while a second order for  $V_m$  errors in  $L^2$  and  $L$  norms.
- P2: for what concerns second order polynomials we can observe slightly different results. Indeed we see a flatter segment that some error trends have for small elements sizes. However, this is due to the influence of other causes of errors (especially the time-discretization errors) when the space discretization errors become very small. This is the reason why this was not present in the previous case where space errors were still too big and other causes of error negligible. We observe a quadratic convergence rate for the  $V_m$  error in the energy norm and a cubic convergence rate for the  $V_m$  error in the  $L^2$  and  $L$  norms.
- P3: for polynomials of third degree we still find the expected convergence rate, that are third order for  $V_m$  errors in  $H^1$  and DG norms and fourth order for  $L^2$  and  $L$  norms, excepted for the phenomena discussed in the previous point.

Thus we can state that the introduction of the scaling parameter  $\alpha$  does not affect the goodness of the numerical scheme.

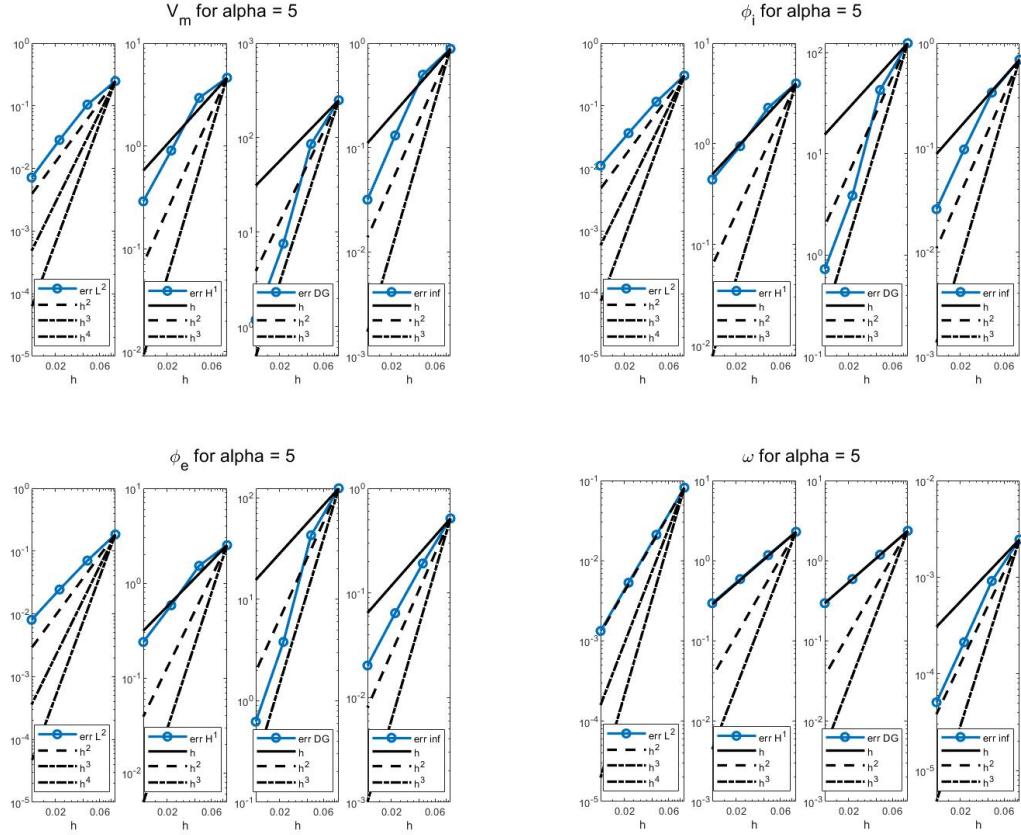
We replicate the same error analysis for Dubiner basis functions, adopting the different time discretization methods described previously. As expected we obtain the same results of Bottà and Calafà, so we refer to [1] for a more detailed study.

### 3.2.1 Dominant reaction case

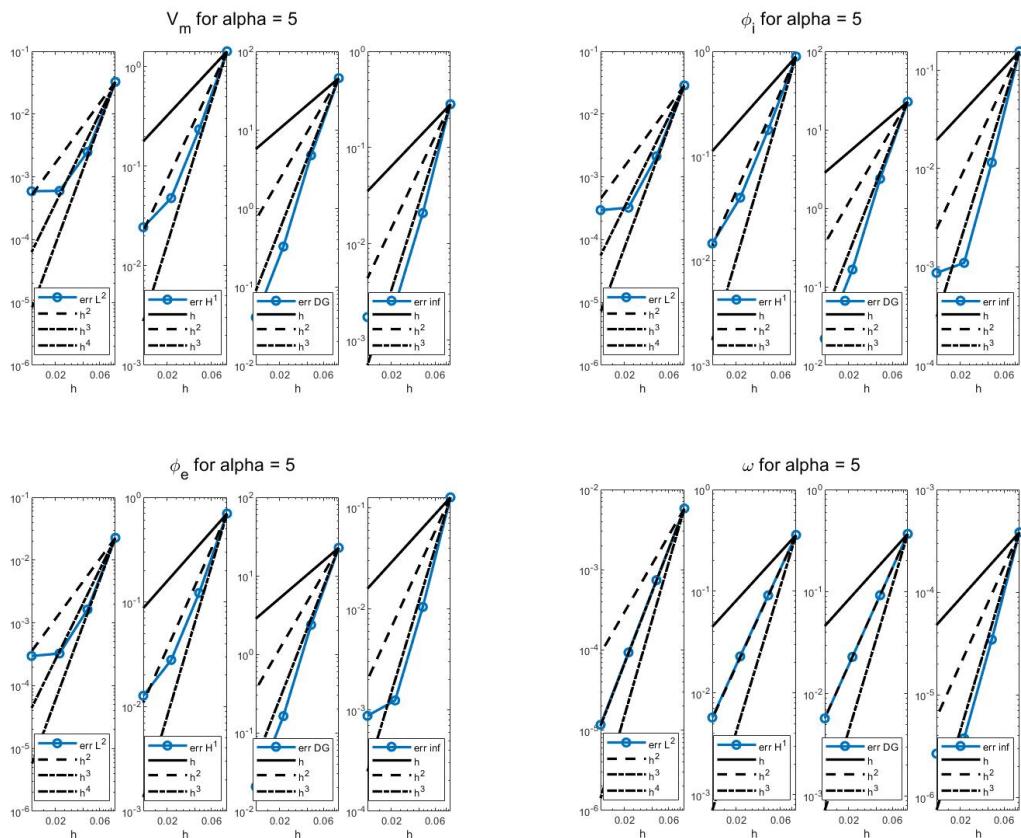
We performed this error analysis adopting different value of  $\alpha$  in order to analyse what happens when the reaction term becomes dominant. Indeed we expect that some stability problems may arise, compromising the goodness and efficiency of our numerical scheme. After several simulations it turns out that  $\alpha = 10$  is the best cut-off value between good stability and physiological behaviour of our dynamic system. For this reason  $\alpha = 10$  will be used in the next simulations.

