POLITECNICO
MILANO 1863

Course of Numerical Analysis for Partial Differential Equations

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

Authors: Federica Botta, Matteo Calafà

Supervisors: Christian Vergara, Paola Antonietti

A.Y. 2020/2021

Contents

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 human heart.
This work is basically the continuation of a two-years-long study carried out by three past course projects ([4], [2], [10]). In particular, the broad goal of this project is to improve the results obtained in [10] (Marta and Perego) for the Bidomain model. Indeed, we provide further analysis to go deeper in the study of stability and convergence of a *Discontinuous Galerkin* discretization and to assess the reliability of this method for more realistic scenarios. As a matter of fact, this work is primarily based on these provided data and MATLAB© codes.

1.1 The physical problem

We provide in what follows a brief introduction to the Bidomain equations. For a more complete explanation, we instead refer to [12].
The mechanical contraction and expansion of the human heart has its origin in the *electrical activation* of the cardiac cells. At each heartbeat, myocytes are activated and deactivated following a characteristic electrical cycle (Fig. 1).

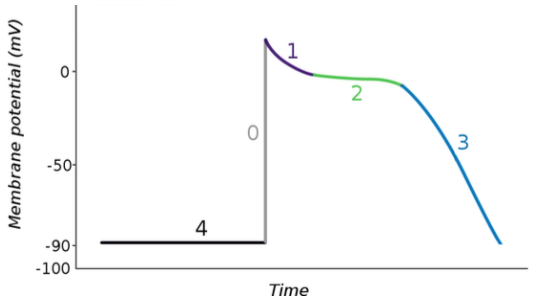


Figure 1: Membrane potential in function of time (one cardiac cycle)

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; cf. Fig. 1. 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 describe 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.

1.2 Mathematical models

Starting from the circuit in Figure 2 and applying general electromagnetism laws, the Bidomain model has been formulated (see [12] for more details and/or [7] for the complete derivation).

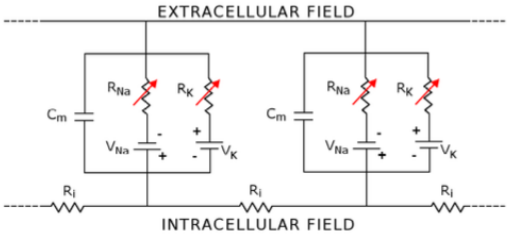


Figure 2: Simplified circuit to model the intracellular and extracellular potential dynamics

The general formulation is the following:

Problem 1 (Bidomain model)**.** *Find* ϕ_i *and* ϕ_e *such that:*

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

where:

- ϕ_i, ϕ_e are the *Intracellular and Extracellular 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:

Problem 2. *[Bidomain + FitzHugh-Nagumo model with Neumann boundary conditions] Find ϕ_i and ϕ_e such that:*

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

where:

- 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.

Moreover, if we sum the first two equations of Problem 2, we integrate both sides over Ω and we use the divergence theorem, we easily get the following compatibility condition, necessary for the existence of the solution:

Compatibility condition

$$\int_\Omega I_i^{ext} - \int_\Omega I_e^{ext} = -\int_{\partial\Omega} b_i - \int_{\partial\Omega} b_e \tag{1}$$

Even when it is not explicitly declared, boundary conditions and forcing terms are always chosen to satisfy equation (1).

From now on, the system of equations given by Problem 2 will be the reference analytical problem for the development of the forthcoming 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 [2] instead of analyzing directly the Bidomain model reported.

1.3 State of the art

As we have already introduced, our project initially aimed to continue and improve the work of a previous project [10].
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 (fact that was already noticed in [2]). This issue heavily limits the use of the MATLAB code for research and/or experimental simulations as it guarantees convergence to the right solution only in few and non-realistic problems.

2 Semi-discrete Discontinuous Galerkin formulation

2.1 Discontinuous Galerkin discrete formulation

Starting from the strong form given by the Problem 2, the next step is the achievement of a suitable Discontinuous Galerkin semi-discrete formulation. Full descriptions and justifications of all the terms are present in [10] .
Let us introduce a triangulation τ_h over Ω, where $\mathcal{F}_h = \mathcal{F}_h^I \cup \mathcal{F}_h^B$ are the set of the faces of the partition, which includes the internal ($\mathcal{F}_h^I$) and boundary ($\mathcal{F}_h^B$) faces. Let the DG space be defined as $V_h^p = \{v_h \in L^2(\Omega) : v_h|_{\mathcal{K}} \in \mathbb{P}^p(\mathcal{K}) \quad \forall \mathcal{K} \in \tau_h\}$, where p is the degree of the piecewise continuous polynomial, i.e. $p \geq 1$. Moreover, we define $N_h = dim(V_h^p) < \infty$.

Problem 3 (DG weak formulation)**.** *For any $t \in [0,T]$ find $\mathbf{\Phi_h}(t) = [\phi_i^h(t), \phi_e^h(t)]^T \in [V_h^p]^2$ and $w_h(t) \in V_h^p$ such that:*

$$
\begin{cases}
\displaystyle\sum_{\mathcal{K}\in\tau_h}\int_{\mathcal{K}} \chi_m C_m \frac{\partial V_m^h}{\partial t} v_h d\omega + a_i(\phi_i^h, v_h) + \sum_{\mathcal{K}\in\tau_h}\int_{\mathcal{K}} \chi_m k(V_m^h - 1)(V_m^h - a)V_m^h v_h d\omega + \\
\qquad + \displaystyle\sum_{\mathcal{K}\in\tau_h}\int_{\mathcal{K}} \chi_m w_h v_h d\omega = (I_i^{ext}, v_h), \qquad \forall v_h \in V_h^p, \\
\\
\displaystyle -\sum_{\mathcal{K}\in\tau_h}\int_{\mathcal{K}} \chi_m C_m \frac{\partial V_m^h}{\partial t} v_h d\omega + a_e(\phi_e^h, v_h) - \sum_{\mathcal{K}\in\tau_h}\int_{\mathcal{K}} \chi_m k(V_m^h - 1)(V_m^h - a)V_m^h v_h d\omega + \\
\qquad - \displaystyle\sum_{\mathcal{K}\in\tau_h}\int_{\mathcal{K}} \chi_m w_h v_h d\omega = (-I_e^{ext}, v_h), \qquad \forall v_h \in V_h^p, \\
\\
\displaystyle\sum_{\mathcal{K}\in\tau_h}\int_{\mathcal{K}} \frac{\partial w_h}{\partial t} v_h d\omega = \sum_{\mathcal{K}\in\tau_h}\int_{\mathcal{K}} \epsilon(V_m^h - \gamma w_h) v_h d\omega, \qquad \forall v_h \in V_h^p,
\end{cases}
$$

where:

- $a_l(\phi_l^h, v_h) = \displaystyle\sum_{\mathcal{K}\in\tau_h}\int_{\mathcal{K}} (\Sigma_l \nabla_h \phi_k^h)\cdot\nabla_h v_h d\omega - \sum_{F\in\mathcal{F}_h^I}\int_F \{\{\Sigma_l \nabla_h \phi_l^h\}\}\cdot[[v_h]] d\sigma +$
  $- \delta \displaystyle\sum_{F\in\mathcal{F}_h^I}\int_F \{\{\Sigma_l \nabla_h v_h\}\}\cdot[[\phi_l^h]] d\sigma + \sum_{F\in\mathcal{F}_h^I}\int_F \Gamma[[\phi_l^h]]\cdot[[v_h]] d\sigma \qquad l = i, e,$ (2)
- $(I_i^{ext}, v_h) = \displaystyle\sum_{\mathcal{K}\in\tau_h}\int_{\mathcal{K}} I_i^{ext} v_h d\omega + \int_{\partial\Omega} b_i v_h d\sigma,$
- $(-I_e^{ext}, v_h) = -\displaystyle\sum_{\mathcal{K}\in\tau_h}\int_{\mathcal{K}} I_e^{ext} v_h d\omega + \int_{\partial\Omega} b_e v_h d\sigma.$

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

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

In equation (2), Γ is the so called stability parameter, which is defined edge-wise as: $\Gamma := \alpha\frac{p^2}{h}$, where $\alpha \in \mathbb{R}$ has to be chosen high enough for the SIP and IIP formulations.

2.2 Algebraic formulation

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

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

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

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

Where $\phi_{i,j}$, $\phi_{e,j}$ and $w_j \in \mathbb{R}$ are the unknown expansion coefficients $\forall i,j = 1,\dots N_h$. Then, we introduce the matrices:

$$\left.\begin{aligned}(V_l)_{ij} &= \textstyle\int_\Omega \nabla\varphi_j \cdot \Sigma_l \nabla\varphi_i,\\ (I_l^T)_{ij} &= \textstyle\sum_{F\in\mathcal{F}_h^I}\int_F \{\{\Sigma_l\nabla\varphi_j\}\}\cdot[[\varphi_i]],\\ (I_l)_{i,j} &= \textstyle\sum_{F\in\mathcal{F}_h^I}\int_F [[\varphi_j]]\cdot\{\{\Sigma_l\nabla\varphi_i\}\},\\ (S_l)_{i,j} &= \textstyle\sum_{F\in\mathcal{F}_h^I}\int_F \Gamma_l[[\varphi_j]]\cdot[[\varphi_i]],\end{aligned}\right\} \quad \begin{aligned}A_l &= (V_l - I_l^T - \theta I_l + S_l)\\ l &= i,e,\end{aligned}$$

$$\Gamma_l|_F = (n_F^T\,\Sigma_l\,n_F)\,\Gamma, \quad \text{with } n_F \text{ outward normal vector of F},$$

We also define:

$$A_i \in \mathbb{R}^{N_h\times N_h} \qquad \text{Intra-cellular stiffness matrix},$$

$$A_e \in \mathbb{R}^{N_h\times N_h} \qquad \text{Extra-cellular stiffness matrix},$$

$$M_{ij} = \sum_{\mathcal{K}\in\tau_h}\int_{\mathcal{K}}\varphi_j\varphi_i \qquad \text{Mass matrix},$$

