POLITECNICO

MILANO 1863

A HIGH-ORDER DISCONTINUOUS GALERKIN METHOD FOR THE BIDOMAIN PROBLEM OF CARDIAC ELECTROPHYSIOLOGY

Authors: Federica Botta, Matteo Calafà

Supervisors: Christian Vergara, Paola Antonietti

Contents

1 Introduction

1.1 Abstract

The aim of the project is to study and implement a suitable numerical scheme for the resolution of the *Bidomain Problem*, a famous system of equations that has been developed in the context of the electrophysiology of human heart.

This work is basically the continuation of a two-years-long study carried out by three past course projects ([3], [1], [8]). In particular, the very goal of this project is to improve the results obtained in [8] (Marta and Perego) for the Bidomain model. In fact, even if a *Discontinuous Galerkin* discretization has been successfully implemented, results are not satisfactory from the point of view of stability and convergence. We think this notice is noteworthy as this work is primarily based on these provided data and codes. Through this article, it will be illustrated how we managed to solve these problems extending, optimizing and correcting these past numerical strategies.

1.2 The physical problem

We intend to present the physical meaning of the Bidomain equations only briefly since it has already been widely shown in the previous project (Marta and Perego). For a more complete explanation, we instead refer to [10].

The mechanical contraction and expansion of human heart has its origin in the *electrical activation* of the cardiac cells. At every heart-beat, myocyties are activated and deactivated following a characteristic electrical cycle (fig 1).
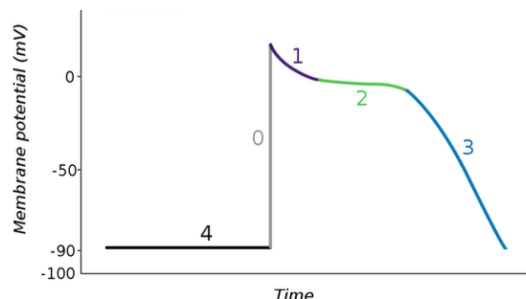
Figure 1: Membrane potential in function of time (one cardiac cycle)

The cell is initially at rest ($-90mV$, step 4). At a certain point, its potential increases rapidly ($\approx 2ms$) and reaches the value of $+20mV$: the cell is activated. Later, a plateau near $0mV$ is observed and then a slow repolarization to the initial potential.

From a microscopical point of view, we could study the dynamics acting in each single cell (as a consequence of the passage of chemical ions through specific channels, e.g. $Ca2+, Na+, K+$). From a macroscopical point of view, instead, one can observe it as a continuous electrical diffusion over the entire cardiac surface. Even if this consists in a very rapid phenomenon, the study of such propagation could be very interesting in order, for instance, to detect diseases in sick patients.

2

1.3   Mathematical models

Starting from the circuit in figure 2, applying some general electromagnetism laws and some calculations, the Bidomain model has been formulated (see [10] for more details and/or [5] for the complete passages).
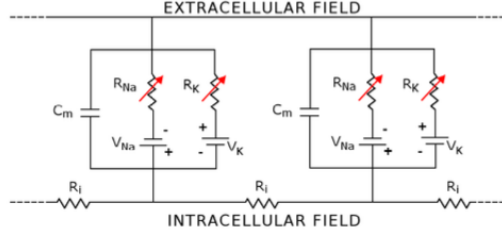
Figure 2: Simplified circuit to model the intracellular and extracellular potentials dynamics

The general formulation is then:

Definition 1 (Bidomain model)**.**

$$\begin{cases} \chi_m C_m \frac{\partial V_m}{\partial t} - \nabla \cdot (\Sigma_i \nabla \phi_i) + \chi_m I_{ion} = I_i^{ext} & in \ \Omega_{mus} \times (0, T] \\ -\chi_m C_m \frac{\partial V_m}{\partial t} - \nabla \cdot (\Sigma_e \nabla \phi_e) - \chi_m I_{ion} = -I_e^{ext} & in \ \Omega_{mus} \times (0, T] \end{cases}$$

where:

– $\boldsymbol{\phi_i, \phi_e}$ are the *Intracellular and Extracelllular Potentials* (unknowns)

– $V_m = \phi_i - \phi_e$ is the *Trans-membrane Potential*

– χ_m, C_m are known positive constants

– Σ_i, Σ_e are known positive definite tensors

– I_i^{ext}, I_e^{ext} are applied currents

– I_{ion} is the *Ionic Current*

– Ω_{mus} is the cardiac domain (myocardium + endocardium + epicardium)

Actually, this system is not complete since it misses boundary and initial conditions and a suitable model for I_{ion}. Initial conditions and Neumann boundary conditions for ϕ_i and ϕ_e are then imposed. For the definition of I_{ion}, instead, a *reduced ionic model* is chosen, in particular the *FitzHugh-Nagumo model*. Summing up:

Definition 2 (Bidomain + FitzHugh-Nagumo model with Neumann boundary conditions).

$$
\begin{cases}
\chi_m C_m \frac{\partial V_m}{\partial t} - \nabla \cdot (\Sigma_i \nabla \phi_i) + \chi_m I_{ion}(V_m, w) = I_i^{ext} & in \ \Omega_{mus} \times (0, T] \\
-\chi_m C_m \frac{\partial V_m}{\partial t} - \nabla \cdot (\Sigma_e \nabla \phi_e) - \chi_m I_{ion}(V_m, w) = -I_e^{ext} & in \ \Omega_{mus} \times (0, T] \\
I_{ion}(V_m, w) = k V_m (V_m - a)(V_m - 1) + w & in \ \Omega_{mus} \times (0, T] \\
\frac{\partial w}{\partial t} = \epsilon(V_m - \gamma w) & in \ \Omega_{mus} \times (0, T] \\
\Sigma_i \nabla \phi_i \cdot n = b_i & on \ \partial\Omega_{mus} \times (0, T] \\
\Sigma_e \nabla \phi_e \cdot n = b_e & on \ \partial\Omega_{mus} \times (0, T] \\
Initial \ conditions \ for \ \phi_i, \phi_e, w & in \ \Omega_{mus} \times \{t = 0\}
\end{cases}
$$

where:

– $\boldsymbol{w}$ is the *gating variable* (unknown)

– k, a, ϵ, γ are known constants

– b_i, b_e are the boundary conditions data

– n is the outward normal vector

From now on, the system of definition 2 will be the reference analytical problem for the development of numerical schemes.

To conclude, there exist other famous and useful models, such as the *Monodomain model*. But this is just a simplification of the Bidomain as in this case it is assumed that ϕ_i and ϕ_e are proportional. However, thanks to its simplicity, we often tested the code starting from the Monodomain implementation of the project [1] instead of analyzing directly the Bidomain.

1.4 A short discussion about the past works issue

As we have already introduced, our project initially aimed to continue and improve the work of a previous project ([8]).

Results obtained using unitary parameters, namely $\chi_m = \Sigma_i = \Sigma_e = C_m = k = \epsilon = \gamma = a = 1$, were actually quite satisfactory. On the other hand, the choice of more realistic/experimental values for the parameters (that are often very big or very small) caused bad consequences to the accuracy of the schemes or even to their stability. In particular, we observed that the choice of $C_m \approx 10^{-2}$ highly compromised the stability of the numerical schemes. This issue heavily limits the use of the code for research and/or experimental simulations as it guarantees convergence to the right solution only in few and non-realistic problems.

After a while, we realized that an inverted sign of the FitzHugh-Nagumo model formula occurred in [10].

This oversight was not only essential for the fidelity to the real phenomena but also crucial for the well-posedness of the problem.

We could give two motivations to reinforce this last statement: first of all, if we consider the well known study of Bourgault, Coudière, and Pierre ([4]), the conditions required for the well-posedness of the Bidomain problem are not satisfied if the sign is inverted (neither for the existence, hypothesis H4, nor for the uniqueness).

Secondly, suppose to discretize the Bidomain problem in time and to treat the non-linear term semi-implicitly, as it will be done in the following sections. Then, if we fix the timestep and if data from the previous timestep are given, we achieve a linear problem that can be easily switched into a weak formulation. From this analysis, we could observe that if $V_m < a$ or $V_m > 1$ and C_m is sufficiently small, the associated bilinear form is not coercive.

This second motivation, even if not very formal and with a mixed approach, is particularly interesting since a few confirmations occurred during simulations, as shown for instance in figure 3.
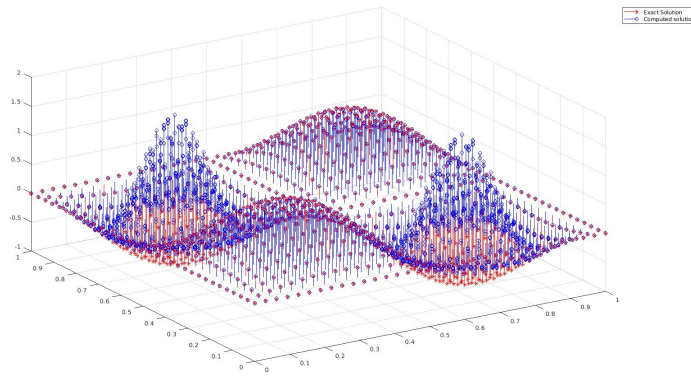

Figure 3: Comparison between exact and computed solution with inverted FitzHugh-Nagumo model: huge errors arise only when $Vm < 0$

In conclusion, the issue of the past works [1], [8] had not a numerical nature as expected, but instead an analytical origin due to an ill-posed problem.