In Figure 3.7 (left) it's possible to see that, due to instability, for an higher value of the parameter convergence properties described above are lost. Therefore for higher value of  $\alpha$  we

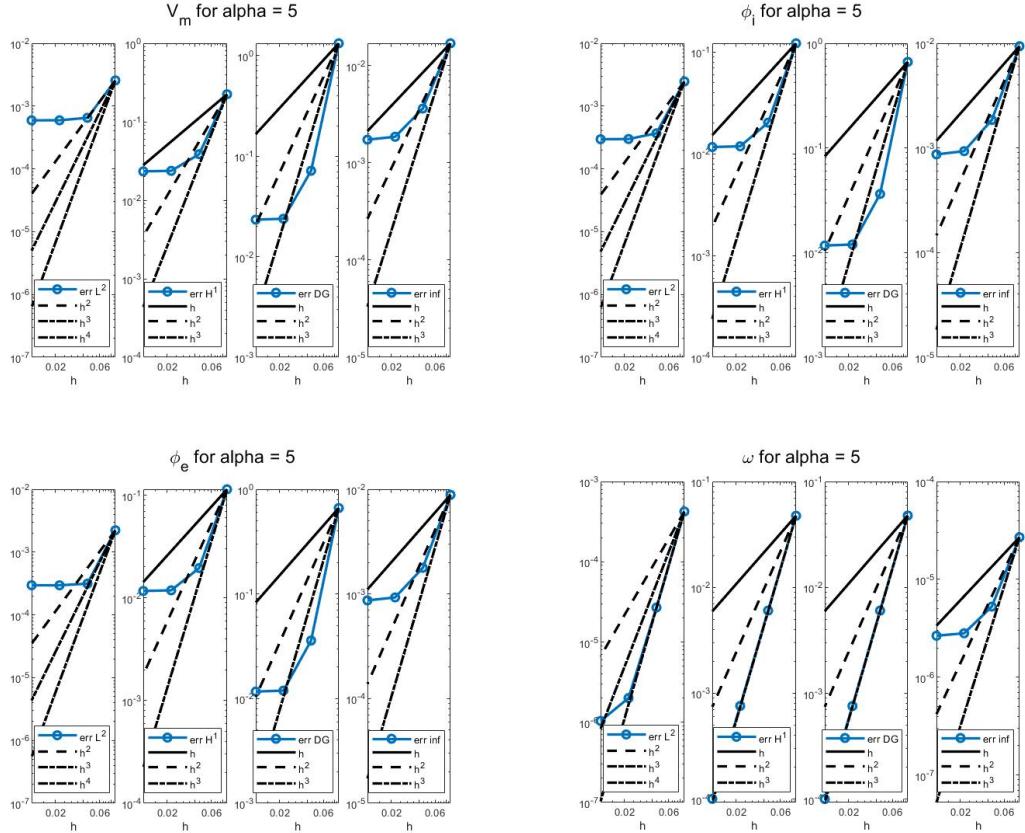
decided to implement the lumping technique applied on  $C(V_m^n)$ , as explained in 1. In Figure we can see that the results obtained are not good as expected.



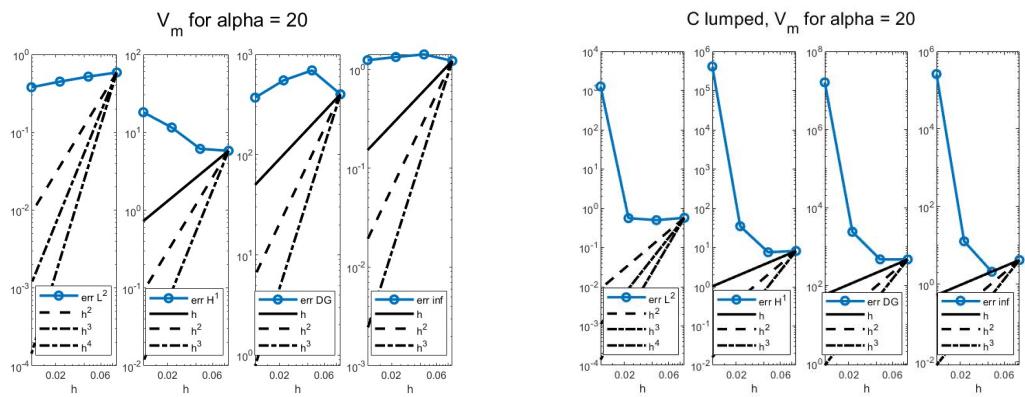
**Figure 3.4:** Computed errors for FEM with first order polynomials



**Figure 3.5:** Computed errors for FEM with second order polynomials



**Figure 3.6:** Computed errors for FEM with third order polynomials



**Figure 3.7:** Computed errors for FEM with first order polynomials, non-lumped version (left) vs lumped version (right)

### 3.3 REALISTIC SIMULATIONS

In order to mimic a pseudo-realistic behaviour of our simulations we brought some changes in the parameters of the model, especially in what concerns the direction of the fibres and the dimension of the cardiac domain.

Physically, our domain represents a square approximation of a ventricle, that is electrically isolated (homogeneous boundary conditions), with an external current which is applied in a small region for a limited interval of time ( $t \in (0.001, 0.002]$ ). The intensity of the current varies in our test cases, depending on the various situations that we studied.

The time interval has been chosen  $\Delta t = 0.0001$ .

#### 3.3.1 Conductivity values

In literature, a lot of different sets of conductivity values exist. Possible reasons for the discrepancies between the sets of conductivity values could be experimental conditions, mathematical models and methods used to determine the conductivities from data; it could also depend from the animal species considered during the experiment ([11]).