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

$$F_{i,j}^h = \int_\Omega I_i^{ext}\varphi_j + \sum_{F\in\mathcal{F}_h^B}\int_F b_i\varphi_j \qquad \text{Intra-cellular forcing term},$$

$$F^h_{e,j} = -\int_\Omega I^{ext}_e \varphi_j + \sum_{F\in\mathcal{F}^B_h}\int_F b_e\varphi_j \qquad \text{Extra-cellular forcing term,}$$

Therefore, our semi-discrete algebraic formulation is as follows:

Problem 4 (DG algebraic formulation). *Find* $\mathbf{\Phi_h}(t) = [\phi^h_i(t), \phi^h_e(t)]^T \in [V^p_h]^2$ *and* $w_h(t) \in V^p_h$ *for any* $t \in (0;T]$ *such that:*

$$\begin{cases} \chi_m C_m M \dot{V^h_m} + A_i\phi^h_i + C(V^h_m)V^h_m + \chi_m M w_h = F^h_i, \\ -\chi_m C_m M \dot{V^h_m} + A_e\phi^h_e - C(V^h_m)V^h_m - \chi_m M w_h = F^h_e, \\ M\dot{w}_h(t) = \epsilon M(V^h_m(t) - \gamma w_h(t)), \end{cases}$$

Supplemented with suitable initial conditions.
An alternative and more compact version is given by:

Problem 5 (DG algebraic formulation - 2). *Find* $\mathbf{\Phi_h}(t) = [\phi^h_i(t), \phi^h_e(t)]^T \in [V^p_h]^2$ *and* $w_h(t) \in V^p_h$ *for any* $t \in (0;T]$ *such that:*

$$\begin{cases} \chi_m C_m \begin{bmatrix} M & -M \\ -M & M \end{bmatrix} \begin{bmatrix} \dot{\phi^h_i}(t) \\ \dot{\phi^h_e}(t) \end{bmatrix} + \begin{bmatrix} A_i & 0 \\ 0 & A_e \end{bmatrix} \begin{bmatrix} \phi^h_i(t) \\ \phi^h_e(t) \end{bmatrix} + \begin{bmatrix} C(V^h_m) & -C(V^h_m) \\ -C(V^h_m) & C(V^h_m) \end{bmatrix} \begin{bmatrix} \phi^h_i(t) \\ \phi^h_e(t) \end{bmatrix} + \\ \qquad\qquad + \chi_m \begin{bmatrix} M & 0 \\ 0 & -M \end{bmatrix} \begin{bmatrix} w_h(t) \\ w_h(t) \end{bmatrix} = \begin{bmatrix} F^h_i \\ F^h_e \end{bmatrix}, \\ \dot{w}_h(t) = \epsilon(V^h_m(t) - \gamma w_h(t)). \end{cases}$$

3 High-order Dubiner basis functions

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^p space. Usually, the common choice consists in the classical hat functions from FEM. 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*"[8], which is well suited to high-order approximations.
How we will see soon, the peculiarity of this family of functions is that it consists of orthogonal polynomials defined on the reference triangle

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

and not on the reference square

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

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

$$\xi = \frac{(1+a)(1-b)}{4}, \quad \eta = \frac{(1+b)}{2}, \tag{3}$$

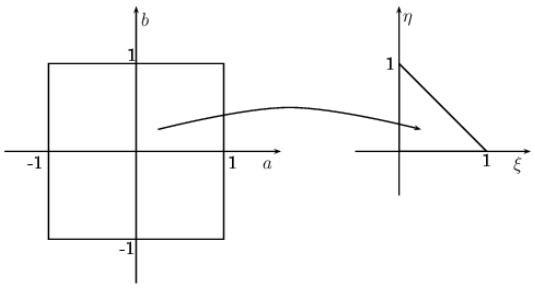


Figure 3: Transformation between the reference square ($\hat{Q}$) to the reference triangle ($\hat{K}$)

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 $(-1, 1)$, as described in the following definition.

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

- $n = 0$

$$J_0^{\alpha,\beta}(z) = 1,$$

- $n = 1$

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

- $n \geq 2$

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

The main property of these polynomials is the following:

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

$$\int_{-1}^{1} (1-x)^\alpha (1+x)^\beta J_m^{\alpha,\beta}(x) J_q^{\alpha,\beta}(x) dx = \frac{2}{2m+1} \delta_{mq} \qquad \forall i, j, m, q \geq 0.$$

We can now define explicitly the Dubiner basis.

Definition 2 (Dubiner basis)**.** *The Dubiner basis that generates the space $\mathbb{P}^p(\hat{K})$ of polynomials of degree p over the reference triangle is the set of functions:*

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

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

for $i, j = 0, \dots, p$ and $i + j \leq p$, where

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

and $J_i^{\alpha,\beta}(\cdot)$ is the i-th Jacobi polynomial defined in Def. 1.

As we have anticipated, the following result holds, cf. [14].

Proposition 2. *The Dubiner basis is orthonormal in $L^2(\hat{K})$ $\forall p \geq 0$:*

$$\int_{\hat{K}} \phi_{ij}(\xi, \eta) \phi_{mq}(\xi, \eta) d\xi d\eta = \delta_{im} \delta_{jq} \qquad \forall i, j, m, q \geq 0.$$

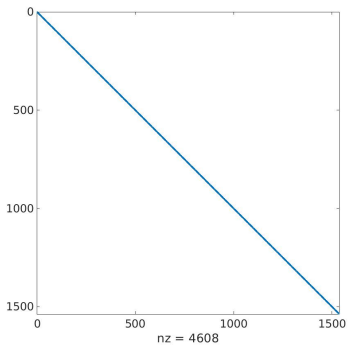


Figure 4: Non-zero elements in the mass matrix when adopting Dubiner basis

Notice that, thanks to Proposition 2, the mass matrix of the DG space turns out to be diagonal. See Fig. 4 as a confirmation.
It is noteworthy to point out that transformation 3 is bijective, it can be inverted but it needs some care. The natural inverse would be:

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

that has already been used for Definition 2. However, a is not defined for $\eta = 1$, i.e. for the sole point $(0, 1)$ of the reference triangle. Moreover, in [14] it is shown that a value cannot be achieved with continuity prolongation because of a second type singularity. Even so, the transformation is still effective because it can be easily shown that the evaluations of the Dubiner basis functions in $(\xi, \eta) = (0, 1)$ are the same for every value of a from -1 to 1. For the code implementation, we bypassed this issue with some conditional statements in which we set a the conventional value -1. Another solution would be to avoid the evaluation in the exact point choosing instead a near point.

In general, the orthogonality property implies some good numerical qualities, not only the diagonalization of the mass matrix. For instance, in [3] interesting bounds for the conditional number have been proved. 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 Section 3.3. 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 [11] for a short de-

scription of this procedure. Fortunately, we do not need to set this transformation as in the discontinuous formulation Dirichlet boundary conditions are imposed only weakly. It means that the boundary conditions' choice does not imply the choice of the vectorial space as in continuous Galerkin methods. 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 [14] for the transformation and the definition of the Dubiner basis with tetrahedra, i.e. in dimension $n = 3$.

3.2 Implementation

Our MATLAB code allows the user to select which basis to adopt (FEM or Dubiner) and the order of polynomials until p=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 functions.
As explained in Section 1.3, our starting point was the code implemented by the previous projects, namely the resolution of the Bidomain Model through FEM basis (P_1, P_2, P_3). As follows, our first goal was the implementation of methods to evaluate the Dubiner basis functions (D_1, D_2, D_3) 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 scripts are: `eval_jacobi_polynomial.m`, `basis_legendre_dubiner.m`, `evalshape_tria_dubiner.m`.

Moreover, some conditional statements and some extra methods (as `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 expansion coefficients to the Lagrangian ones

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 and 0 otherwise, i.e.:

$$\psi_i(x_j) = \delta_{ij}. \tag{4}$$

Obviously, this property does not hold when we work with the Dubiner basis. Indeed, these functions are not associated to any spacial point. This implies that the Dubiner coefficients of a function $u \in V_h^p$ are not the evaluation at suitable 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 transform the coefficients of the solution w.r.t. the FEM basis to the coefficients w.r.t. the Dubiner basis and viceversa.

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

3.3.1 From Dubiner basis to FEM basis

Let us start from the transformation to the FEM coefficients. We now exploit the Property 4, i.e. the coefficient $\hat{u}_i$ is nothing else but the evaluation of u_h on the i-th degree-of-freedom point, then:

$$\hat{u}_i = \sum_{j=1}^{q} \tilde{u}_j \varphi_j(x_i), \qquad i = 1, \dots p \tag{5}$$

where x_i is the point associated to the ψ_i basis function.
This formula has been implemented in `dubiner_to_fem.m` script (Section 8.1).

3.3.2 From FEM basis to Dubiner basis

Instead, to compute the coefficients conversely, we need to exploit the fact that the Dubiner basis is 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 is:

$$\tilde{u}_j = \int_{\mathcal{K}} u_h(x)\varphi_j(x)\,dx = \int_{\mathcal{K}} \sum_{i=1}^{p} \hat{u}_i \psi_i(x)\varphi_j(x)\,dx = \sum_{i=1}^{p} \Big(\int_{\mathcal{K}} \psi_i(x)\varphi_j(x)\,dx \Big) \hat{u}_i \tag{6}$$
$$j = 1, \dots q$$

This slightly more difficult formula has been reproduced in `fem_to_dubiner.m` (Section 8.2) using *Gauss-Legendre* quadrature formulas.

3.3.3 Final remarks

If the Dubiner functions are chosen as Galerkin basis, both the transformations are needed for the code implementation. Formula (5) is needed to plot and compute errors after the resolution of the system. Formula (6) 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, where P_n stands for the FEM basis of n-th polynomial degree, meanwhile D_n is the Dubiner basis of n-th polynomial degree. With this choice, the two bases generate the same space V_h^p and then the transformation infers only on the coefficients without modifying the functions. 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 simplifications, 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 N subintervals $(t^n, t^n + 1]$ of length Δt such that $t^n = n\Delta t \quad \forall n = 0, \cdots, N-1$, we then assume our fully discretized solution as $V_m^n \approx V_m^h(t^n)$.
Starting from a *semi-implicit* scheme, we implemented and tested 2 further temporal strategies that we will refer to as: *Godunov operator-splitting* and *quasi-implicit operator-splitting*.