2 Semi discretized numerical methods

2.1 DG discrete formulation

Starting from the strong form in definition 2, the next step is the achievement of a suitable Discontinuous Galerkin weak formulation. Full descriptions and justifications of all the terms are present in [8] .

Let us introduce a triangulation τ_h over Ω, with $\mathcal{F}_h = \mathcal{F}_h^I \cup \mathcal{F}_h^B$ set of the faces of the elements, which includes the internal and boundary faces respectively, and the DG space $V_h^k = \{v_h \in L^2(\Omega) : v_h|_{\mathcal{K}} \in \mathbb{P}^k(\mathcal{K}) \quad \forall \mathcal{K} \in \tau_h\}$, where k is the degree of the piecewise continuous polynomial. Moreover, we define $N_h = dim(V_h^k) < \infty$.

Problem 1 (DG weak formulation). *For any $t \in [0, T]$ find $\Phi_h(t) = [\phi_i^h(t), \phi_e^h(t)]^T \in [V_h^k]^2$ and $w_h(t) \in V_h^k$ such that:*

1.

$$\sum_{K \in \tau_h} \int_K \chi_m C_m \frac{\partial V_m^h}{\partial t} V_h dw + a_i(\phi_i^h, v_h) + \sum_{K \in \tau_h} \int_K \chi_m k(V_m^h - 1)(V_m^h - a)V_m^h v_h dw +$$

$$+ \sum_{K \in \tau_h} \int_K \chi_m w_h v_h dw = (I_i^{ext}, v_h) \qquad \forall v_h \in V_h^p$$

2.

$$-\sum_{K \in \tau_h} \int_K \chi_m C_m \frac{\partial V_m^h}{\partial t} V_h dw + a_e(\phi_e^h, v_h) - \sum_{K \in \tau_h} \int_K \chi_m k(V_m^h - 1)(V_m^h - a)V_m^h v_h dw +$$

$$- \sum_{K \in \tau_h} \int_K \chi_m w_h v_h dw = (-I_e^{ext}, v_h) \qquad \forall v_h \in V_h^p$$

3.

$$\sum_{K \in \tau_h} \int_K \frac{\partial w_h}{\partial t} v_h dw = \sum_{k \in \tau_h} \int_K \epsilon(V_m^h - \gamma w_h) v_h dw \qquad \forall v_h \in V_h^p$$

where:

- $a_k(\phi_k^h, v_h) = \sum_{K \in \tau_h} \int_K (\Sigma_k \nabla_h \phi_k^h) \cdot \nabla_h v_h dw - \sum_{F \in \mathcal{F}_h^I} \int_F \{\{\Sigma_k \nabla_h \phi_k^h\}\} \cdot [[v_h]] d\sigma +$

  $- \delta \sum_{F \in \mathcal{F}_h^I} \int_F \{\{\Sigma_k \nabla_h v_h\}\} \cdot [[\phi_k^h]] d\sigma + \sum_{F \in \mathcal{F}_h^I} \int_F \gamma[[\phi_k^h]] \cdot [[v_h]] d\sigma \qquad k = i, e$

- $(I_i^{ext}, v_h) = \sum_{K \in \tau_h} \int_K I_i^{ext} v_h dw + \int_{\partial w} b_i v_h d\sigma$

- $(-I_e^{ext}, v_h) = -\sum_{K \in \tau_h} \int_K I_e^{ext} v_h dw + \int_{\partial w} b_e v_h d\sigma$

6

Moreover, according to the choice of the coefficient δ, we can define:

- *δ = 1: Symmetric Interior Penalty method (SIP)*
- *δ = 0: Incomplete Interior Penalty method (IIP)*
- *δ = −1: Non Symmetric Interior Penalty method (NIP)*

And $\gamma := \alpha \frac{k^2}{h}$ ("Stabilization parameter"), $\quad \alpha \in \mathbb{R}$ to be chosen high enough.

2.2 Algebraic formulation

Taking $\{\varphi_j\}_{j=1}^{N_h}$ base of V_h^k, so that we can write

$$\Phi_h(t) = \begin{bmatrix} \phi_i^h(t) \\ \phi_e^h(t) \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^{N_h} \phi_{i,j}(t)\varphi_j \\ \sum_{j=1}^{N_h} \phi_{e,j}(t)\varphi_j \end{bmatrix}$$

$$w_h(t) = \sum_{j=1}^{N_h} w_j(t)\varphi_j$$

$$V_m^h(t) = \sum_{j=1}^{N_h} V_{m,j}(t)\phi_j = \sum_{j=1}^{N_h} (\phi_{i,j}(t) - \phi_{e,j}(t))\varphi_j$$

Then, we introduce the matrices:

$$\left.\begin{aligned}
(V_k)_{ij} &= \int_w \nabla\varphi_j \cdot \Sigma_k \nabla\varphi_i \\
(I_k^T)_{ij} &= \sum_{F \in F_h^I} \int_F \{\{\Sigma_k \nabla\varphi_j\}\} \cdot [[\varphi_i]] \\
(I_k)_{i,j} &= \sum_{F \in F_h^I} \int_F [[\varphi_j]] \cdot \{\{\Sigma_k \nabla\varphi_i\}\} \\
(S_k)_{i,j} &= \sum_{F \in F_h^I} \int_F \gamma_k [[\varphi_j]] \cdot [[\varphi_i]]
\end{aligned}\right\} \quad \begin{aligned} A_k &= (V_k - I_k^T - \theta I_k + S_k) \\ k &= i, e \end{aligned} \tag{1}$$

$$\gamma_k|_F = (n_F^T \Sigma_k\, n_F)\,\gamma, \quad n_F \text{ outward normal vector of F}$$

$$A_i \qquad \text{Intra-cellular stiffness matrix} \tag{2}$$

$$A_e \qquad \text{Extra-cellular stiffness matrix} \tag{3}$$

$$M_{ij} = \sum_{K \in \tau_h} \int_K \varphi_j \varphi_i \qquad \text{Mass matrix} \tag{4}$$

$$C(u_h)_{ij} = \sum_{K \in \tau_h} \int_K \chi_m k(u_h - 1)(u_h - a)\varphi_j\varphi_i \qquad \text{Non-linear matrix} \tag{5}$$

$$F_k = \begin{bmatrix} F_{i,k} \\ F_{e,k} \end{bmatrix} = \begin{bmatrix} \int_w I_i^{ext}\varphi_k - \sum_{F \in F_h^B} \int_F b_i\varphi_k \\ -\int_w I_e^{ext}\varphi_k - \sum_{F \in F_h^B} \int_F b_e\varphi_k \end{bmatrix} \tag{6}$$

Therefore, our semi-discrete algebraic formulation is:

Problem 2 (DG algebraic formulation). *Find* $\Phi_h(t) = [\phi_i^h(t), \phi_e^h(t)]^T \in [V_h^k]^2$ *and* $w_h(t) \in V_h^k$ *for any* $t \in (0; T]$ *such that:*

$$\begin{cases} \chi_m Cm M \dot{V}_m^h + A_i \phi_i^h + C(V_m^h) V_m^h + \chi_m M w_h = F_i^h \\ -\chi_m Cm M \dot{V}_m^h + A_e \phi_e^h - C(V_m^h) V_m^h - \chi_m M w_h = F_e^h \\ M \dot{w}_h(t) = \epsilon M(V_m^h(t) - \gamma w_h(t)) \end{cases}$$

An alternative and more compact version with block matrices:

Problem 3 (DG algebraic formulation - 2). *Find* $\Phi_h(t) = [\phi_i^h(t), \phi_e^h(t)]^T \in [V_h^k]^2$ *and* $w_h(t) \in V_h^k$ *for any* $t \in (0; T]$ *such that:*

I

$$\chi_m C_m \begin{bmatrix} M & -M \\ -M & M \end{bmatrix} \begin{bmatrix} \dot{\phi}_i^h(t) \\ \dot{\phi}_e^h(t) \end{bmatrix} + \begin{bmatrix} A_i & 0 \\ 0 & A_e \end{bmatrix} \begin{bmatrix} \phi_i^h(t) \\ \phi_e^h(t) \end{bmatrix} +$$

$$\begin{bmatrix} C(V_m^h) & -C(V_m^h) \\ -C(V_m^h) & C(V_m^h) \end{bmatrix} \begin{bmatrix} \phi_i^h(t) \\ \phi_e^h(t) \end{bmatrix} + \chi_m \begin{bmatrix} M & 0 \\ 0 & -M \end{bmatrix} \begin{bmatrix} w_h(t) \\ w_h(t) \end{bmatrix} = \begin{bmatrix} F_i^h \\ F_e^h \end{bmatrix}$$

II

$$\dot{w}_h(t) = \epsilon(V_m^h(t) - \gamma w_h(t))$$

3 Dubiner Basis

3.1 Analytical aspects

So far, we have described a general semi-discrete discontinuous formulation without examining which basis to use to generate the V_h^k space. Usually, the common choice consists in the classical hat functions from FEM, even if they need to be modified in order to be used in a discontinuous context. It is also one of the simplest choices, for this reason our provided code was initially implemented with this basis. However, the very novelty of this study is the adoption of a new kind of basis, completely different from the previous and commonly known as "*Dubiner Basis*" [6].
How we will soon see, the peculiarity of this family of functions is that it consists of orthogonal polynomials defined on the reference triangle

$$\hat{K} = \{(\xi, \eta) : \xi, \eta \geq 0, \xi + \eta \leq 1\} \tag{7}$$

and not on the reference square

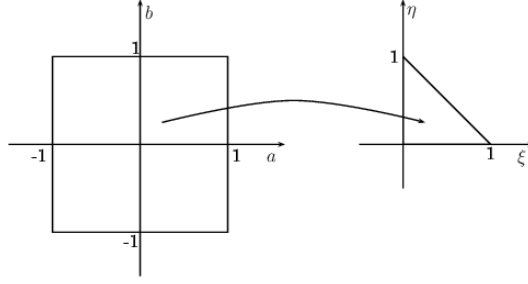$$\hat{Q} = \{(a, b) : -1 \leq a \leq 1, -1 \leq b \leq 1\} \tag{8}$$

Figure 4: Transformation between the reference square to the reference triangle

Formally, if we consider the transformation from $\hat{Q}$ to $\hat{K}$

$$\xi := \frac{(1+a)(1-b)}{4}, \eta := \frac{(1+b)}{2} \tag{9}$$

the Dubiner basis is the transformation of a suitable basis initially defined on the reference square. This initial basis is simply obtained with a two dimensional modified tensor product of the Jacobi polynomials on the interval $(-1, 1)$.