For example Roth stated a set of acceptable ranges for the conductivity values that are:

$$\sigma_l^i \in [0.17, 0.34]; \sigma_t^i \in [0.02, 0.06]; \sigma_l^e \in [0.12, 0.62]; \sigma_t^e \in [0.08, 0.24]$$

We took into account the upper bounds of these values:

**Table 1:** Max values of Roth's ranges

$\sigma_l^i$	0.34
$\sigma_t^i$	0.06
$\sigma_l^e$	0.62
$\sigma_t^e$	0.24

Other sets of values, as cited here [10], could be:

**Table 2:** L. Clerc

$\sigma_l^i$	0.17
$\sigma_t^i$	0.019
$\sigma_l^e$	0.62
$\sigma_t^e$	0.24

**Table 3:** D.E. Roberts, L.T. Hersh, A.M. Scher

$\sigma_l^i$	0.28
$\sigma_t^i$	0.026
$\sigma_l^e$	0.22
$\sigma_t^e$	0.13

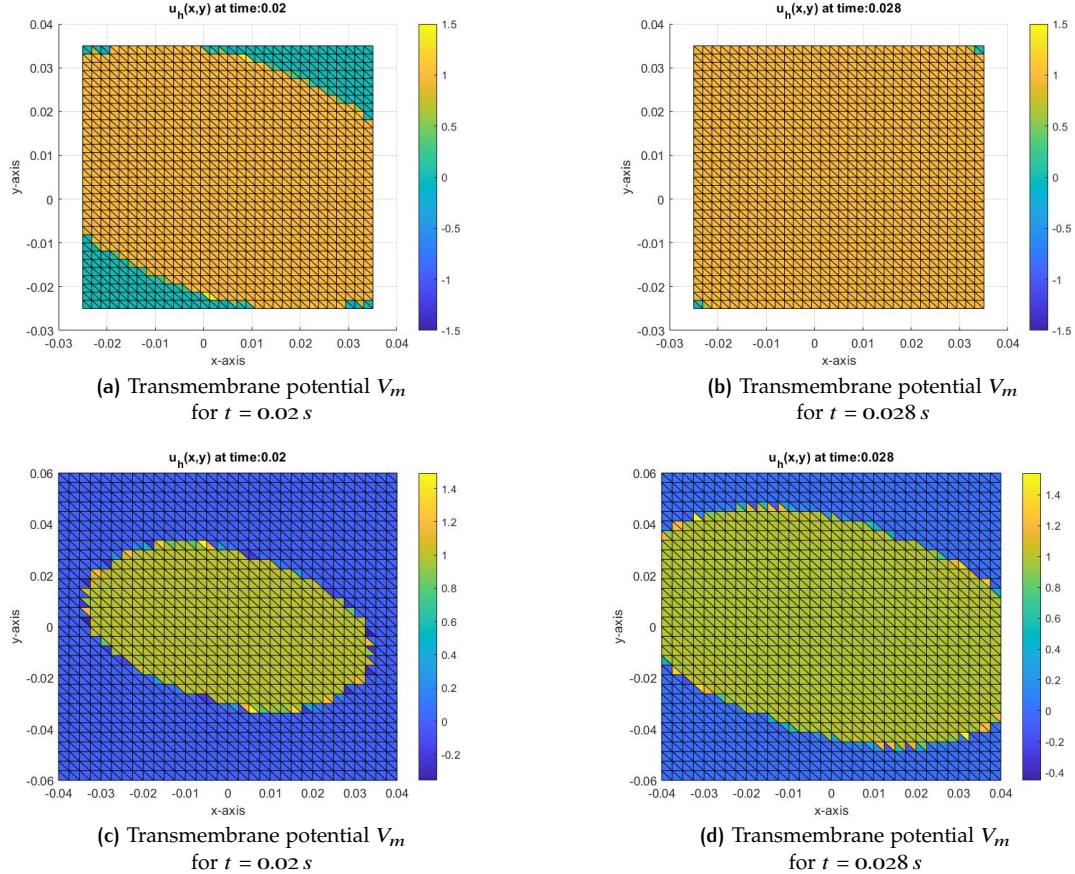
**Table 4:** D.E. Roberts, A.M. Scher

$\sigma_l^i$	0.34
$\sigma_t^i$	0.06
$\sigma_l^e$	0.12
$\sigma_t^e$	0.08

According to [10], the best among these three possibilities is the one proposed by D.E. Roberts, L.T. Hersh, A.M. Scher and shown in table 3. Thus, we compared these values with the maximum values of the Roth's acceptable ranges.

As we can see in figure 3.8, the overall membrane potential behaviour does not change; the only main difference is the diffusion velocity, the diffusion considering the Roth's values is a little bit faster than the other one. For that reason, in our test cases, we decided to use the set of conductivity values reported in Table 1.

**Figure 3.8:** Comparing potential diffusion between using Table 1 set of values (first row) and Table ?? set of values (second row).



### 3.3.2 1<sup>st</sup> test-case (*overflow vs underflow case*)

In our first test case, our aim was to consider the same situation considered by Botta and Calafà in their paper [1], in order to better understand how our modifications in the model reflect on the numerical results. In this test case, we aim to show two different situations depending on the external current intensity: in the first one, for too weak currents, the electrical activation should miss and this implies that the potentials are not capable to hold up (*underflow* case). In the second one, if the intensity is over a certain threshold, we should see the electrical activation and the resulting diffusion (*overflow* case). After several simulations, we have proved that:

- $I_{ext} = 150 \cdot 10^3 \text{ A m}^{-3}$  is a suitable value for the *underflow* case (figure 3.9).
- $I_{ext} = 200 \cdot 10^3 \text{ A m}^{-3}$  is a suitable value for the *overflow* case (figure 3.10).