4.1 The 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 [12]. 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.:

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

at each time-step $n+1$. Moreover, the gating variable ODE is treated implicitly with the exception of the term V_m:

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

Therefore, we can transform the semi-discrete Problem 5 into:

$$\begin{cases} \chi_m C_m \begin{bmatrix} M & -M \\ -M & M \end{bmatrix} \begin{bmatrix} \frac{\phi_i^{n+1} - \phi_i^n}{\Delta t} \\ \frac{\phi_e^{n+1} - \phi_e^n}{\Delta t} \end{bmatrix} + \begin{bmatrix} A_i & 0 \\ 0 & A_e \end{bmatrix} \begin{bmatrix} \phi_i^{n+1} \\ \phi_e^{n+1} \end{bmatrix} + \\ \begin{bmatrix} C(V_m^h) & -C(V_m^h) \\ -C(V_m^h) & C(V_m^h) \end{bmatrix} \begin{bmatrix} \phi_i^{n+1} \\ \phi_e^{n+1} \end{bmatrix} + \chi_m \begin{bmatrix} M & 0 \\ 0 & -M \end{bmatrix} \begin{bmatrix} w^{n+1} \\ w^{n+1} \end{bmatrix} = \begin{bmatrix} F_i^{n+1} \\ F_e^{n+1} \end{bmatrix}, \\ M\frac{w^{n+1} - w^n}{\Delta t} = \epsilon M(V_m^n - \gamma w^{n+1}). \end{cases}$$

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

$$\begin{cases} \left(\frac{\chi_m C_m}{\Delta t} \begin{bmatrix} M & -M \\ -M & M \end{bmatrix} + \begin{bmatrix} A_i & 0 \\ 0 & A_e \end{bmatrix} + \begin{bmatrix} C(V_m^n) & -C(V_m^n) \\ -C(V_m^n) & C(V_m^n) \end{bmatrix} \right) \begin{bmatrix} \phi_i^{n+1} \\ \phi_e^{n+1} \end{bmatrix} = \\ \begin{bmatrix} F_i^{n+1} \\ F_e^{n+1} \end{bmatrix} - \chi_m \begin{bmatrix} M & 0 \\ 0 & -M \end{bmatrix} \begin{bmatrix} w^{n+1} \\ w^{n+1} \end{bmatrix} + \frac{\chi_m C_m}{\Delta t} \begin{bmatrix} M & 0 \\ 0 & -M \end{bmatrix} \begin{bmatrix} V_m^n \\ V_m^n \end{bmatrix}, \\ (\frac{1}{\Delta t} + \epsilon\gamma) M w^{n+1} = \epsilon M V_m^n + \frac{M}{\Delta t} w^n. \end{cases}$$

If we define:

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

we get the system in its final form.

Problem 6 (Semi-implicit discretized system)**.** *Find* $\mathbf{\Phi^{n+1}} = [\phi_i^{n+1}\phi_e^{n+1}]^T$ *and* w^{n+1} $\forall n = 0, \cdots, N-1$ *such that:*

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

The implementation can be found at Section 8.3.

4.2 The Godunov operator-splitting

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 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 [15]. The formulation is:

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

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

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

Putting into a unique system:

$$\begin{cases} \chi_m C_m M\frac{V_m^{n+1} - V_m^n}{\Delta t} + C(V_m^n)V_m^n + \chi_m M w^n + A_i\phi_i^{n+1} = F_i^{n+1}, \\ \chi_m C_m M\frac{V_m^{n+1} - V_m^n}{\Delta t} + C(V_m^n)V_m^n + \chi_m M w^n - A_e\phi_e^{n+1} = -F_e^{n+1}, \\ w^{n+1} = (1 - \epsilon\gamma\Delta t)w^n + \epsilon\Delta t V_m^n. \end{cases} \tag{7}$$

The equations in system (7) can be rewritten as:

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

Then, we obtain the final form:

Problem 7 (Godunov operator-splitting discretized system)**.** *Find* $\mathbf{\Phi^{n+1}} = [\phi_i^{n+1} \phi_e^{n+1}]^T$ *and* $w^{n+1} \quad \forall n = 0, \cdots, N-1$ *such that:*

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

The implementation is written at Section 8.4.

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

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

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

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

Putting into a unique system, we obtain:

$$\begin{cases} \chi_m C_m M \frac{V_m^{n+1}-V_m^n}{\Delta t} + C(V_m^n)V_m^{n+1} + \chi_m M w^{n+1} + A_i \phi_i^{n+1} = F_i^{n+1}, \\ \chi_m C_m M \frac{V_m^{n+1}-V_m^n}{\Delta t} + C(V_m^n)V_m^{n+1} + \chi_m M w^{n+1} - A_e \phi_e^{n+1} = -F_e^{n+1}, \\ \frac{w^{n+1}-w^n}{\Delta t} = \epsilon(V_m^{n+1} - \gamma w^{n+1}). \end{cases} \tag{8}$$

If we define:

- $Q_n = \frac{\chi_m C_m}{\Delta t} M + C(V_m^n) + \frac{\epsilon \chi_m \Delta t}{1+\epsilon\gamma\Delta t} M$,
- $R_n = \frac{\chi_m C_m}{\Delta t} M V_m^n - \frac{\chi_m}{1+\epsilon\gamma\Delta t} M w^n$,

the equations in system (8) can be written as:

$$\begin{aligned} &\chi_m C_m M \frac{\phi_i^{n+1} - \phi_e^{n+1} - V_m^n}{\Delta t} + C(V_m^n)(\phi_i^{n+1} - \phi_e^{n+1}) + \\ &+\chi_m M \left(\frac{w^n + \epsilon \Delta t (\phi_i^{n+1} - \phi_e^{n+1})}{1 + \epsilon\gamma\Delta t} \right) + A_i \phi_i^{n+1} = F_i^{n+1}, \end{aligned} \tag{9}$$

$$\Rightarrow \quad (Q_n + A_i)\phi_i^{n+1} - Q_n \phi_e^{n+1} = R_n + F_i^{n+1},$$

$$\begin{aligned} &\chi_m C_m M \frac{\phi_i^{n+1} - \phi_e^{n+1} - V_m^n}{\Delta t} + \cdot C(V_m^n)(\phi_i^{n+1} - \phi_e^{n+1}) + \\ &+\chi_m M \left(\frac{w^n + \epsilon \Delta t (\phi_i^{n+1} - \phi_e^{n+1})}{1 + \epsilon\gamma\Delta t} \right) - A_e \phi_e^{n+1} = -F_e^{n+1}, \end{aligned} \tag{10}$$

$$\Rightarrow \quad Q_n \phi_i^{n+1} - (Q_n + A_e)\phi_e^{n+1} = R_n - F_e^{n+1},$$

$$w^{n+1} = \frac{w^n + \epsilon \Delta t (\phi_i^{n+1} - \phi_e^{n+1})}{1 + \epsilon\gamma\Delta t}. \tag{11}$$

The final system becomes:

Problem 8 (Quasi-implicit operator-splitting discretized system). *Find* $\mathbf{\Phi^{n+1}} = [\phi_i^{n+1} \phi_e^{n+1}]^T$ *and* $w^{n+1} \quad \forall n = 0, \cdots, N-1$ *such that:*

$$\begin{cases} \left(\begin{bmatrix} Q_n & -Q_n \\ Q_n & -Q_n \end{bmatrix} + \begin{bmatrix} A_i & 0 \\ 0 & -A_e \end{bmatrix} \right) \begin{bmatrix} \phi_i^{n+1} \\ \phi_e^{n+1} \end{bmatrix} = \begin{bmatrix} R_n \\ R_n \end{bmatrix} + \begin{bmatrix} F_i^{n+1} \\ -F_e^{n+1} \end{bmatrix}, \\ \\ w^{n+1} = \frac{w^n + \epsilon \Delta t (\phi_i^{n+1} - \phi_e^{n+1})}{1 + \epsilon\gamma\Delta t}. \end{cases}$$

The implementation is shown at Section 8.5.

5 About uniqueness of the potentials

5.1 Analytical concepts

Examining for a moment the Bidomain problem analytical formulation (Problem 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:

$$\begin{gathered}\phi_i, \phi_e \text{ classical solutions of Bidomain} \Rightarrow \phi_i + \varphi,\ \phi_e + \varphi \text{ are solutions as well} \\ \forall \varphi : [0,T] \to \mathbb{R} \text{ sufficiently regular.}\end{gathered} \tag{12}$$

However, this fact should not surprise nor confuse the reader. First of all, we remind again that in [5] and [7] 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 (12) are also the only admissible:

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

Proof. We remind that existence and uniqueness for V_m and w have already been proved in [5]. Suppose now there exist two couples (ϕ_i^1, ϕ_e^1),(ϕ_i^2, ϕ_e^2) of potentials solutions of the Bidomain problem. If V_m is unique, then 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 Problem 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 the first two equations and the second two we obtain:

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

That is a classical *Laplace problem* with homogeneous Neumann boundary conditions. From [13], 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:

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

□

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 Problem 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 (12): 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. □

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 if we adopt the standard space H^1. 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_i, ϕ_e unknowns for the following two reasons:

1. To pursue the achievement of the exact potentials and not their values up to a constant. Useful for instance for the error analysis.

2. To strengthen our Galerkin formulation that currently derives from a problem that is well-posed only in $H^1\backslash\mathbb{R}$ and, for this reason, may show bad features, as the generation of a ill-conditioned system or even an indeterminate 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, therefore 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 Lagrangian hat functions, 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 $\{u_j\}_{j=1\ldots N_h}$ 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.

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

2. Imposition of the function mean value.

$$\sum_{j=1\ldots N_h} u_j^n\, w_j = \varphi(t^n) \quad \forall n \in \{1, N\},$$

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 chosen).