Definition 3 (Jacobi polynomials). *The Jacobi polynomials of coefficients $\alpha, \beta \in \mathbb{R}$ evaluated in $z \in (-1, 1)$ are:*

- $n = 0$
$$J_0^{\alpha,\beta}(z) = 1 \tag{10}$$

- $n = 1$
$$J_1^{\alpha,\beta}(z) = \frac{1}{2}(\alpha - \beta + (\alpha + \beta + 2) \cdot z); \tag{11}$$

- $n \geq 2$

$$
\begin{aligned}
J_n^{\alpha,\beta}(z) = \sum_{k=2}^{n} \Big[&\frac{(2k + \alpha + \beta - 1)(\alpha^2 - \beta^2)}{2k(k + \alpha + \beta)(2k + \alpha + \beta - 2)} + \\
&\frac{(2k + \alpha + \beta - 2)(2k + \alpha + \beta - 1)(2k + \alpha\beta)}{2k(k + \alpha + \beta)(2k + \alpha + \beta - 2)} J_{k-1}^{\alpha,\beta}(z) + \\
&-\frac{2(k + \alpha - 1)(k + \beta - 1)(2k + \alpha + \beta)}{2k(k + \alpha + \beta)(2k + \alpha + \beta - 2)} J_{k-2}^{\alpha,\beta}(z) \Big]
\end{aligned} \tag{12}
$$

The main property of these polynomials is:

Proposition 1. *$\{J_i^{\alpha,\beta}, i = 0, 1, 2 \ldots\}$ is orthogonal with respect to the Jacobi weight $w(x) = (1-x)^\alpha (1+x)^\beta$:*

$$\int_{-1}^{1} (1-x)^\alpha (1+x)^\beta J_m^{\alpha,\beta} J_q^{\alpha,\beta}(x) dx = \frac{2}{2m+1} \delta_{mq} \tag{13}$$

9

We can now define explicitly the Dubiner basis.

Definition 4 (Dubiner Basis)**.** *The Dubiner basis that generates the space* $\mathbb{P}^p(\hat{K})$ *of the polynomials of degree* p *over the reference triangle is the set of functions:*

$$\phi_{ij} : \hat{K} \to \mathbb{R}$$

$$\phi_{ij}(\xi, \eta) := c_{ij}(1-b)^j J_i^{0,0}(a) J_j^{2i+1,0}(b) =$$
$$= c_{ij} 2^j (1-\eta)^j J_i^{0,0}(\frac{2\xi}{1-\eta} - 1) J_j^{2i+1,0}(2\eta - 1)$$

$$(14)$$

for $i, j = 1, \ldots, p$ *and* $i + j \leq p$, *where*

$$c_{ij} := \sqrt{\frac{2(2i+1)(i+j+1)}{4^i}} \tag{15}$$

and $J_i^{\alpha, \beta}(\cdot)$ *is the i-th Jacobi polynomial*

As we have anticipated [12]

Proposition 2. *The Dubiner basis is orthonormal in* $L^2(\hat{K})$ $\forall p$:

$$\int_{\hat{K}} \phi_{ij}(\xi, \eta) \phi_{mq}(\xi, \eta) d\xi d\eta = \delta_{im}\delta_{jq} \tag{16}$$

As a consequence, after we successfully implemented the code with Dubiner basis and computed the matrices, we obtained a diagonal mass matrix (figure 5)
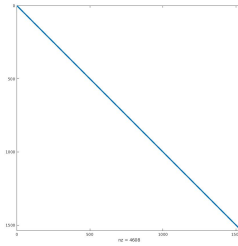

Figure 5: Non-zero elements in the mass matrix when adopting Dubiner basis

It is noteworthy to point out that transformation 9 is bijective, it can be inverted but it needs some care. The natural inverse would be:

$$a = \frac{2\xi}{1-\eta} - 1 \qquad b = 2\eta - 1 \tag{17}$$

10

that has already been used for the definition 4. However, it is not defined for $\eta = 1$, that means for the sole point $(0, 1)$ of the reference triangle. To solve this issue, it is enough to prolong the function with continuity to this special point. For the code implementation, it is suggested avoiding evaluations in the exact point or adding an *if* condition. We opted for the second solution.

In general, the orthogonality property implies some good numerical properties, not only the diagonalization of the mass matrix. For instance, in [2] interesting bounds for the conditional number can be viewed. For this reason, we opted for this choice aiming to improve the previous results, at least from the space discretization side.

On the other hand, there are also some difficulties arising when one chooses to abandon the familiar FEM basis. First of all, the coefficients of a discretized function has only *modal* meaning and they no more represent the *nodal* values of the function itself. This fact needs some extra work when one needs to switch from the continuous functions to the discretized functions and viceversa, as it will be shown in the paragraph 3.2. Secondly, one can notice that these functions are not boundary conditions friendly. What we mean is that, if compared to FEM basis, they have no particular properties on the boundary to let easily impose homogeneous boundary conditions. Thus, they should be again transformed, this time in a *boundary adapted* form. We address to [9] for a short description of this procedure. Fortunately, we do not need to set this transformation as in the discontinuous formulation boundary conditions (both Dirichlet and Neumann) are imposed only weakly. It means that the boundary conditions' choice does not imply the choice of the vectorial space as in continuous Galerkin. The discretized space is always the same, only some terms in the weak formulation have in case of need to be changed. For this reason, the match of Discontinuous Galerkin and Dubiner Basis results to be particularly successful.

To conclude, we refer to [12] for the transformation and the definition of the Dubiner Basis with tetrahedra, i.e. in dimension $n = 3$.

3.2 Implementation

Our code let the user to select which basis to adopt (FEM or Dubiner) and the order of polynomials until the order 3. We chose to call D_1, D_2, D_3 these 3 families of basis functions, thanks to the similarity to the P_1, P_2, P_3 finite element basis.
The starting point was the implementation of some functions to evaluate the Dubiner basis functions and their gradients in the quadrature points. We omit the full code as it is not particular interesting: it barely follows the definitions of section 3.1 with the addition of some technicalities. These mentioned scripts are: `eval_jacobi_polynomial.m`, `basis_legendre_dubiner.m`, `evalshape_tria_dubiner.m`.

Moreover, some conditional statements and some extra methods (as the script `matrix2D_dubiner.m`) were added to let the user easily switch from one basis to another (simply and once via `dati.m`).

More interesting are instead the scripts `dubiner_to_fem.m` and `fem_to_dubiner.m`, used to convert the Dubiner modal coefficients of the vector solution to the nodal values of the approximated function and viceversa. For this reason, they deserve some further explanations.

3.3 Switch from the modal coefficients to the nodal values

One of the many advantages of the FEM basis is that there exists a bijection between the basis functions and some particular spacial points in such a way that the evaluation of a basis function in one of these points is equal to 1 only if that point is the one associated to the function, 0 otherwise:

$$\psi_i(x_j) = \delta_{ij} \tag{18}$$

Obviously, this property does not hold when we work with Dubiner basis. Indeed, these functions are not normalized on the mesh edges and they neither have an associated mesh point. This implies that the Dubiner coefficients of a function $u \in V_h^p$ are not the evaluation over these points of the discretized function itself. They have a completely different meaning, they are now *modal* values instead of being *nodal*. For this reason we introduced two new functions that best transform the coefficients of the solution w.r.t. FEM basis to the coefficients w.r.t. Dubiner basis and viceversa.

Consider an element $\mathcal{K} \in \tau_h$ and $\{\psi_i\}_{i=1}^p, \{\phi_j\}_{j=1}^q$ as, respectively, the set of FEM functions and the set of Dubiner functions with support in $\mathcal{K}$. In addition, consider as $\{\hat{u}_i\}_{i=1}^p, \{\tilde{u}_j\}_{j=1}^q$ as, respectively, the FEM and Dubiner coefficients of the solution.

3.3.1 `dubiner_to_fem.m`

Let us start from the transformation to the FEM coefficients. We now exploit the property 18, i.e. the coefficient $\hat{u}_i$ is nothing else but the evaluation of u_h on the i-th mesh point, then:

$$\hat{u}_i = \sum_{j=1}^q \tilde{u}_j \phi_j(x_i) \tag{19}$$

where x_i is the point associated to the ψ_i basis function.