In these numerical simulations the square domain  $\Omega = (-0.025, 0.035) \times (-0.025, 0.035)$  has been used and we considered that all the fibers of our domain are directed along the x-axis:

$$a_l = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{ and } a_t = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \Rightarrow \Sigma_i = \begin{bmatrix} 0.34 & 0 \\ 0 & 0.06 \end{bmatrix} \text{ and } \Sigma_e = \begin{bmatrix} 0.62 & 0 \\ 0 & 0.24 \end{bmatrix}$$

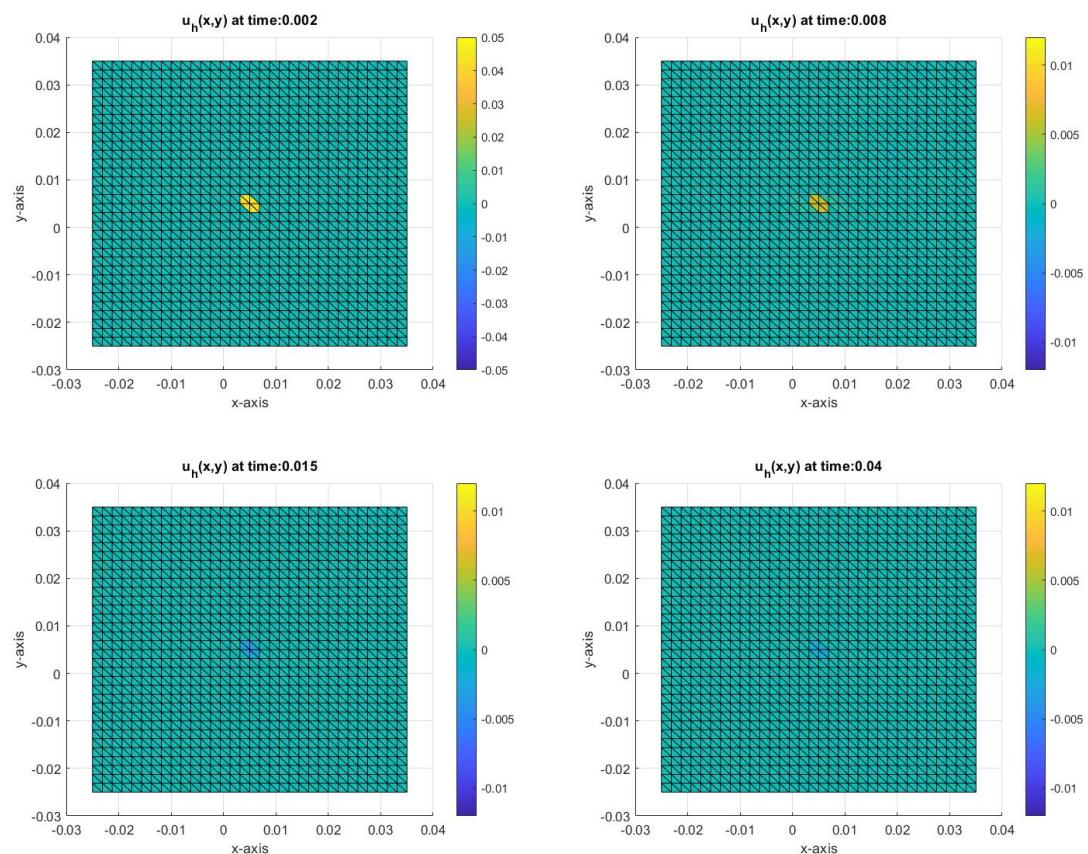


Figure 3.9: Underflow case

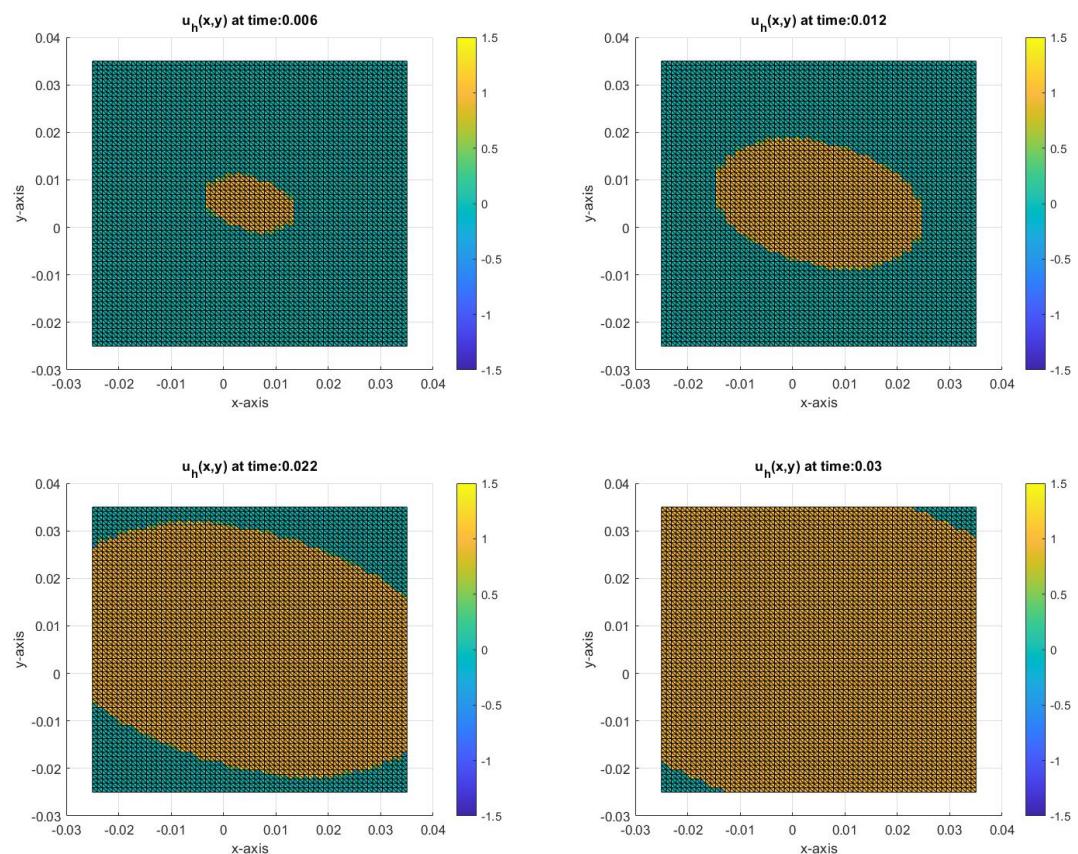


Figure 3.10: *overflow case*

### 3.3.3 2<sup>nd</sup> test-case: impulse from the angle

For the second simulation we applied a current

$$I_{ext} = 200 \cdot 10^3 \cdot \chi_{[x < -0.023] \cdot [y < -0.023]}(x, y) \cdot \chi_{[0.001, 0.002]}(t) A m^{-3}$$

at the left-down corner of the domain. Also in this case we considered the domain  $\Omega = (-0.025, 0.035) \times (-0.025, 0.035)$  and the fibres directed along the x-axis, with conductivity tensors:

$$\Sigma_i = \begin{bmatrix} 0.34 & 0 \\ 0 & 0.06 \end{bmatrix} \text{ and } \Sigma_e = \begin{bmatrix} 0.62 & 0 \\ 0 & 0.24 \end{bmatrix}$$

In this case we can verify very well how the potential propagation is faster along the fibers' direction than along the transversal direction. As a matter of fact, at the time instant when, following the fibers' direction, the potential reaches the right side of the square domain, it needs to cover still circa  $\frac{1}{4}$  of the y-axis, which is also the transversal fibers' direction, to reach the upper side of the domain.