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_1^n has the meaning of:

- Value of u in the first dof point, $\forall$ timestep n (in the case of FEM basis)
- Fourier coefficient of u w.r.t. the first Dubiner basis function, $\forall$ timestep n (in the case of Dubiner basis)

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 n. 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:

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

to:

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

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 initial system is hugely ill-conditioned or even non-solvable (determinant could approximate the machine epsilon when we have homogeneous boundary conditions and/or no forcing terms), this imposition may have an overshooting effect that unbalances the solution. For these problems, a global imposition has to be adopted and this 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` (Section 8.6) that takes the c value from the exact solution.
To conclude this section, we can also 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 9 (Reference zero-mean problem - strong form)**.** *Let Ω be an open, bounded and sufficiently regular domain, $f \in C^0(\bar{\Omega})$. Find $u \in C^2(\Omega) \cap C^1(\bar{\Omega})$ such that:*

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

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

Problem 10 (Reference zero-mean problem - weak form)**.** *Let be an Ω open and bounded set, $f \in L^2(\Omega)$. Find $u \in H^1(\Omega)$ such that:*

$$\begin{cases} \displaystyle\int_\Omega \nabla u \cdot \nabla v = \int_\Omega f v, & \forall v \in H^1(\Omega), \\ \displaystyle\int_\Omega u = 0. \end{cases}$$

As usual, the regularity of f and Ω imply 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 10 admits a unique weak solution u if and only if the compatibility condition $\int_\Omega f = 0$ holds, in other words if f is a zero-mean function. Moreover, u 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 [9], 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 up to 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:

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

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

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

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

□

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 $u + \xi, \xi \in \mathbb{R}$ reaches the same minimum.*

Previous well-posedness and minimization results implies that, if u solves Problem 10, 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 10 in another formulation:

Problem 11 (Reference mean-value problem - 2). *Find $u \in H^1(\Omega)$ such that*

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

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

- $J(u) = \frac{1}{2}\int_\Omega |\nabla u|^2 - \int_\Omega fu$,
- $I(u) = \int_\Omega u$.

Corollary 2. *Problem 10 and Problem 11 are both well-posed and share the same unique solution $u \in H^1$. Thus, they are equivalent problems.*

Proof. From Lemma 1, we know that Problem 10 is well-posed and its unique solution u solves Problem 11 too. Let us show that u is the unique solution of Problem 11 too. Consider $v \in H^1$ another different solution and define $w = u - v \in H^1$.
From the end of Lemma 1 proof, since $J(u) = J(v)$ by hypothesis, it follows that:

$$\int_\Omega |\nabla w|^2 = 0$$

Since $\nabla w \in L^2(\Omega)$, it implies that $\nabla w = 0$ almost everywhere. From *Microteorema 2.2* in [13] about weak gradients, w is constant almost everywhere. But, since $I(w) = I(u) - I(v) = 0$, w is necessarily equal to zero a.e.
Summing up, u always exists and is the unique solution of both the problems.
□

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:

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

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

Problem 12 (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 that the assumptions on data of Problem 10 are satisfied. Then there exists a couple solution* (u, λ) *to Problem 12 and* u *is the same solution of Problems 10 and 11. Moreover,* $\lambda = 0$ *and* u *is unique.*

Proof. For what concerns existence, we can immediately realize that the solution u of Problem 10 solves the Problem 12 with $\lambda = 0$. Then the existence property holds because existence of Problem 10 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:

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

Subtracting, we obtain:

$$\begin{cases} \int_\Omega \nabla \varphi \cdot \nabla v + \lambda \int_\Omega v = 0, \quad \forall v \in H^1(\Omega), \\ \int_\Omega \varphi = 0. \end{cases} \tag{13}$$

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

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

$$\Rightarrow \int_\Omega |\nabla \varphi|^2 + \lambda \int_\Omega \varphi = \int_\Omega |\nabla \varphi|^2 = 0, \quad \forall v \in H^1(\Omega).$$

Then, $\|\nabla \varphi\|_{L^2} = 0$ implies φ constant a.e., 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 (13) becomes:

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

that trivially implies $\lambda = 0$.

□

This analytical digression was intended as a clarification of how Problem 12 can be considered as equivalent to the original Problem 10 (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 12 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 $d_i = \int_\Omega \psi_i$ where ψ_i is the i-th basis function (whether FEM or Dubiner basis). Moreover, define c as the imposed value for the mean. Then the discretized problem at a certain time-step turns to be:

Problem 13 (Discretized mean-value imposition problem)**.** *Find* $\{u_i\}_{i=1\dots N_h+1}$ *such that:*

$$\begin{cases} \displaystyle\sum_{i=1}^{N_h} u_i \int_\Omega \nabla\psi_i \cdot \nabla\psi_j + \lambda\, d_j = \int_\Omega f\psi_j, \quad \forall j = 1 \dots N_h, \\ \displaystyle\sum_{i=1}^{N_h} u_i\, d_i = c. \end{cases}$$

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

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

to:

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

First of all, observe that symmetry is conserved. Moreover, this time no line has been deleted, hence all the information is conserved. For this reason, a similar procedure might be replicated for the first coefficient imposition if the information of the first row and first column are essential. On the other hand, this method is clearly more expensive since it requires the computation of many terms that later fill the matrix. To conclude, we remind from Lemma 2 that λ is an auxiliary unknown, so its value turns out to be always zero.

The implementation of such transformation is carried out in the `assign_null_average.m` script (at Section 8.7).
Some comments:

- `Nh = length(b)/2` at line 3 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 (line 42), its value does not come from an exact solution because the mean-value strategy has been adopted only for realistic simulation where no exact solutions is provided (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 Final remarks

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

- the coefficient imposition for error analysis simulations in Section 6.1 (as boundary conditions and forcing terms were never homogeneous)
- the mean value imposition for realistic simulations in Section 6.2 (as boundary conditions and forcing terms were essentially homogeneous)

6 Numerical results

6.1 Space error analysis

6.1.1 Chosen data

Considering some example problems, we provide now an experimental error analysis that can show the efficacy and goodness of our numerical schemes. For all the simulations, we choose the parameters proposed in [4] and used in [2] and [10]. These values are reported in Table 1.

Table 1: Parameters for space error analysis simulations

Domain (m)	$\Omega = (0,1)^2$
dt (s)	0.0001
T (s)	0.001
χ_m (m^{-1})	10^5
Σ_i (Sm^{-1})	$\begin{bmatrix} 0.12 & 0 \\ 0 & 0.12 \end{bmatrix}$
Σ_e (Sm^{-1})	$\begin{bmatrix} 0.12 & 0 \\ 0 & 0.12 \end{bmatrix}$
C_m (Fm^{-2})	10^{-2}
k	19.5
ε	1.2
γ	0.1
a	$13 \cdot 10^{-3}$

We choose as exact solutions:

$$V_m = \sin(2\pi x)\sin(2\pi y)e^{-5t},$$
$$w = \frac{\varepsilon}{\varepsilon\gamma - 5}\sin(2\pi x)\sin(2\pi y)e^{-5t}.$$

From these assumptions, we compute the r.h.s., the boundary conditions and initial conditions accordingly. Moreover, we remind from Section 5.2.4 that the coefficient imposition is always chosen for the following results. When it is not explicitly declared, $D1$ (Dubiner basis with $p = 1$ as polynomial degree) is chosen as polynomials space and the semi-implicit method is chosen for the time discretization. To conclude, the element size is split in half 5 times with refinement levels from 2 to 6 for all the following plots.

6.1.2 Comparison between Dubiner and FEM basis

At first, an error analysis related to the chosen basis is shown. More precisely, we fix a polynomial order ($p = 1, 2$ and 3) and we compare the errors of the Dubiner and Lagrangian hat basis functions choosing the same polynomial order for both. We expect to see similar results. On the other hand, we expect to see different error orders for different polynomials orders.

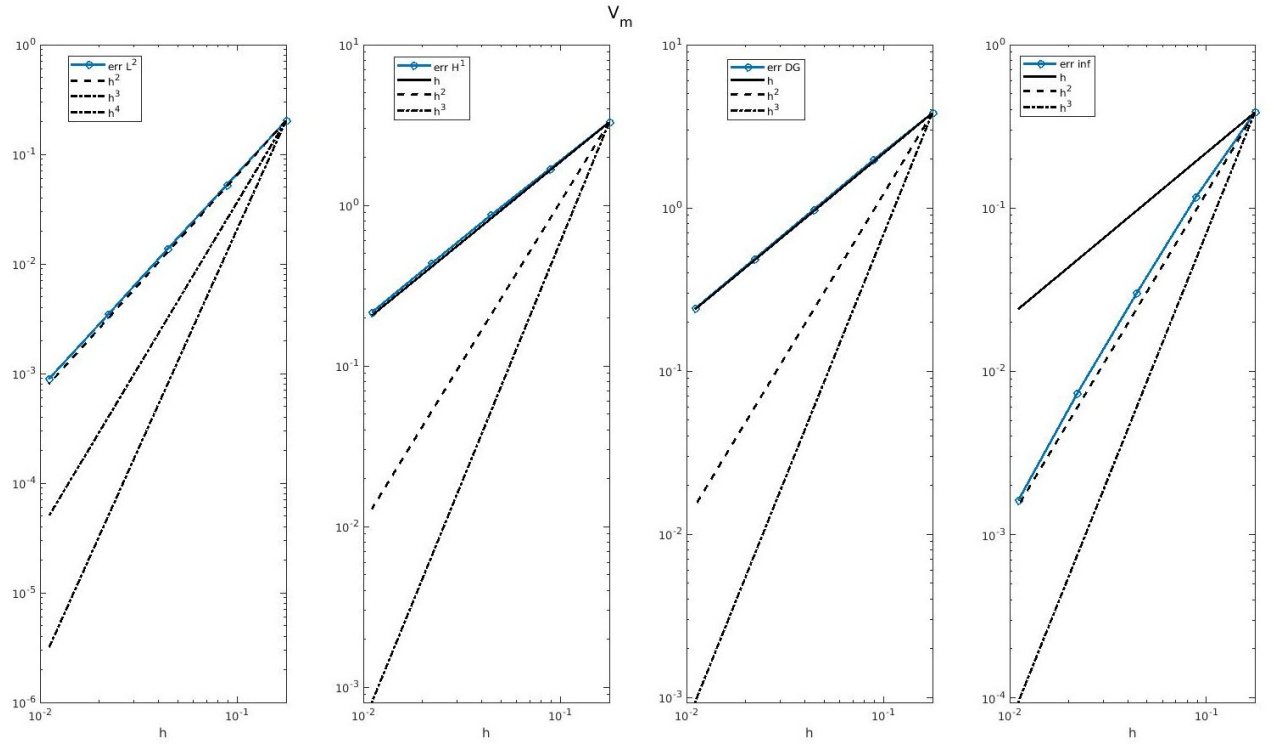
P1-D1 If $p = 1$ is chosen as polynomial order, the computed errors plots are shown in Fig. 5, Fig. 6, Fig. 7 and Fig. 8. It is clear that the convergence is very similar since the errors between the two kinds of basis have the same trend. Then, in this case, results are exactly as expected and show a first order for H^1 and DG errors while a second order for L^2 and L^∞ errors.