This formula has then been implemented in the following lines from the `dubiner_to_fem.m` script.

```
function [u0] = dubiner_to_fem (uh, femregion, Data)

...
...
```

```
...

u0 = zeros(femregion.ndof,1);

% loop over all the elements
for ie = 1:femregion.ne

    % to get the global indexes for the nodes of ie
    index = (ie-1)*femregion.nln*ones(femregion.nln,1) + [1:femregion.nln]';

    for i = 1 : femregion.nln
       for j = 1: femregion.nln
           u0(index(i)) = u0(index(i)) +  uh(index(j))*phi(1,i,j);
       end
    end
end
```

3.3.2   `fem_to_dubiner.m`

Instead, to compute the coefficients conversely, we need to exploit the fact that the Dubiner Basis are L^2-orthonormal (proposition 2). We then need to compute a L^2 scalar product between the FEM discretized function and each Dubiner basis function. That means:

$$\tilde{u}_j = \int_{\mathcal{K}} u_h(x)\phi_j(x)\,dx = \int_{\mathcal{K}} \sum_{i=1}^{p} \hat{u}_i\psi_i(x)\phi_j(x)\,dx = \sum_{i=1}^{p} \Big(\int_{\mathcal{K}} \psi_i(x)\phi_j(x)\,dx \Big)\hat{u}_i \quad (20)$$

This slightly more difficult formula has been reproduced in `fem_to_dubiner.m` using *Gauss-Legendre-Lobatto* quadrature integration formulas. The main steps are the following:

```
function [u0] = fem_to_dubiner (uh, femregion, Data)

...
...
...

u0 = zeros(femregion.ndof,1);

% loop over all the elements
for ie = 1:femregion.ne

    % to get the global indexes for the nodes of ie
    index = (ie-1)*femregion.nln*ones(femregion.nln,1) + [1:femregion.nln]';
```

```
    % loop over local degrees of freedom
    for i = 1 : femregion.nln
        % loop over 2D quadrature points
        for k = 1:length(w_2D)
            uh_eval_k = 0;
            % loop to evaluate uh in a quadrature point
            for j = 1:femregion.nln
                uh_eval_k = uh_eval_k + uh(index(j))*phi_fem(1,k,j);
            end
            u0(index(i)) = u0(index(i)) + uh_eval_k*phi_dub(1,k,i).*w_2D(k);
        end
    end
end
```

3.3.3   Final remarks

If the Dubiner functions are chosen as Galerkin basis, both the transformations are needed for the code implementation. Formula 19 is needed to plot and compute errors after the resolution of the system (otherwise solely Dubiner coefficients are useless). Formula 20 is instead needed to convert the FEM initial data u_0 into a vector of Dubiner coefficients before the resolution of the system.

In order to be rigorous, but also for the sake of simplicity, these transformations are implemented only from P_n to D_n, $n = 1, 2, 3$ and viceversa. With this choice, the two basis generate the same space V_h^n and then the transformation infers only on the coefficients and not on the function's properties. Otherwise, decreasing n would mean to lose significant information, while increasing n does not substantially improve the quality of the solution as it initially belonged to a lower order space. Moreover, choosing the same degree for P and D implies several semplifications, for instance the same number of local nodes (nln). For this reason, both p and q are actually replaced with nln in the code.

4 Temporal discretization

So far, we have just studied the space discretization while a temporal discretization is still needed to totally discretize the Bidomain time-dependent problem. Thus, we divide the interval (0,T] into K subintervals $(t^k, t^k + 1]$ of length Δt such that $t^k = k\Delta t \quad \forall k = 0, \cdots, K - 1$, we then assume that $V_m^k \approx V_m(t^k)$.

We have developed, implemented and tested 3 different temporal strategies that we will refer to as: *semi-implicit*, *Godunov operator-splitting* and *quasi-implicit operator-splitting*.

4.1 Semi-implicit method

One of the most famous and used temporal scheme for a non-linear problem such as the Bidomain is certainly the *Semi-Implicit* scheme [10]. The basic idea is to treat most of the terms implicitly while in fact treating the non-linear term semi-implicitly. Since the non-linear is cubic, the best choice is to treat only one of these V_m terms implicitly, i.e.:

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

at each time-step k (different from the k parameter). Moreover, the gating variable ODE is treated implicitly with the exception of the term V_m:

$$M\frac{w^{k+1} - w^k}{\Delta t} = \epsilon M(V_m^k - \gamma w^{k+1})$$

Therefore, we can transform the semi-discrete problem 3 into:

$$\chi_m C_m \begin{bmatrix} M & -M \\ -M & M \end{bmatrix} \begin{bmatrix} \frac{\phi_i^{k+1} - \phi_i^k}{\Delta t} \\ \frac{\phi_e^{k+1} - \phi_e^k}{\Delta t} \end{bmatrix} + \begin{bmatrix} A_i & 0 \\ 0 & A_e \end{bmatrix} \begin{bmatrix} \phi_i^{k+1} \\ \phi_e^{k+1} \end{bmatrix} +$$

$$\begin{bmatrix} C(V_m^h) & -C(V_m^h) \\ -C(V_m^h) & C(V_m^h) \end{bmatrix} \begin{bmatrix} \phi_i^{k+1} \\ \phi_e^{k+1} \end{bmatrix} + \chi_m \begin{bmatrix} M & 0 \\ 0 & -M \end{bmatrix} \begin{bmatrix} w^{k+1} \\ w^{k+1} \end{bmatrix} = \begin{bmatrix} F_i^{k+1} \\ F_e^{k+1} \end{bmatrix} \quad (21)$$

$$M\frac{w^{k+1} - w^k}{\Delta t} = \epsilon M(V_m^k - \gamma w^{k+1})$$

We remind that $V_m^k = \phi_i^k - \phi_e^k$. Separating known and unknown terms, we obtain:

$$\left(\frac{\chi_m C_m}{\Delta t} \begin{bmatrix} M & -M \\ -M & M \end{bmatrix} + \begin{bmatrix} A_i & 0 \\ 0 & A_e \end{bmatrix} + \begin{bmatrix} C(V_m^k) & -C(V_m^k) \\ -C(V_m^k) & C(V_m^k) \end{bmatrix} \right) \begin{bmatrix} \phi_i^{k+1} \\ \phi_e^{k+1} \end{bmatrix} =$$

$$\begin{bmatrix} F_i^{k+1} \\ F_e^{k+1} \end{bmatrix} - \chi_m \begin{bmatrix} M & 0 \\ 0 & -M \end{bmatrix} \begin{bmatrix} w^{k+1} \\ w^{k+1} \end{bmatrix} + \frac{\chi_m C_m}{\Delta t} \begin{bmatrix} M & 0 \\ 0 & -M \end{bmatrix} \begin{bmatrix} V_m^k \\ V_m^k \end{bmatrix} \quad (22)$$

$$(\frac{1}{\Delta t} + \epsilon\gamma)Mw^{k+1} = \epsilon MV_m^k + \frac{M}{\Delta t}w^k$$

If we define:

- $B = \frac{\chi_m C_m}{\Delta t} \begin{bmatrix} M & -M \\ -M & M \end{bmatrix} + \begin{bmatrix} A_i & 0 \\ 0 & A_e \end{bmatrix}$

- $C_{nl}(V_m^k) = \begin{bmatrix} C(V_m^h) & -C(V_m^h) \\ -C(V_m^h) & C(V_m^h) \end{bmatrix}$

- $r^{k+1} = \begin{bmatrix} F_i^{k+1} \\ F_e^{k+1} \end{bmatrix} - \chi_m \begin{bmatrix} M & 0 \\ 0 & -M \end{bmatrix} \begin{bmatrix} w^{k+1} \\ w^{k+1} \end{bmatrix} + \frac{\chi_m C_m}{\Delta t} \begin{bmatrix} M & -M \\ -M & M \end{bmatrix} \begin{bmatrix} \phi_i^k \\ \phi_e^k \end{bmatrix}$

we get the system in the final form:

Problem 4 (Semi-implicit discretized system). *Find* $\Phi^{k+1} = [\phi_i^{k+1} \phi_e^{k+1}]^T$ *and* w^{k+1}
$\forall k = 0, \cdots, T - 1$ *such that:*

$$\begin{cases} (\frac{1}{\Delta t} + \epsilon\gamma)Mw^{k+1} = \epsilon M V_m^k + \frac{M}{\Delta t} w^k \\ (B + C_{nl}(\Phi^k))\Phi^{k+1} = r^{k+1} \end{cases}$$

4.1.1 Implementation

```
MASS = [M -M; -M M];
ZERO = sparse(length(M), length(M));
MASS_W = [M ZERO; ZERO -M];
STIFFNESS = [Ai ZERO; ZERO  Ae];

for t=dt:dt:T

   w1 = 1/(1+epsilon*gamma*dt)*(w0+epsilon*dt*Vm0);
   w1=cat(1,w1, w1);
   Vm0 = cat(1,Vm0,Vm0);

   fi = assemble_rhs_i(femregion,neighbour,Data,t);
   fe = assemble_rhs_e(femregion,neighbour,Data,t);
   f1 = cat(1, fi, fe);

   [C] = assemble_nonlinear(femregion,Data,Vm0);
   NONLIN = [C -C; -C C];

   r = f1 + ChiM*Cm/dt * MASS_W * Vm0 - ChiM * MASS_W *w1;

   B=ChiM*Cm/dt * MASS + (STIFFNESS + NONLIN);

   u1 = B \ r;
```

```
    f0 = f1;
    Vm0 = u1(1:ll)-u1(ll+1:end);
    u0 = u1;
    w0 = w1(1:ll);
end
```

4.2   Godunov operator-splitting

The main feature of a general operator-splitting method is the division of the prob-
lem into two different problems to be solved sequentially. This is possible and justified
when the original functional operator L is splitted into 2 different operators such that
$L(u) = L_1(u) + L_2(u)$. Two operator-splitting methods have been implemented, the first
is of *Godunov* type and a detailed study together with its properties can be found in
[13]. The formulation is:

$$\text{Find } \hat{V}_m^{k+1}, \phi_i^{k+1}, \phi_e^{k+1}, w^{k+1} \text{ such that:}$$