Moreover this simulation allows to verify the fact that repolarization is well captured and occurs in a much more physiological time ( $\sim 0.3s$ ).

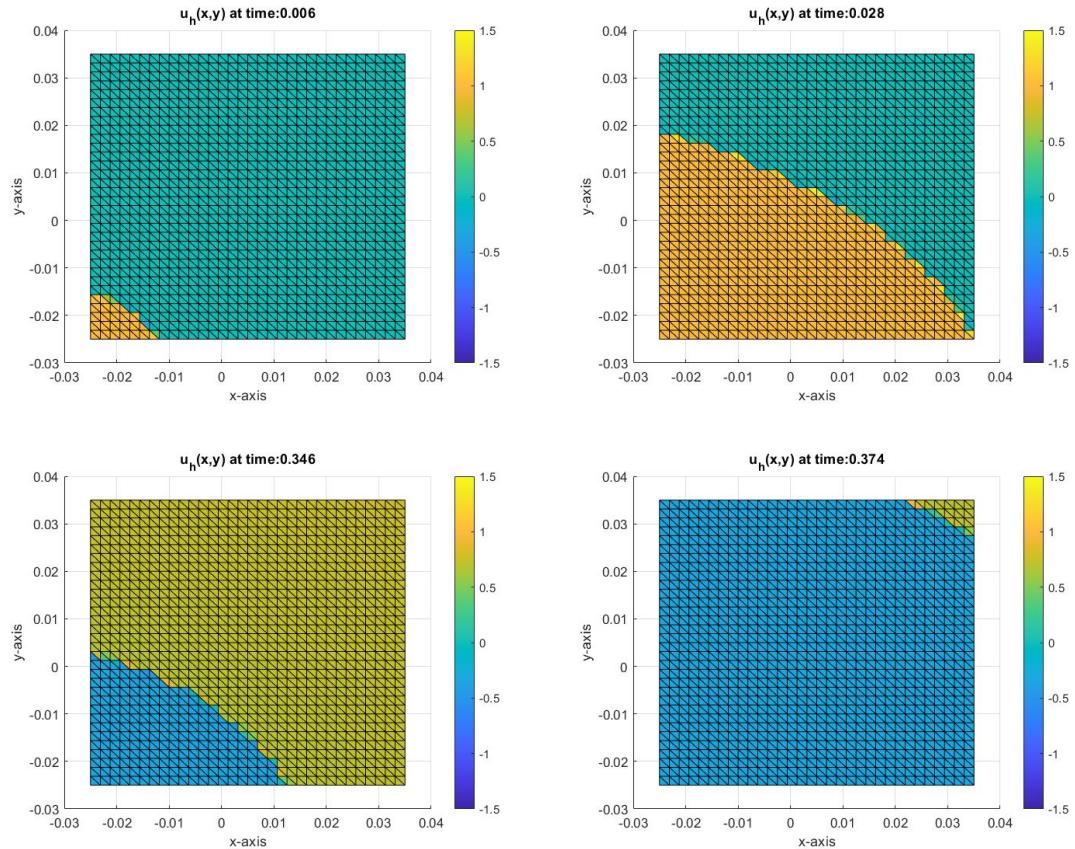


Figure 3.11: applied current to the corner

### 3.3.4 3<sup>rd</sup> test-case: realistic fibers directions

To enrich our pseudo-realistic simulations, we change the dimension of the domain ( $\Omega = (-0.04, 0.04) \times (-0.06, 0.06)$ ), to make it more similar to the section of a ventricle.

As explained in section 1.2.1 the myocardium exhibits a complex three-dimensional spatial organization where myocytes are connected to form cardiac fibers with varying orientation. Thus the exact fiber orientation is difficult to measure and it's impossible to replicate in a 2D-model. Nevertheless to perform a simulation where the fibers orientation is slightly more physiological, we exploited the results obtained in [7], as we can see from figure 3.12 We divide our domain in three zones, depending on the direction of the fibres in each zone:

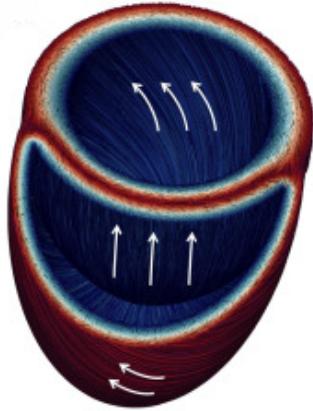


Figure 3.12: Idealized biventricular model

- $y \leq -0.03$  : the fibers are oriented as an angle of 30, that is  $a_l = \begin{bmatrix} -\frac{\sqrt{3}}{2} \\ \frac{1}{2} \end{bmatrix}$
- $-0.03 < y < 0.03$  : the fibres have a vertical direction, that is  $a_l = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$
- $y \geq 0.03$  the fibers have a diagonal direction, that is  $a_l = \begin{bmatrix} -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{bmatrix}$

The conductivity tensors are then

$$\Sigma_i = \begin{bmatrix} 0.2 & -0.14 \\ -0.14 & 0.2 \end{bmatrix} \cdot \mathbf{I}_{\{y \geq 0.03\}} + \begin{bmatrix} 0.06 & 0 \\ 0 & 0.34 \end{bmatrix} \cdot \mathbf{I}_{\{y < 0.03 \& y > -0.03\}} + \begin{bmatrix} 0.27 & -0.121 \\ -0.121 & 0.13 \end{bmatrix} \cdot \mathbf{I}_{\{y \leq -0.03\}}$$

$$\Sigma_e = \begin{bmatrix} 0.43 & -0.19 \\ -0.19 & 0.43 \end{bmatrix} \cdot \mathbf{I}_{\{y \geq 0.03\}} + \begin{bmatrix} 0.24 & 0 \\ 0 & 0.62 \end{bmatrix} \cdot \mathbf{I}_{\{y < 0.03 \& y > -0.03\}} + \begin{bmatrix} 0.525 & -0.1645 \\ -0.1645 & 0.335 \end{bmatrix} \cdot \mathbf{I}_{\{y \leq -0.03\}}$$

where each conductivity tensor is computed as follow:

$$\Sigma = \sigma_l \mathbf{a}_l \cdot \mathbf{a}_l^T + \sigma_t \mathbf{a}_t \cdot \mathbf{a}_t^T$$

Then we needed to modify the part of the code (matrix2D.m) where the stiffness matrix is assembled and the penalty coefficients computed (see A.2).