P2-D2 For what concerns the second order polynomials, i.e. $p = 2$, we observe slightly different results, see Fig. 9, Fig. 10, Fig. 11 and Fig. 12.
Indeed, we notice two unexpected facts: first of all ϕ_i and ϕ_e errors trends are not identical when we adopt the two bases. However, this is not a huge inconsistency as it is simply due to the uniqueness imposition that have different effects for Dubiner and FEM basis. Indeed, the differences are visible only in the L^2 and L^∞ errors and, moreover, these differences disappear when potentials subtract to get V_m.
The second behavior we see is the flatter segment that some errors trends have for small mesh sizes. This is probably due to the influence of other cause of errors, especially the time-discretization errors, when the space discretization errors become very small. This is the reason why this effect was not present in the previous case, where space errors were still too big and other causes of error negligible.
We then underline that these two effects are not due to the Dubiner discretization implementation itself. The first difference is due to the coefficient imposition that can be improved with a mean value imposition as seen in Section 5.2. The second fact is due to time discretization errors that can be reduced if we simply reduce the time-step. Moreover, if this happens, it means that space discrezitation errors are very small (in general, a positive fact). This is why we neglect these effects for the error order estimations and we can state that Dubiner method goodness keeps intact.
We observe, before the plateau, a quadratic convergence rate for the error measure in the energy norm and a cubic convergence rates for the error measure in the L^2 and L^∞ norms.

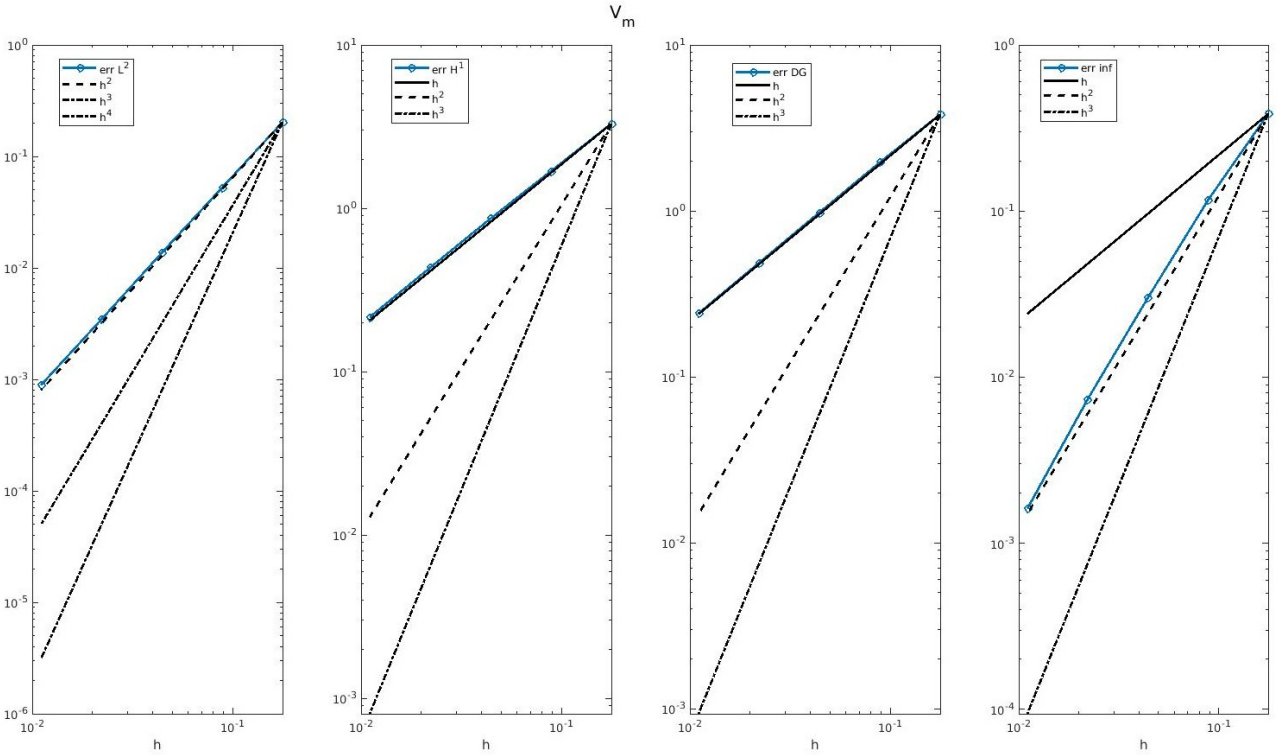
P3-D3 Finally, for polynomials of degree 3, we still find the expected convergence rate except for the two phenomena already discussed in Section 6.1.2. The results are shown in Fig. 13, Fig. 14, Fig. 15 and Fig. 16. However, because of the third order precision, space errors are smaller and then these effects are amplified. If we do not consider them, we still get the expected orders that are third order for H^1 and DG errors and fourth order for L^2 and L^∞ norms.

Computed errors for Dubiner and FEM with first order polynomials

Figure 5: Comparison of the trans-membrane potential (V_m)