I

$$\begin{cases} \chi_m C_m M \frac{\hat{V}_m^{k+1} - V_m^k}{\Delta t} + C(V_m^k) V_m^k + \chi_m M w^k = 0 \\ \frac{w^{k+1} - w^k}{\Delta t} = \epsilon(V_m^k - \gamma w^k) \end{cases}$$

II

$$\begin{cases} \chi_m C_m M \frac{V_m^{k+1} - \hat{V}_m^{k+1}}{\Delta t} + A_i \phi_i^{k+1} = F_i^{k+1} \\ -\chi_m C_m M \frac{V_m^{k+1} - \hat{V}_m^{k+1}}{\Delta t} + A_e \phi_e^{k+1} = F_e^{k+1} \end{cases}$$

Putting in a unique system:

$$\begin{cases} \chi_m C_m M \frac{V_m^{k+1} - V_m^k}{\Delta t} + C(V_m^k) V_m^k + \chi_m M w^k + A_i \phi_i^{k+1} = F_i^{k+1} \\ \chi_m C_m M \frac{V_m^{k+1} - V_m^k}{\Delta t} + C(V_m^k) V_m^k + \chi_m M w^k - A_e \Phi_e^{k+1} = -F_e^{k+1} \\ w^{k+1} = (1 - \epsilon\gamma\Delta t) w^k + \epsilon\Delta t V_m^k \end{cases} \quad (23)$$

The equations in the system 23 can be rewritten as:

$$\begin{cases} \left(\frac{\chi_m C_m}{\Delta t} M + A_i\right) \phi_i^{k+1} - \frac{\chi_m C_m}{\Delta t} M \phi_e^{k+1} = F_i^{k+1} - \chi_m M w^k + \left(\frac{\chi_m C_m}{\Delta t} M - C(V_m^k)\right) V_m^k \\ \frac{\chi_m C_m}{\Delta t} M \phi_i^{n+1} - \left(\frac{\chi_m C_m}{\Delta t} M + A_e\right) \phi_e^{k+1} = -F_e^{k+1} - \chi_m M w^k + \left(\frac{\chi_m C_m}{\Delta t} M - C(V_m^k)\right) V_m^k \\ w^{k+1} = (1 - \epsilon\gamma\Delta t) w^k + \epsilon\Delta t V_m^k \end{cases}$$

And then, we obtain the final form:

Problem 5 (Godunov operator-splitting discretized system). *Find* $\Phi^{k+1} = [\phi_i^{k+1} \phi_e^{k+1}]^T$ *and* w^{k+1} $\forall k = 0, \cdots, K-1$ *such that:*

$$
\begin{cases}
\left(\dfrac{\chi_m C_m}{\Delta t} \begin{bmatrix} M & -M \\ M & -M \end{bmatrix} + \begin{bmatrix} A_i & 0 \\ 0 & -A_e \end{bmatrix} \right) \begin{bmatrix} \phi_i^{k+1} \\ \phi_e^{k+1} \end{bmatrix} = \begin{bmatrix} F_i^{k+1} \\ -F_e^{k+1} \end{bmatrix} + \\[2ex]
-\chi_m \begin{bmatrix} M & 0 \\ 0 & M \end{bmatrix} \begin{bmatrix} w^k \\ w^k \end{bmatrix} + \left(\dfrac{\chi_m C_m}{\Delta t} \begin{bmatrix} M & 0 \\ 0 & M \end{bmatrix} - \begin{bmatrix} C(V_m^k) & 0 \\ 0 & C(V_m^k) \end{bmatrix} \right) \begin{bmatrix} V_m^k \\ V_m^k \end{bmatrix} \\[2ex]
w^{k+1} = (1 - \epsilon\gamma\Delta t)w^k + \epsilon\Delta t V_m^k
\end{cases}
$$

4.2.1 Implementation

```
ZERO = sparse(ll,ll);
MASS = (ChiM*Cm/dt)*[M, -M; M -M];
MASSW = ChiM*[M, ZERO; ZERO, M];

for t=dt:dt:T

   fi = assemble_rhs_i(femregion,neighbour,Data,t);
   fe = assemble_rhs_e(femregion,neighbour,Data,t);
   f1 = cat(1, fi, -fe);

   [C] = assemble_nonlinear(femregion,Data,Vm0);

   w1 = (1 -epsilon*gamma*dt)*w0 + epsilon*dt*Vm0;
   B = MASS + [Ai, ZERO; ZERO, -Ae];
   r = -MASSW*[w0;w0] + ((Cm/dt)*MASSW - [C, ZERO; ZERO, C])
   *[Vm0;Vm0] + f1;

   Vm0 = u1(1:ll) - u1(ll+1:end);
   u0 = u1;
   w0 = w1;
end
```

4.3 Quasi-implicit operator-splitting

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 full implicit method, we hope to achieve more stability if compared to the previous Godunov-kind scheme. This time, the formulation turns to be:

Find $\tilde{V}_m^{k+1}, \phi_i^{k+1}, \phi_e^{k+1}, w^{k+1}$ such that:

I

$$\begin{cases} \chi_m C_m M \frac{\tilde{V}_m^{k+1} - V_m^k}{\Delta t} + C(V_m^k)V_m^{k+1} + \chi_m M w^{k+1} = 0 \\ \frac{w^{k+1} - w^k}{\Delta t} = \epsilon(V_m^{k+1} - \gamma w^{k+1}) \end{cases}$$

II

$$\begin{cases} \chi_m C_m M \frac{V_m^{k+1} - \tilde{V}_m^{k+1}}{\Delta t} + A_i \phi_i^{k+1} = F_i^{k+1} \\ -\chi_m C_m M \frac{V_m^{k+1} - \tilde{V}_m^{k+1}}{\Delta t} + A_e \phi_e^{k+1} = F_e^{k+1} \end{cases}$$

Putting into a unique system:

$$\begin{cases} \chi_m C_m M \frac{V_m^{k+1} - V_m^k}{\Delta t} + C(V_m^k)V_m^{k+1} + \chi_m M w^{k+1} + A_i \phi_i^{k+1} = F_i^{k+1} \\ \chi_m C_m M \frac{V_m^{k+1} - V_m^k}{\Delta t} + C(V_m^k)V_m^{k+1} + \chi_m M w^{k+1} - A_e \phi_e^{k+1} = -F_e^{k+1} \\ \frac{w^{k+1} - w^k}{\Delta t} = \epsilon(V_m^{k+1} - \gamma w^{k+1}) \end{cases} \qquad (24)$$

If we define:

- $Q_k = \frac{\chi_m C_m}{\Delta t} M + C(V_m^k) + \frac{\epsilon \chi_m \Delta t}{1 + \epsilon \gamma \Delta t} M$

- $R_k = \frac{\chi_m C_m}{\Delta t} M V_m^k - \frac{\chi_m}{1 + \epsilon \gamma \Delta t} M w^k$

the equations in system 24 can be written as:

1.

$$\chi_m C_m M \frac{\phi_i^{k+1} - \phi_e^{k+1} - V_m^k}{\Delta t} + C(V_m^k)(\phi_i^{k+1} - \phi_e^{k+1}) +$$

$$+ \chi_m M \left(\frac{w^k + \epsilon \Delta t(\phi_i^{k+1} - \phi_e^{k+1})}{1 + \epsilon \gamma \Delta t} \right) + A_i \phi_i^{k+1} = F_i^{k+1}$$

$$\Rightarrow \quad (Q_k + A_i)\phi_i^{k+1} - Q_k \phi_e^{k+1} = R_k + F_i^{k+1}$$

2.

$$\chi_m C_m M \frac{\phi_i^{k+1} - \phi_e^{k+1} - V_m^k}{\Delta t} + \cdot C(V_m^k)(\phi_i^{k+1} - \phi_e^{k+1}) +$$

$$+ \chi_m M \left(\frac{w^k + \epsilon \Delta t(\phi_i^{k+1} - \phi_e^{k+1})}{1 + \epsilon \gamma \Delta t} \right) - A_e \phi_e^{k+1} = -F_e^{k+1}$$

$$\Rightarrow \quad Q_k \phi_i^{k+1} - (Q_k + A_e)\phi_e^{k+1} = R_k - F_e^{k+1}$$

19

3.

$$w^{k+1} = \frac{w^k + \epsilon\Delta t(\phi_i^{k+1} - \phi_e^{k+1})}{1 + \epsilon\gamma\Delta t}$$

The final system becomes:

Problem 6 (Quasi-implicit operator-splitting discretized system). *Find* $\Phi^{k+1} = [\phi_i^{k+1}\phi_e^{k+1}]^T$ *and* w^{k+1} $\forall k = 0, \cdots, K-1$ *such that:*

$$\begin{cases} \left(\begin{bmatrix} Q_k & -Q_k \\ Q_k & -Q_k \end{bmatrix} + \begin{bmatrix} A_i & 0 \\ 0 & -A_e \end{bmatrix} \right) \begin{bmatrix} \boldsymbol{\phi_i^{k+1}} \\ \boldsymbol{\phi_e^{k+1}} \end{bmatrix} = \begin{bmatrix} R_k \\ R_k \end{bmatrix} + \begin{bmatrix} F_i^{k+1} \\ -F_e^{k+1} \end{bmatrix} \\ \\ w^{k+1} = \dfrac{w^k + \epsilon\Delta t(\phi_i^{k+1} - \phi_e^{k+1})}{1 + \epsilon\gamma\Delta t} \end{cases}$$