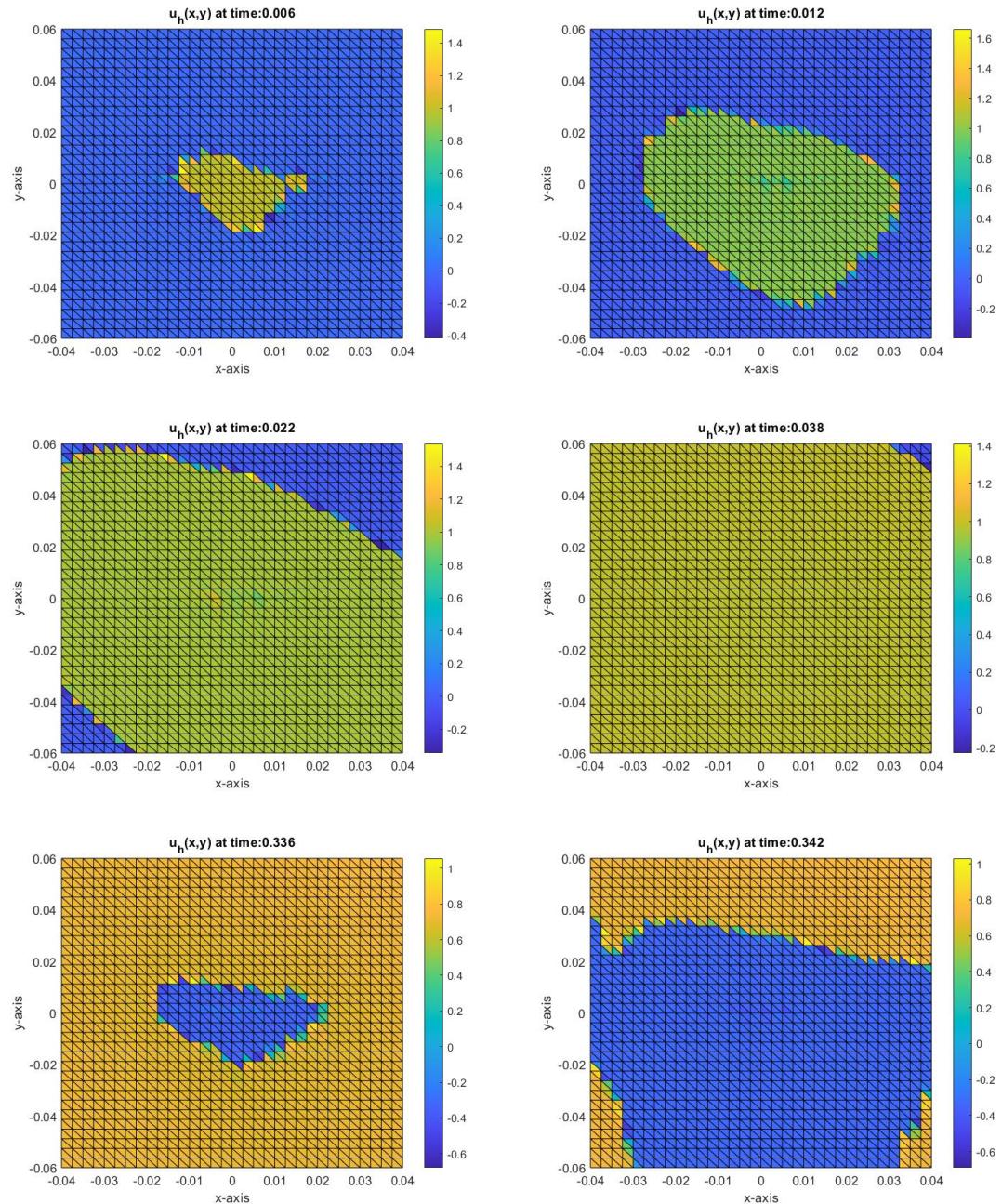
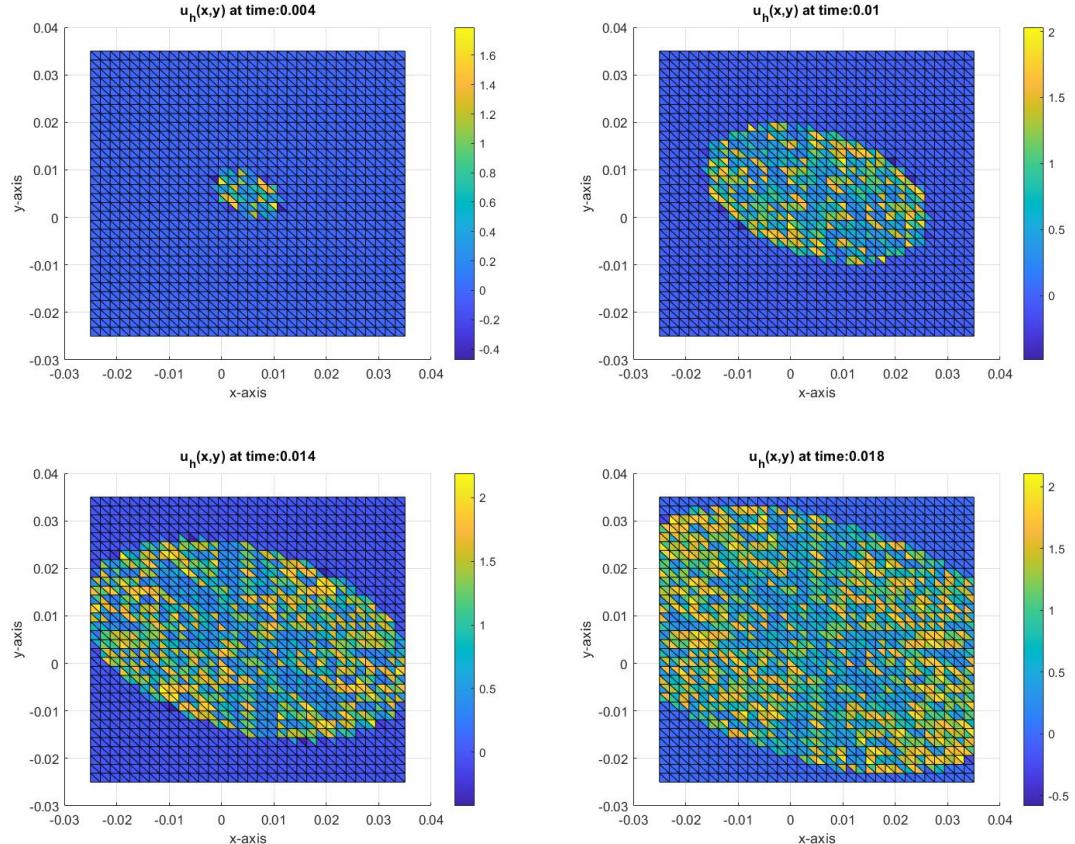