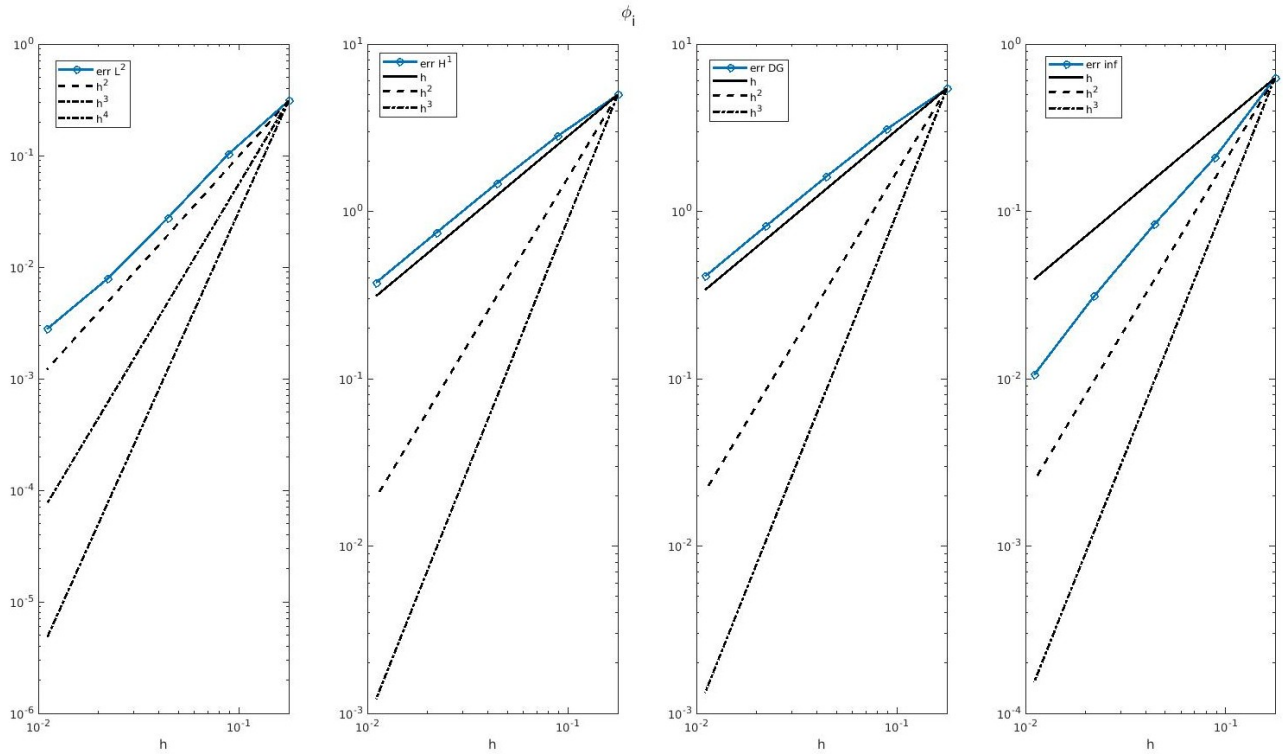


(a) Trans-membrane potential (V_m) with $p = 1$ Dubiner basis

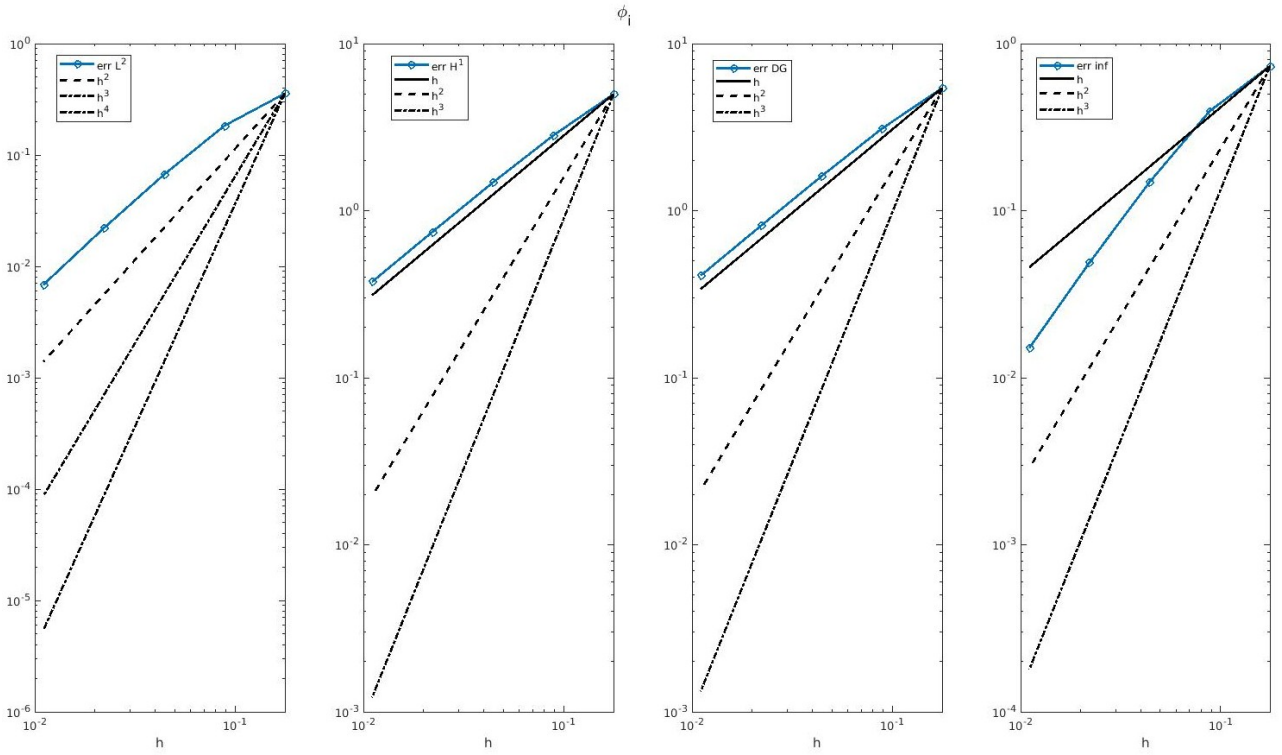


(b) Trans-membrane potential (V_m) with $p = 1$ Lagrangian hat functions

Figure 6: Comparison of the intracellular potential (ϕ_i)

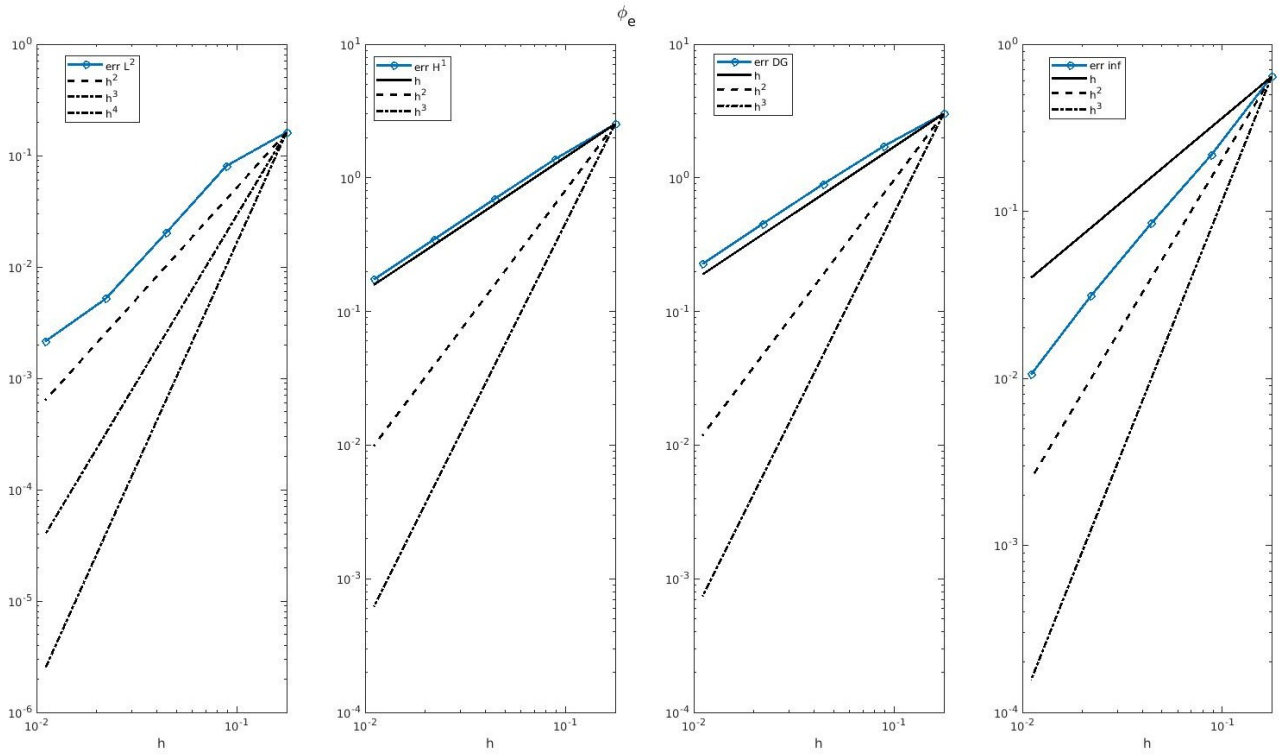


(a) Intracellular potential (ϕ_i) with $p = 1$ Dubiner basis

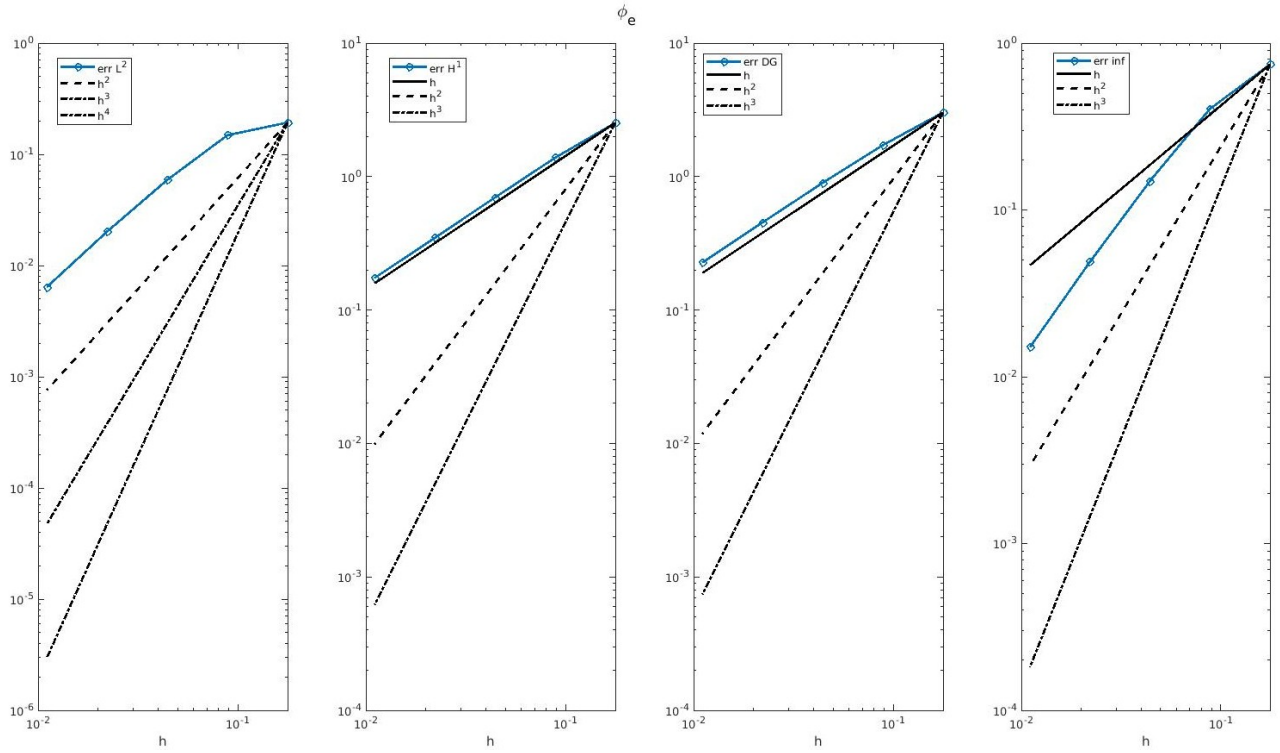


(b) Intracellular potential (ϕ_i) with $p = 1$ Lagrangian hat functions

Figure 7: Comparison of the extracellular potential (ϕ_e)