4.3.1   Implementation

```
ZERO = sparse(ll,ll);

for t=dt:dt:T

    [C] = assemble_nonlinear(femregion,Data,Vm0);
    Q  = (ChiM*Cm/dt)*M + C - (epsilon*ChiM*dt)/(1+epsilon*gamma*dt)*M;
    R  = (ChiM*Cm/dt)*M*Vm0 - (ChiM)/(1+epsilon*gamma*dt)*M*w0;

    fi = assemble_rhs_i(femregion,neighbour,Data,t);
    fe = assemble_rhs_e(femregion,neighbour,Data,t);
    f1 = cat(1, fi, -fe);

    B = [Q, -Q; Q, -Q] + [Ai, ZERO; ZERO, -Ae];
    r = [R;R] + f1;

    u1 = B \ r;

    Vm1 = u1(1:ll)-u1(ll+1:end);

    w1 = (w0 + epsilon*dt*Vm1)/(1+epsilon*gamma*dt);

    f0 = f1;
    Vm0 = u1(1:ll) - u1(ll+1:end);
    u0 = u1;
    w0 = w1;
end
```

5 About the potentials uniqueness

5.1 Analytical concepts

Observing for a moment the Bidomain problem analytical formulation (definition 2), we can immediately realize that the intracellular and the extracellular potentials appear only through their difference V_m or their gradient. This means that there cannot be uniqueness for the two functions. Namely:

$$\phi_i, \phi_e \text{ classical solutions of Bidomain} \Rightarrow \phi_i + \varphi, \ \phi_e + \varphi \text{ are solutions as well}$$
$$\forall \varphi : [0, T] \to \mathbb{R} \quad \text{sufficiently regular} \tag{25}$$

However, this fact should not surprise nor confuse the reader. First of all, we remind again that in [4] and [5] there are proofs for the V_m and w uniqueness, then this is taken for granted. Secondly, this statement reflects the physical intuition of the problem: cellular dynamics is not involved by potentials exact values but instead from their difference, in addition a potential value is nonsense if a convention value to compare it with has not been set. The dependence on time can be interpreted as follows: if, at any time instant, we change the conventional potential value, the dynamics of the problem does not change.

Moreover, we can give this simple result to show that the solutions of the form of equation 25 are also the only admissible:

Theorem 1. *For the Bidomain problem coupled with Fitzhugh-Nagumo model with Neumann boundary conditions (definition 2), the ϕ_i, ϕ_e solutions are unique minus a constant depending only on time.*

Proof.   We remind that existence and uniqueness for V_m and w have already been proved in [4]. Suppose now there exist two couples $(\phi_i^1, \phi_e^1), (\phi_i^2, \phi_e^2)$ of potentials solutions of the Bidomain problem. If V_m uniqueness holds, there must exist a unique value of V_m such that:

$$\phi_i^1 - \phi_e^1 = \phi_i^2 - \phi_e^2 = V_m$$

Then, we define a function $\varphi : \Omega \times [0, T] \to \mathbb{R}$ as:

$$\varphi := \phi_i^1 - \phi_i^2 = \phi_e^1 - \phi_e^2$$

If we consider the starting problem in definition 2, the following equations must hold:

$$\begin{cases} \chi_m C_m \frac{\partial V_m}{\partial t} - \nabla \cdot (\Sigma_i \nabla \phi_i^1) + \chi_m I_{ion}(V_m, w) = I_i^{ext} & \text{in } \Omega_{mus} \times (0, T] \\ \chi_m C_m \frac{\partial V_m}{\partial t} - \nabla \cdot (\Sigma_i \nabla \phi_i^2) + \chi_m I_{ion}(V_m, w) = I_i^{ext} & \text{in } \Omega_{mus} \times (0, T] \\ \Sigma_i \nabla \phi_i^1 \cdot n = b_i & \text{on } \partial\Omega_{mus} \times (0, T] \\ \Sigma_i \nabla \phi_i^2 \cdot n = b_i & \text{on } \partial\Omega_{mus} \times (0, T] \end{cases}$$

Subtracting:

$$\begin{cases} -\nabla \cdot (\Sigma_i \nabla \varphi) = 0 & \text{in } \Omega_{mus} \times (0, T] \\ \Sigma_i \nabla \varphi \cdot n = 0 & \text{on } \partial\Omega_{mus} \times (0, T] \end{cases}$$

That is a classical *Laplace problem* with homogeneous Neumann boundary conditions. From theory [11], we know that the solution set is composed by all constant terms (remember that Σ_i is positive definite). However, we must pay attention to the fact that φ is a time-dependent function, even if time does not compare in the system. Thus, we can state:

$$\exists \tilde{\varphi} : [0, T] \to \mathbb{R} \text{ such that } \varphi(x, t) = \tilde{\varphi}(t) \quad \forall x \in \Omega, \forall t \in [0, T]$$

To conclude, we can observe now that if these two couples of solutions exist, then:

$$\phi_i^1 - \phi_i^2 = \phi_e^1 - \phi_e^2 = \tilde{\varphi} \quad \forall x \in \Omega, \forall t \in [0, T]$$

$\square$

Remark. *For what concerns the regularity of φ, we can certainly state that, as a difference of two sufficiently regular functions, it belongs to the same class of regularity of the potentials if restricted to the sole time variable.*

We can conclude this analytical digression with an accomplished necessary and sufficient condition for the potentials solutions.

Corollary 1. *Suppose the couple (ϕ_i, ϕ_e) is a classical solution of the problem of definition 2 (for a certain w). The couple $(\tilde{\phi}_i, \tilde{\phi}_e)$ of sufficiently regular real functions defined in $\Omega \times [0, T]$ is another couple solution if and only if both $\tilde{\phi}_i, \tilde{\phi}_e$ differ respectively from ϕ_i, ϕ_e for a time-dependent function φ that belongs to the union of the regularity classes of the previous functions if restricted to time variable.*

Proof. The regularity statement is trivial and already discussed. The right implication is due to the previous theorem. Finally, the left implication follows what has been shown in equation 25: it is enough to insert $\phi_i + \varphi$ and $\phi_e + \varphi$ in the Bidomain system to find out that φ disappears and the remaining system is the same as the one with ϕ_i, ϕ_e, thus solved by hypothesis. $\square$

5.2 Numerical correction

Previous analytical results are crucial for what concerns the numerical computations since the Bidomain problem turns now to be *not exactly* well-posed. Even if, in general, the right V_m is most of the times achieved thanks to its uniqueness, our aim is to impose a further condition on the ϕ unknowns for the following two reasons:

1. To pursue the achievement of the exact ϕ_i, ϕ_e and not their values minus a constant. Useful for instance for the error analysis.

2. To strengthen our Galerkin formulation that currently derives from a ill-posed problem and, for this reason, may show bad features as the generation of a ill-conditioned system or even a non-solvable system.

We firstly observe that the additional condition should be applied only to one of the potentials, for instance to ϕ_i. Indeed, the difference of the two possible solutions is φ for both intracellular and extracellular potentials. Imposing φ at each time-step implies the uniqueness imposition for both ϕ_i, ϕ_e.

The most common and simple strategies are the following:

1. Imposition of the value of the function in a specific point.

$$\phi_i(\bar{x}, t) = \varphi(t) \quad \forall t \in [0, T]$$

2. Imposition of the function mean value.

$$\int_\Omega \phi_i(x, t) \, dx = \varphi(t) \quad \forall t \in [0, T]$$

Notice that the first strategy would be useless if we keep working with an analytical and abstract weak formulation in Sobolev spaces. However, in the numerical context, we can assume certain regularities for the solution that let it makes sense and be the most common choice for numerical implementations.

As we will examine later on, it is demanding to implement the first strategy in a Dubiner context without losing the main system properties. Thus, we slightly change the first strategy and we instead opt for the imposition of a vector solution coefficient. For what concerns the FEM basis, this has the same meaning as before, provided that $\bar{x}$ is not a whatever point but a *dof* point. On the other hand, for Dubiner basis, this has a completely different and abstract meaning: we remind that this time it has the role of modal coefficient.

Consider $\{u_j\}_{j=1...N_h}$ as the list of the vector solution. As a consequence, we give the numerical version of the previous strategies:

1. Imposition of a coefficient of the vector solution.

$$u_l^k = \varphi(t^k) \quad \forall k \in \{1, K\}$$

2. Imposition of the function mean value.

$$\sum_{j=1...N_h} u_j^k \, w_j = \varphi(t^k) \quad \forall k \in \{1, K\}$$

where l is a fixed value $\in \{1, N_h\}$ and w_j stands for a suitable weight (that depends on the mesh geometry, the basis choice and the quadrature formula choice).

We remind that our aim is to impose such conditions *directly* into the system. The easiest way would certainly be to impose these conditions *after* the system resolution, as it has been reproduced in other past works. However, in this case, some issues related to the ill-posedness arise, especially ill-conditioning. Then, in the next sections we will illustrate how we managed to impose potential uniqueness only changing matrices and vectors coefficients before the resolution.