Figure 3.13: Different fibers direction

### 3.3.5 4<sup>th</sup> test-case: high value of $\alpha$ , stability and lumping

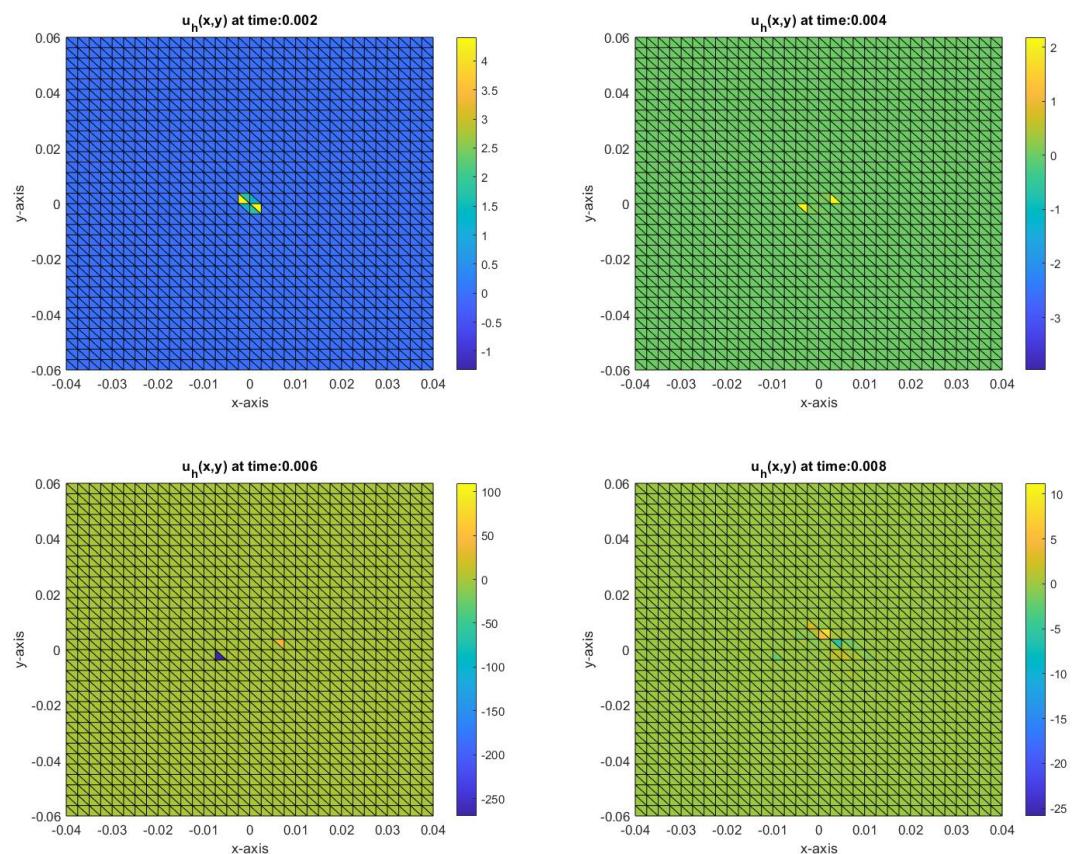
In this last test case our aim was to verify our predictions about stability when the value becomes large (see sections 1 and 3.2.1).

We performed the same simulation described in the first test case (3.3.2) in the overflow situation ( $I_{ext} = 200 \cdot 10^3 A m^{-3}$ ) with  $\alpha = 15$ . As expected, the solution is characterized by a considerable instability, as one can see in figure 3.14.



**Figure 3.14:** simulation with  $\alpha = 15$

We tried also in this case to perform the same simulation with the lumped version of the matrix  $C$ , but as expected after the bad results in the error analysis, the result is not satisfactory: the dynamic of the phenomenon is totally lost, as one can see in figure 3.15.



**Figure 3.15:** simulation with  $\alpha = 15$  and lumped version of C

# 4 | CONCLUSIONS

We succeed in our goal of obtaining a more physiological behaviour of our dynamic system, by the introduction of the scaling parameter  $\alpha$ . The phenomenon of repolarization is now well captured and occurs in realistic times. Moreover we add in the code the possibility of using fibers with different directions inside the domain. In this way it is possible to better simulate the real behaviour of the physical phenomenon. For what concerns the stabilization of reaction dominated problems with the lumping technique, we do not obtain the desired results. This may happen because of the non-linearity of the reaction term; indeed, we did not find in literature other attempts of the usage of this technique for reaction term which are not linear. More studies about this topic could be carried out in the future.