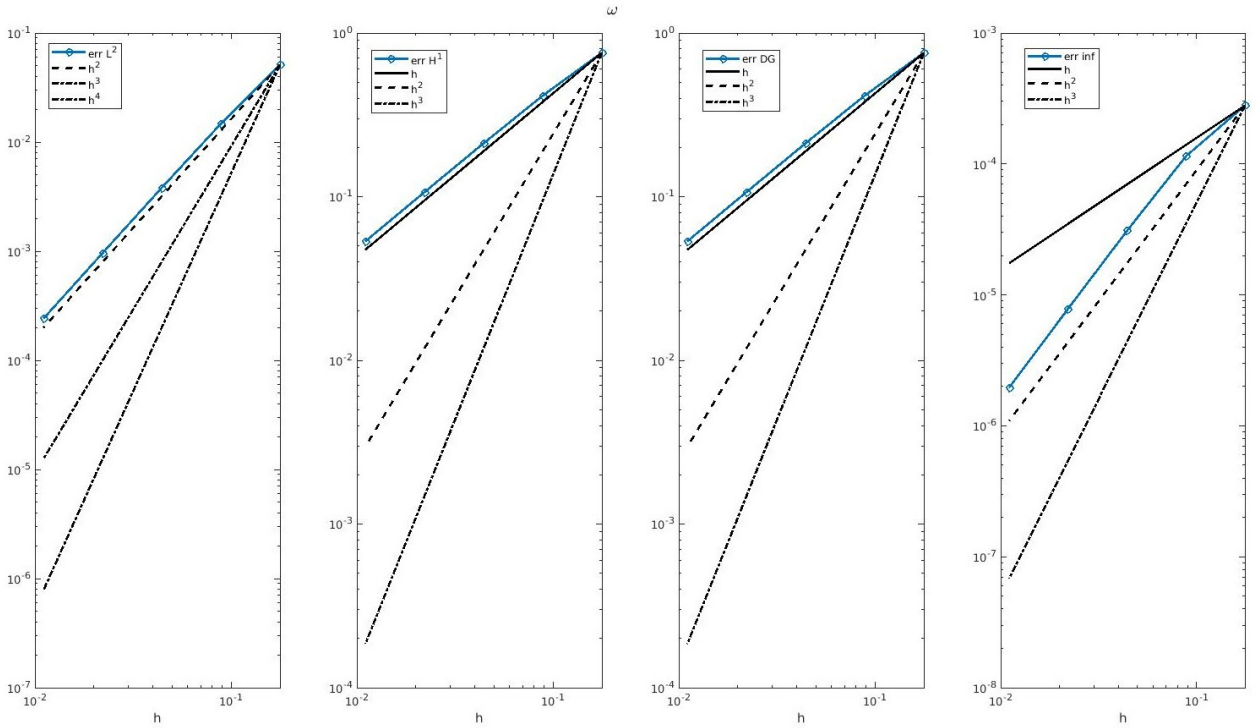


(a) Extracellular potential (ϕ_e) with $p = 1$ Dubiner basis

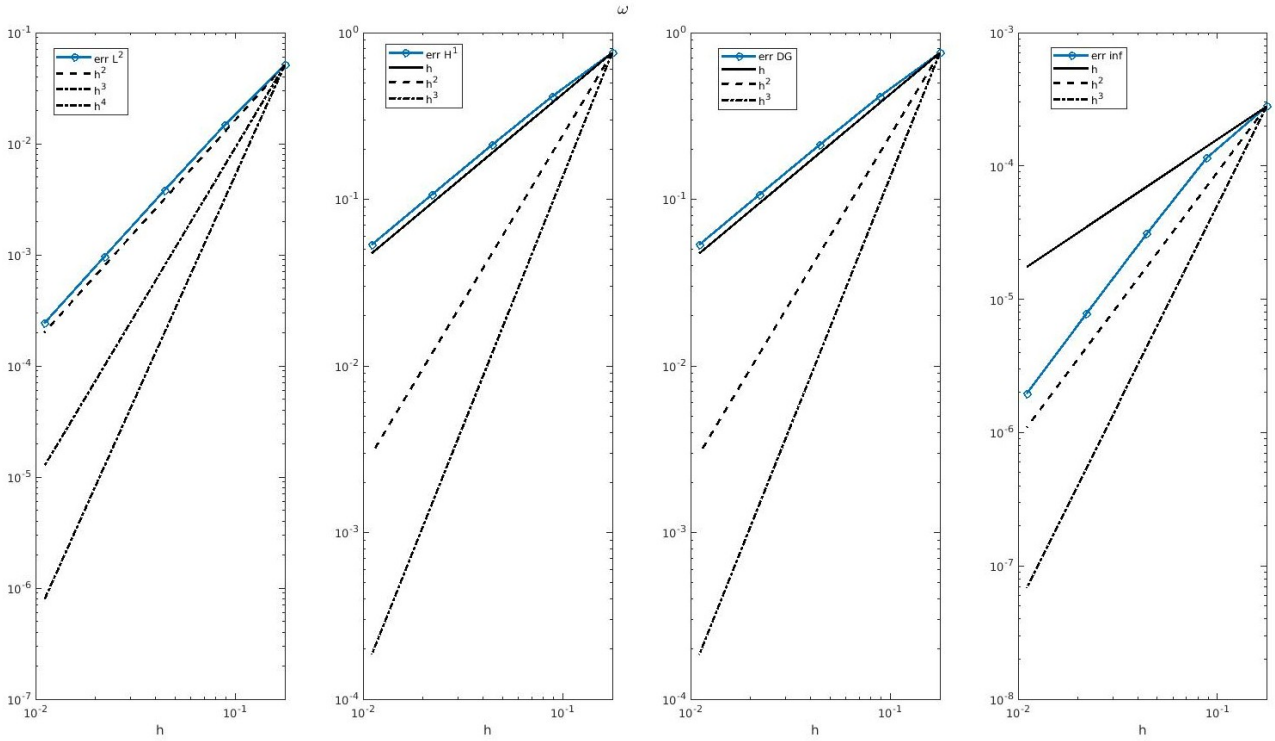


(b) Extracellular potential (ϕ_e) with $p = 1$ Lagrangian hat functions

Figure 8: Comparison of the gating variable (w)



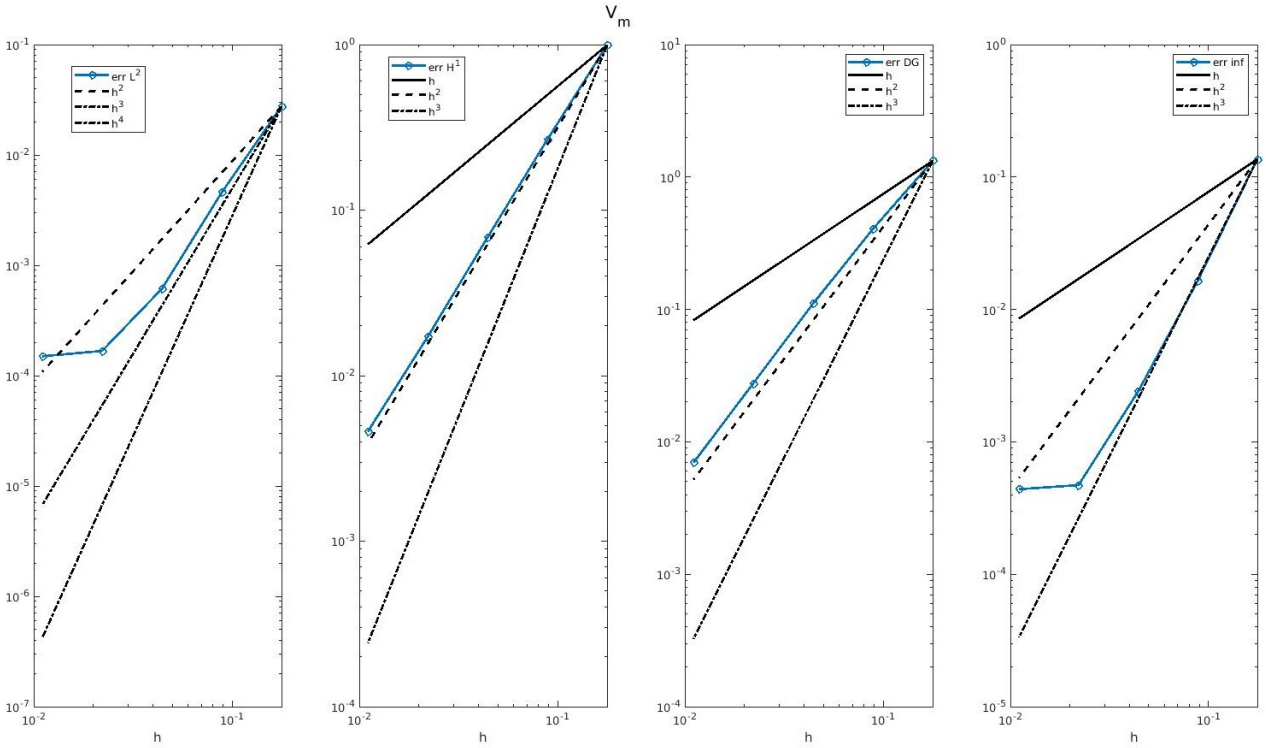
(a) Gating variable (w) with $p = 1$ Dubiner basis



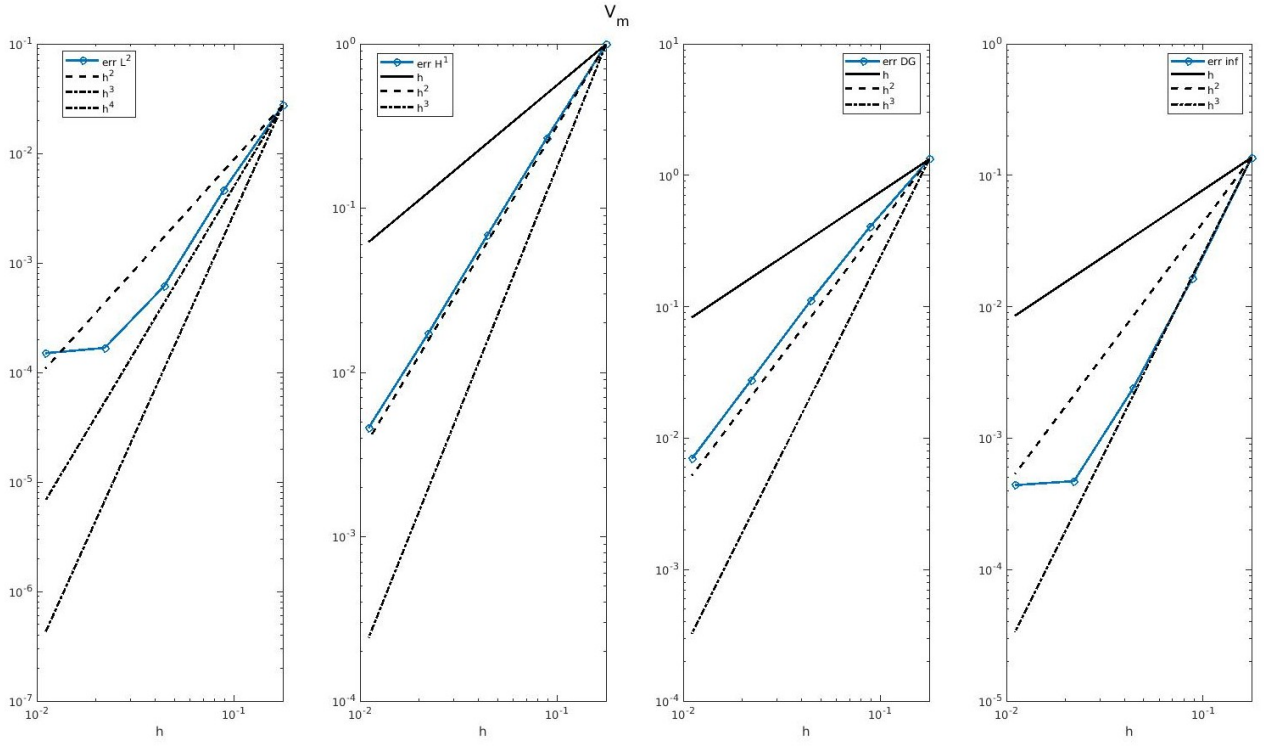
(b) Gating variable (w) with $p = 1$ Lagrangian hat functions