5.2.1 Implementation of the first coefficient imposition

For simplicity, we choose $l = 1$. Then, u_l^k has the meaning of:

- Value of u in the first dof point, $\forall$ timestep k (FEM)

- Fourier coefficient of u w.r.t. the first Dubiner basis function, $\forall$ timestep k (Dubiner)

What follows will be independent from basis choice. Suppose $c \in \mathbb{R}$ is the value to impose in the system $Au = \vec{b}$ for a certain timestep k. Since first coefficient occupies the first cell in the unknown vector and influences other coefficient values only through the first matrix column, we can switch the system from:

$$A = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1N} \\ a_{21} & a_{22} & \ldots & a_{2N} \\ a_{31} & a_{32} & \ldots & a_{3N} \\ \ldots & \ldots & \ldots & \ldots \\ a_{N1} & a_{N2} & \ldots & a_{NN} \end{bmatrix} \qquad b = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \ldots \\ b_N \end{bmatrix}$$

to:

$$\tilde{A} = \begin{bmatrix} 1 & 0 & \ldots & 0 \\ 0 & a_{22} & \ldots & a_{2N} \\ 0 & a_{32} & \ldots & a_{3N} \\ \ldots & \ldots & \ldots & \ldots \\ 0 & a_{N2} & \ldots & a_{NN} \end{bmatrix} \qquad \tilde{b} = \begin{bmatrix} c \\ b_2 - a_{21}c \\ b_3 - a_{31}c \\ \ldots \\ b_N - a_{N1}c \end{bmatrix}$$

This is certainly correct since in the first system line $u_1 = c$ is automatically imposed and, in the other lines, u_1 is no more treated as unknown but as a known data and then moved to the r.h.s. of the system.
The very advantage of this procedure is the conservation of A symmetry. As we have previously anticipated, we discarded the nodal value strategy because, using Dubiner basis, we would have lost this crucial property.
Moreover, the value c can be freely chosen, for instance from the exact solution (when error analysis needs to be executed) or a conventional fixed value as zero.

On the other hand, there are two main disadvantages. First of all, the system first line information has been deleted during this procedure. However, if the mesh is composed by many elements, this information is not essential and the solution behavior is practically the same as this information were provided.

Secondly, if the system is hugely ill-conditioned or even non-solvable (this is the case when we have homogeneous boundary conditions and/or homogeneous forcing terms), this imposition may have an overshooting effect that unbalances the solution. For these problems, a global imposition has to be adopted and that is the reason why we implemented the more complicated mean value imposition strategy.

The coefficient imposition procedure has been implemented in the script `assign_phi_i.m` that takes the c value from the exact solution:

```
function [A, b] = assign_phi_i (A, b, t, Data, femregion)

...
...
...

if (Data.fem(1)=='P')

   x = femregion.dof(1,1); % x-coordinate of the first dof point
   y = femregion.dof(1,2); % y-coordinate of the first dof point
   exact_coeff = eval(Data.exact_sol_i); % evaluation of the exact solution


elseif (Data.fem(1)=='D')
   x0=femregion.dof(3,1); % bottom-left corner of the first element
   y0=femregion.dof(3,2);
   h=femregion.dof(1,1)-femregion.dof(3,1); % length of the element

   exact_coeff = 0;
   index = 1;

   ...
   ...
   ...

   % the first coefficient is the L2 scalar product of uh with the first
   % basis function. To the get the right coefficient, we compute the scalar
   % product between the exact solution and the first basis function

   for k = 1:length(w_2D) % loop over 2D quadrature points
      x = x0 + h*node_2D(k,1);  %physical coords of the integration point
      y = y0 + h*node_2D(k,2);
```

```
    eval_sol = eval(Data.exact_sol_i);
    exact_coeff = exact_coeff + eval_sol*phi_dub(1,k,index).*w_2D(k);
  end

end


% we now change the system coefficients in order to impose u(1)=exact_coeff
Nh = length(b);
b = b - A(:,1)*exact_coeff;
b(1) = exact_coeff;
A(:,1) = zeros(Nh,1);
A(1,:) = zeros(1,Nh);
A(1,1) = 1;
```

To conclude this section, it is interesting to observe that the numerical imposition is done at *every* time-step. This is confirmed from previous analytical theory as the difference φ is a constant but depending on time: therefore, it is needed to fix it at every time-step.

5.2.2 An analytical motivation for the mean value imposition method

It is easy to realize that the procedure in section 5.2.1 cannot be replicated for the mean unless losing symmetry. For instance, in the case of FEM basis and zero mean, a first line full of ones would imply also a first column full of ones and thus the resolution would be compromised. We should look for a different strategy. Let us start with a simple reference problem:

Problem 7 (Reference zero-mean problem - strong form). *If Ω open, bounded and sufficiently regular set, $f \in C^0(\bar{\Omega})$, find $u \in C^2(\Omega) \cap C^1(\bar{\Omega})$ such that:*

$$\begin{cases} -\Delta u = f & in\ \Omega \\ \int_\Omega u = 0 & in\ \Omega \\ \nabla u \cdot n = 0 & on\ \partial\Omega \end{cases}$$

For our scopes, it is convenient to move to the variational formulation:

Problem 8 (Reference zero-mean problem - weak form). *If Ω open and bounded set, $f \in L^2(\Omega)$, find $u \in H^1(\Omega)$ such that:*

$$\begin{cases} \int_\Omega \nabla u \cdot \nabla v = \int_\Omega f v & \forall v \in H^(\Omega) \\ \int_\Omega u = 0 \end{cases}$$

As usual, elliptic regularity and f, Ω regularities implies that the weak solution is the classical solution as well. For this reason, let us focus only on the weak form. In addition, observe that if Ω is bounded, then $H^1(\Omega) \subset L^2(\Omega) \subset L^1(\Omega)$, therefore the second equation is justified. The next step is the study of the well-posedness.

Lemma 1. *The problem 8 admits a weak solution u if and only if the compatibility condition $\int_\Omega f = 0$ holds, in other words f is a zero-mean function. Moreover, if u exists, it is unique and it minimizes the Laplace energy functional $J(u) = \frac{1}{2} \int_\Omega |\nabla u|^2 - \int_\Omega fu$*

Proof. Consider for the moment the first equation only. It is the Laplace problem with homogeneous Neumann boundary conditions. In a more general form, it is equivalent to a specific reaction-diffusion problem:

$$\begin{cases} -\Delta u + \alpha u = f & \text{in } \Omega \\ \nabla u \cdot n = 0 & \text{on } \partial\Omega \end{cases}$$

$$\text{with } \alpha = 0$$

We have already discussed that $\alpha = 0$ is an eigenvalue of the Laplace operator with Neumann boundary conditions and its eigenspace is composed by all and only constant terms. From theorem 7.1.14 in [7], we can state that, since α belongs to the spectrum and $f \in L^2(\Omega)$, existence of the weak solution holds if and only if the compatibility condition:

$$\int_\Omega f = \int_{\partial\Omega} g = \int_{\partial_\Omega} 0 = 0$$

holds. Then we solved the point about existence of the weak solution.

For what concerns the uniqueness, we know from the same theorem that u is unique except for other functions that differ from u for an eigenfunction associated to $\alpha = 0$. Since the eigenfunctions of zero are the functions that are constant *a.e.*, we can state that the weak solution is unique minus a constant term. Then, if we add the second equation $\int_\Omega u = 0$, we achieve existence and uniqueness of the solution.

Suppose now u is the weak solution and v another function $\in H^1(\Omega)$. Then:

$$\exists w \in H^1(\Omega): \quad v = w + u$$

$$\Rightarrow J(v) = J(u+w) = \frac{1}{2} \int_\Omega |\nabla u + \nabla w|^2 - \int_\Omega fu - \int_\Omega fw =$$

$$= \overbrace{\frac{1}{2} \int_\Omega |\nabla u|^2 - \int_\Omega fu}^{J(u)} + \frac{1}{2} \int_\Omega |\nabla w|^2 + \overbrace{\int_\Omega \nabla u \cdot \nabla w - \int_\Omega fw}^{=0,\ \text{by def of weak solution}} =$$

$$= J(u) + \frac{1}{2} \int_\Omega |\nabla w|^2 \geq J(u)$$

$$\square$$

Remark. *The minimization of the functional J for the Laplace problem is a known fact. However, in this case where the sole Laplace-Neumann problem is not well-posed, this result was not trivial and thus it needed a check. Indeed, it is noteworthy to underline that minimization property holds but in a slightly different way: u is not the absolute minimum point, every $u + k, k \in \mathbb{R}$ reaches the same minimum.*

Previous well-posedness and minimization results implies that, if u solves problem 8, then it is unique and, moreover, it is the unique zero mean value function that minimizes the functional $J(u)$. Thus, we can transform problem 8 in another formulation:

Problem 9 (Reference mean-value problem - 2). *Find $u \in H^1(\Omega)$ such that*

$$\begin{cases} J(u) = \min_{v \in H^1(\Omega)} J(v) \\ I(u) = 0 \end{cases}$$

where $f \in L^2(\Omega)$ and

- *$J(u) = \frac{1}{2} \int_\Omega |\nabla u|^2 - \int_\Omega f u$*
- *$I(u) = \int_\Omega u$*

Surely a solution of problem 8 is a solution of problem 9. The converse implication could be easily proved too. Then, the two problems are well-posed and completely equivalent. The advantage of the second form is that it consists in a minimization problem with constraints, a kind of problem that can be solved with generalized *Lagrange Multipliers*. It means that:

$$\exists \lambda \in \mathbb{R} \text{ such that } \quad <J'(u), v> + \lambda <I'(u), v> = 0 \quad \forall v \in H^1(\Omega)$$

where J', I' are the *Frechét derivatives* of the two operators J, I and $< \cdot, \cdot >$ represents the H^1 duality. Computing the derivatives, we indeed obtain:

$$\exists \lambda \in \mathbb{R} \text{ such that } \quad \int_\Omega \nabla u \cdot \nabla v + \lambda \int_\Omega v = \int_\Omega f v \quad \forall v \in H^1(\Omega)$$

We can then formulate a third and last version of the reference problem:

Problem 10 (Reference mean-value problem - 3). *Find $u \in H^1(\Omega), \lambda \in \mathbb{R}$ such that:*

$$\begin{cases} \int_\Omega \nabla u \cdot \nabla v + \lambda \int_\Omega v = \int_\Omega f v \quad \forall v \in H^1(\Omega) \\ \int_\Omega u = 0 \end{cases}$$

It may seem a very trivial result but actually it will be the very essence of our mean-value imposition strategy. First of all, let us check that existence and uniqueness properties have been conserved.