# A | THE CODE

## A.1 SINGLE CELL SIMULATIONS

The single cell simulations described in section 3.1 have been obtained through the following code, which exploit the Matlab function `ode45` to solve a system of ordinary differential equations.

```

1     fun = @(t,x)
2         [ (Ie*(t<0.2)-Chim*(k*x(1).*(x(1)-a).* (x(1)-1)+x(2)))/(Chim*Cm) ;
3             eps*(x(1)-gamma*x(2)) ];
4
5     [t,y] = ode45(fun, [0 T_max], [0 0]);
6     figure
7     plot(t,y(:,1)) % plot of V_m
8     plot(t,y(:,2)) % plot of w

```

To compute the nullclines of the Fitzhugh-Nagumo system, we readapted the code that can be found in [14]

## A.2 FIBERS DIRECTIONS

In order to have the possibility to prescribe different fiber directions in the domain, we modified the code in the function `matrix2d.m` and `matrix2d_Dubiner.m`:

```

97   for k = 1:length(w_2D)
98
99       dx = w_2D(k)*det(BJ);
100
101      x = pphys_2D(k,1);
102      y = pphys_2D(k,2);
103
104      F_i = eval(Data.source_i);
105      F_e = eval(Data.source_e);
106
107      sigma_i = eval(Data.Sigma_i);
108      sigma_e = eval(Data.Sigma_e);
109
110      for i = 1 : femregion.nln
111          % assembly load vector
112          f_i(index(i)) = f_i(index(i)) + F_i*dphiq(1,k,i).*dx;
113          f_e(index(i)) = f_e(index(i)) + F_e*dphiq(1,k,i).*dx;
114
115      for j = 1 : femregion.nln
116          % assembly stiffness matrix
117          Vi(index(i),index(j)) = Vi(index(i),index(j)) ...
118              + ((Grad(k,:,i) * BJinv) * (Grad(k,:,j) * BJinv * ...
119              sigma_i')').*dx ;
120          Ve(index(i),index(j)) = Ve(index(i),index(j)) ...

```

```

120      + ((Grad(k,:,i) * BJinv) * (Grad(k,:,j)* BJinv * sigma_e' ...
121      ') .*dx ;
122      ;
123      end
124  end

```

```

140  for iedg = 1 : neighbour.nedges
141
142      neigedge = neighedges_ie(iedg);
143
144      penalty_scaled = penalty_coeff./meshsize(iedg);
145
146          % assembly of interface matrices
147      for k = 1:nqn_1D    % loop over 1D quadrature nodes
148
149          x = pphys_1D(k,1,iedg);
150          y = pphys_1D(k,2,iedg);
151
152          sigma_i = eval(Data.Sigma_i);
153          sigma_e = eval(Data.Sigma_e);
154
155          penalty_sigma_i = abs(normals(:,iedg)'*sigma_i*normals(:,iedg));
156          penalty_sigma_e = abs(normals(:,iedg)'*sigma_e*normals(:,iedg));

```

### A.3 LUMPING

To introduce the possibility of lumping the matrix  $C$ , we modified the code in the function `assemble_nonlinear.m`

```

90  if (Data.C_lump == 'Y')
91  last = femregion.ndof;
92  n = zeros(last, 1);
93
94  switch Data.fem
95      case 'P1'
96          for j = 0:last/3-3
97              x = 3*j + 1;
98              n(x) = C(x,x) + C(x,x+1) + C(x,x+2);
99              x = x + 1;
100             n(x) = C(x,x-1) + C(x,x) + C(x,x+1);
101             x = x + 1;
102             n(x) = C(x,x-2) + C(x,x-1) + C(x,x);
103         end
104         C = diag(n);
105         C = sparse(C);
106
107     case 'P2'
108         sum = 0;
109         for i=1:last
110             n(i) = C(i,i);
111             sum = sum + C(i,i);
112         end
113         n = n / sum;
114         C = diag(n);
115         C = sparse(C);

```

```
116      end  
117  
118  end
```

## BIBLIOGRAPHY

- [1] F. Botta, M. Calafà. A High-Order Discontinuous Galerkin Method for the Bidomain Problem of Cardiac Electrophysiology (Politecnico di Milano, NAPDE project 2021).
- [2] F. Andreotti and D. Uboldi. Discontinuous Galerkin approximation of the monodomain problem (Politecnico di Milano, NAPDE project 2021).
- [3] L. Marta, M. Perego. Discontinuous Galerkin approximation of the bidomain system for cardiac electrophysiology (Politecnico di Milano, NAPDE project 2021).
- [4] A. Quarteroni, A. Manzoni, and C. Vergara. The cardiovascular system: Mathematical modelling, numerical algorithms and clinical applications. In: *Acta Numerica* (2017), pp. 365-590.
- [5] P. Colli Franzzone, L. F. Pavarino, S. Scacchi. Mathematical Cardiac Electrophysiology.
- [6] B. E. Griffith, C. S. Peskin. C.S.: Electrophysiology. *Commun. Pure. Appl. Math.* 66, 1837–1913 (2013)
- [7] R. Piersanti, P. C. Africa, M. Fedele et al. Modeling cardiac muscle fibers in ventricular and atrial electrophysiology simulations
- [8] A. Quarteroni, A. Valli. Numerical approximation of partial differential equations, p. 261.
- [9] A. Quarteroni. Numerical models for differential problems, pp. 326-328.
- [10] Peter R. Johnston. A sensitivity study of conductivity values in the passive bidomain equation (2011).
- [11] Barbara M. Johnston, Peter R. Johnstone. Approaches for determining cardiac bidomain conductivity values: progress and challenges (2020).
- [12] S. J. Sherwin, G. E. Karniadakis. A new triangular and tetrahedral basis for high-order finite element methods. In: *International journal for numerical methods in engineering* 38 (1995), pp. 3775-3802.
- [13] R. Spiteri and S. Torabi Ziaratgahi. Operator splitting for the bidomain model revisited. In: *Journal of Computational and applied Mathematics* 296 (2016), pp. 550-563.
- [14] Ian Cooper. Fitzhugh-Nagumo model for spiking neurons. <https://d-arora.github.io/Doing-Physics-With-Matlab/mpDocs/pbs006.htm>
- [15] Anthony Kanai, Guy Salama. Optical Mapping Reveals That Repolarization Spreads Anisotropically and Is Guided by Fiber Orientation in Guinea Pig Hearts (1995)
- [16] R. Doste, D. Soto-Iglesias, G. Bernardino, A. Alcaine, R. Sebastian, S. Giffard-Roisin, M. Sermesant, A. Berrueto, D. Sanchez-Quintana, O. Camara. A rule-based method to model myocardial fiber orientation in cardiac biventricular geometries with outflow tracts