Lemma 2. *Suppose previous assumptions on data are satisfied. Then there exists a couple solution (u, λ) to the problem 10. Moreover, $\lambda = 0$ and u is unique.*

Proof.   For what concerns existence, we can immediately realize that the solution u of problem 8 solves the problem 10 with $\lambda = 0$. Then the existence property holds because existence of problem 8 has already been proved.

Suppose now there exist two couples $(u_1, 0), (u_2, \lambda)$ solutions of the problem and define $\varphi = u_2 - u_1$. Then:

$$\begin{cases} \int_\Omega \nabla u_1 \cdot \nabla v = \int_\Omega fv & \forall v \in H^1(\Omega) \\ \int_\Omega \nabla u_2 \cdot \nabla v + \lambda \int_\Omega v = \int_\Omega fv & \forall v \in H^1(\Omega) \\ \int_\Omega u_1 = \int_\Omega u_2 = 0 \end{cases}$$

Subtracting:

$$\begin{cases} \int_\Omega \nabla \varphi \cdot \nabla v + \lambda \int_\Omega v = 0 & \forall v \in H^1(\Omega) \\ \int_\Omega \varphi = 0 \end{cases}$$

If we assign $v = \varphi \in H^1(\Omega)$, then:

$$\begin{cases} \int_\Omega |\nabla \varphi|^2 + \lambda \int_\Omega \varphi = \int_\Omega |\nabla \varphi|^2 = 0 & \forall v \in H^1(\Omega) \\ \int_\Omega \varphi = 0 \end{cases}$$

Then, $\|\nabla \varphi\|_{L^2} = 0$ implies φ constant. But, since it has zero mean, $\varphi = 0$ a.e.
To conclude, if $u_1 = u_2$ a.e. as just proved, the previous system becomes:

$$\lambda \int_\Omega v = 0 \quad \forall v \in H^1(\Omega)$$

that trivially implies $\lambda = 0$

$\square$

This analytical digression was intended as a clarification of how problem 10 can be considered as equivalent to the original problem 8 (indeed, they are both well-posed and have the same solution). For this reason, the system modifications of the next section will be in some way justified by the previous results even if Bidomain problem is hugely more complicated than the simple Laplace problem.

5.2.3   Implementation of the mean value imposition

Following the problem 10 new formulation, the basic idea is to consider λ as a new coefficient of the vector solution, for instance the last one. The vector u is now of

dimension $N_h + 1$. Let us define $d_i = \int_\Omega \varphi_i$ where φ_i is the i-th basis function (both for FEM and Dubiner). Moreover, define c as the imposed value for the mean. Then the discretized problem at a certain time-step turns to be:

Problem 11 (Discretized mean-value imposition problem). *Find $\{u_i\}_{i=1...N_h+1}$ such that:*

$$
\begin{cases}
\displaystyle\sum_{i=1}^{N_h} u_i \int_\Omega \nabla \varphi_i \cdot \nabla \varphi_j + \lambda \, d_j = \int_\Omega f \phi_j & \forall j = 1 \ldots N_h \\[3mm]
\displaystyle\sum_{i=1}^{N_h} u_i \, d_i = c
\end{cases}
$$

Reminding that $\lambda = u_{N_h+1}$, the previous problem consists in the system transformation from:

$$
A = \begin{bmatrix}
a_{11} & a_{12} & \ldots & a_{1N} \\
a_{21} & a_{22} & \ldots & a_{2N} \\
\ldots & \ldots & \ldots & \ldots \\
a_{N1} & a_{N2} & \ldots & a_{NN}
\end{bmatrix}
\qquad
b = \begin{bmatrix}
b_1 \\
b_2 \\
\ldots \\
b_N
\end{bmatrix}
$$

to:

$$
\tilde{A} = \begin{bmatrix}
a_{11} & a_{12} & \ldots & a_{1N} & d_1 \\
a_{21} & a_{22} & \ldots & a_{2N} & d_2 \\
\ldots & \ldots & \ldots & \ldots & \ldots \\
a_{N1} & a_{N2} & \ldots & a_{NN} & d_N \\
d_1 & d_2 & \ldots & d_N & 0
\end{bmatrix}
\qquad
\tilde{b} = \begin{bmatrix}
b_1 \\
b_2 \\
\ldots \\
b_N \\
c
\end{bmatrix}
$$

First of all, observe that symmetry is conserved. Moreover, this time no line has been deleted, all the information is conserved. To conclude, we remind from lemma 2 that λ is a fake unknown, its value is always zero and, if it is not, there is some issue.

The implementation of such transformation is located in the `assign_null_average.m` script. Some lines follow:

```
function [A, b] = assign_null_average (A, b, Data, femregion)

ll = length(b)/2;

if (Data.fem(1)=='P')

   ...
   ...
   ...

   for k = 1:length(w_2D)
```

```
         coeff = coeff + phi_fem(1,k,1).*w_2D(k);
    end

    for i = 1:ll
        A(i,2*ll+1)=coeff;
        A(2*ll+1,i)=coeff;
    end


elseif (Data.fem(1)=='D')

    ...
    ...
    ...

    for p = 1:femregion.nln
        p_int = 0;
        for k = 1:length(w_2D)
            p_int = p_int + phi_dub(1,k,p).*w_2D(k);
        end
        coeff(p)=p_int;
    end

    for i = 1:femregion.nln:ll
        A(2*ll+1,i:i+femregion.nln-1)=coeff';
        A(i:i+femregion.nln-1,2*ll+1)=coeff;
    end

end

b = [b;0];
```

Some comments:

- `Nh = length(b)/2` because the original system is a block matrix system. We remind that this transformation concerns only ϕ_i, then it applies to the first half of the system only.

- c is chosen to be zero, its value does not come from an exact solution because the mean-value strategy has been adopted only for realistic simulation with no exact solutions (see 5.2.4).

- We avoided to compute all d_i for FEM basis as they all have the same value.

- On the other hand, for Dubiner basis, d_i values are different. However, it is not needed to compute these values for all the global polynomials as they repeat for every element. For this reason, we only iterate over the local degrees of freedom.

5.2.4 Results

As already discussed, the mean-value imposition was implemented and adopted only for very ill-conditioned systems. For all other cases, the coefficient imposition worked perfectly. This is why, in our research, we chose to adopt:

- the coefficient imposition for error analysis simulations (as boundary conditions and forcing terms were never homogeneous)

- the mean value imposition for realistic simulations (as boundary conditions and forcing terms were essentially homogeneous)

In conclusion, we successfully imposed the potentials uniqueness directly into the system. For what concerns our two initial objectives, we achieved both them since:

- Error analysis studies demonstrated that ϕ_i, ϕ_e potentials converge to the exact potentials as V_m and w do.

- Conditional numbers turn to be considerably decreased. Our measures proved that they passed from $\approx 10^{17}$ to $\approx 10^7$ for both the strategies.

References

[1] F. Andreotti and D. Uboldi. *Discontinuous Galerkin approximation of the monodomain problem.* Politecnico di Milano, 2021.

[2] P. F. Antonietti and P. Houston. "A Class of Domain decomposition Preconditioners for hp-Discontinuous Galerkin Finite Element Methods". In: *Journal of Scientific Computing* 46 (2011), pp. 124–149.

[3] M. Bagnara. *The Inverse Potential Problem of Electrocardiography Regularized with Optimal Control.* Politecnico di Milano, 2020.

[4] Y. Bourgault, Y. Coudière, and C. Pierre. "Existence and uniqueness of the solution for the bidomain model used in cardiac electrophysiology". In: *Nonlinear Analysis: Real World Applications* 10 (2009), pp. 458–482.

[5] P. Colli Franzone, L. F. Pavarino, and S. Scacchi. *Mathematical Cardiac Electrophysiology.* Cham: Springer-Verlag, 2014.

[6] M. Dubiner. "Spectral Methods on Triangles and Other Domains". In: *Journal of Scientific Computing* 6 (1991), pp. 345–390.

[7] A. Ferrero, F. Gazzola, and M. Zanotti. *Elementi di analisi superiore per la fisica e l'ingegneria.* Bologna: Società editrice Esculapio, 2015.

[8] L. Marta and M. Perego. *Discontinuous Galerkin approximation of the bidomain system for cardiac electrophysiology.* Politecnico di Milano, 2021.

[9] A. Quarteroni. *Modellistica Numerica per Problemi Differenziali.* Milan: Springer-Verlag, 2016.

[10] A. Quarteroni, A. Manzoni, and C. Vergara. "The cardiovascular system: Mathematical modelling, numerical algorithms and clinical applications". In: *Acta Numerica* (2017), pp. 365–590.

[11] S. Salsa. *Equazioni a derivate parziali.* Milan: Springer-Verlag, 2016.

[12] S. J. Sherwin and G. E. Karniadakis. "A new triangular and tetrahedral basis for high-order finite element methods". In: *Internation 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.