Computed errors for Dubiner and FEM with second order polynomials

Figure 9: Comparison of the trans-membrane potential (V_m)

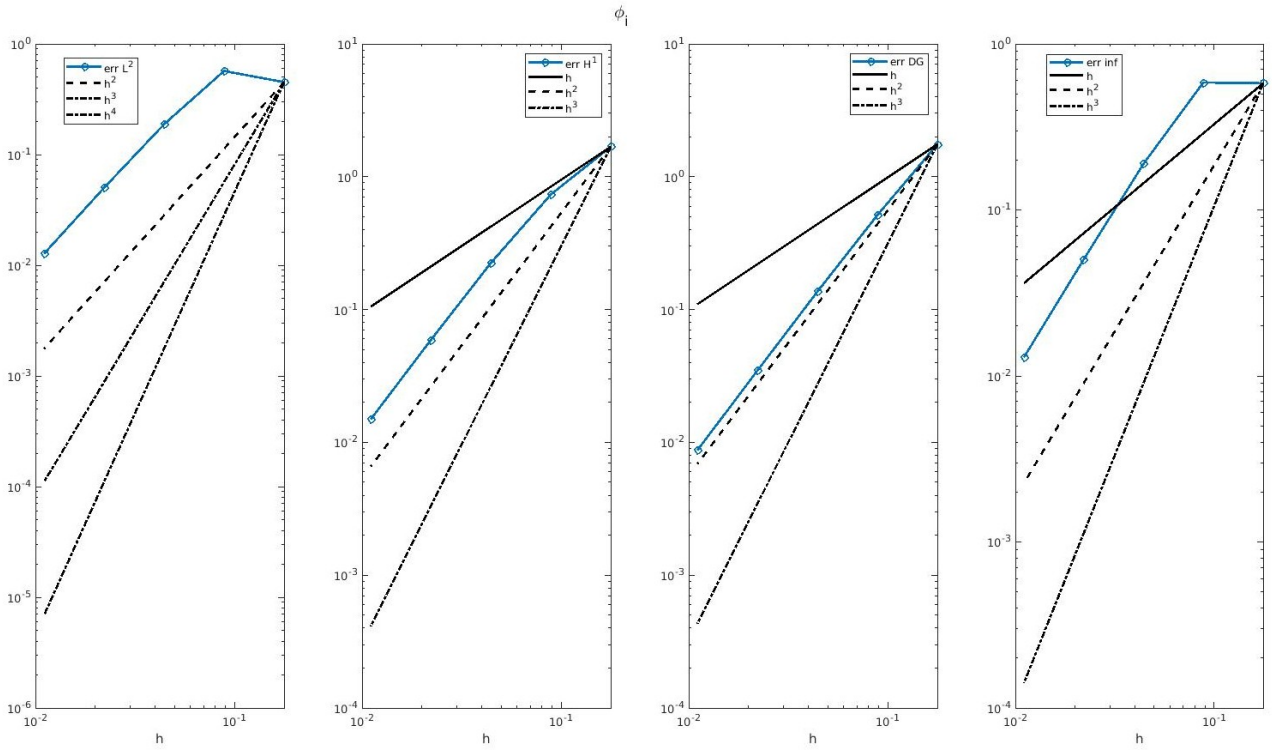


(a) Trans-membrane potential (V_m) with $p = 2$ Dubiner basis

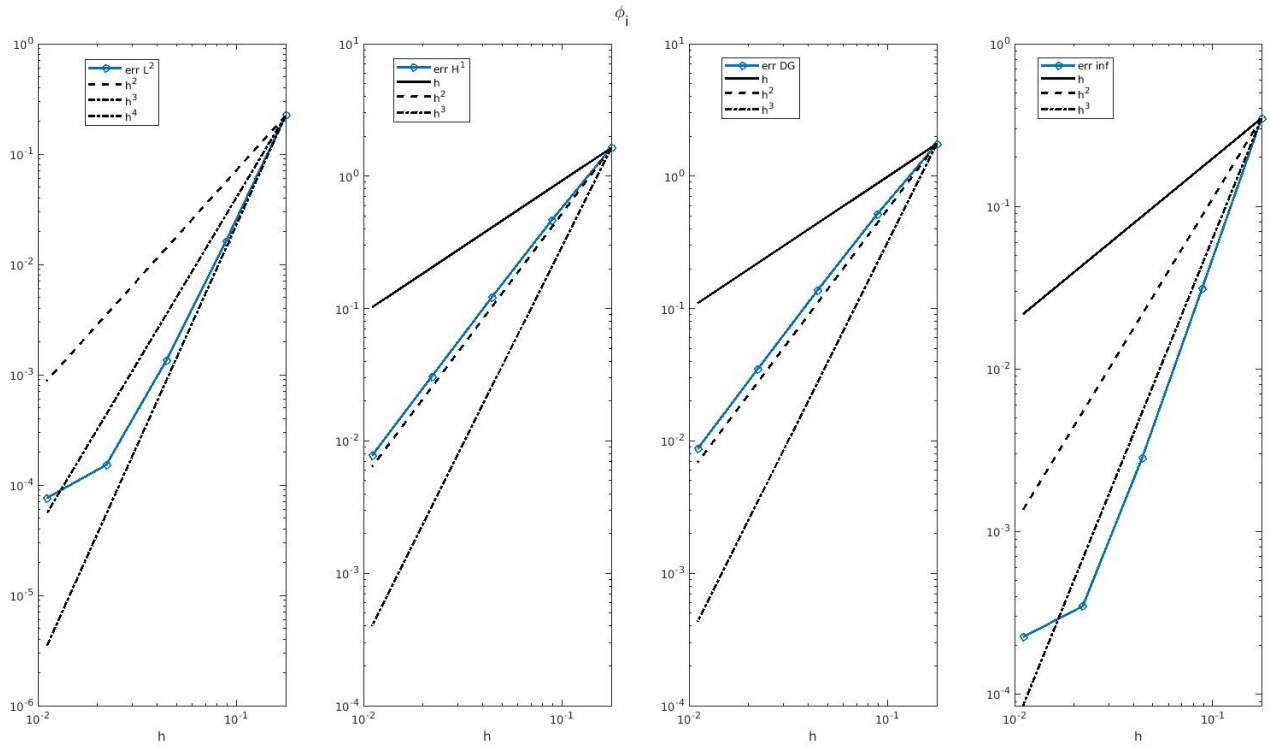


(b) Trans-membrane potential (V_m) with $p = 2$ Lagrangian hat functions

Figure 10: Comparison of the intracellular potential (ϕ_i)

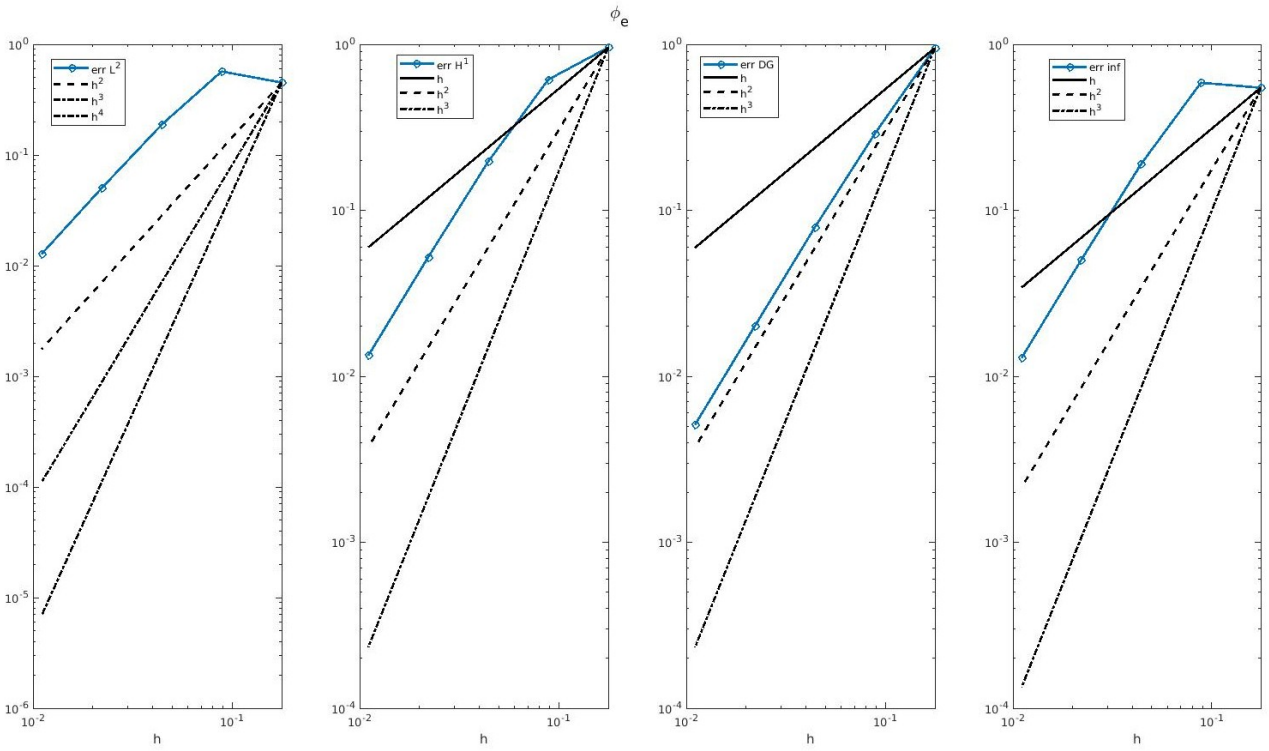


(a) Intracellular potential (ϕ_i) with $p = 2$ Dubiner basis

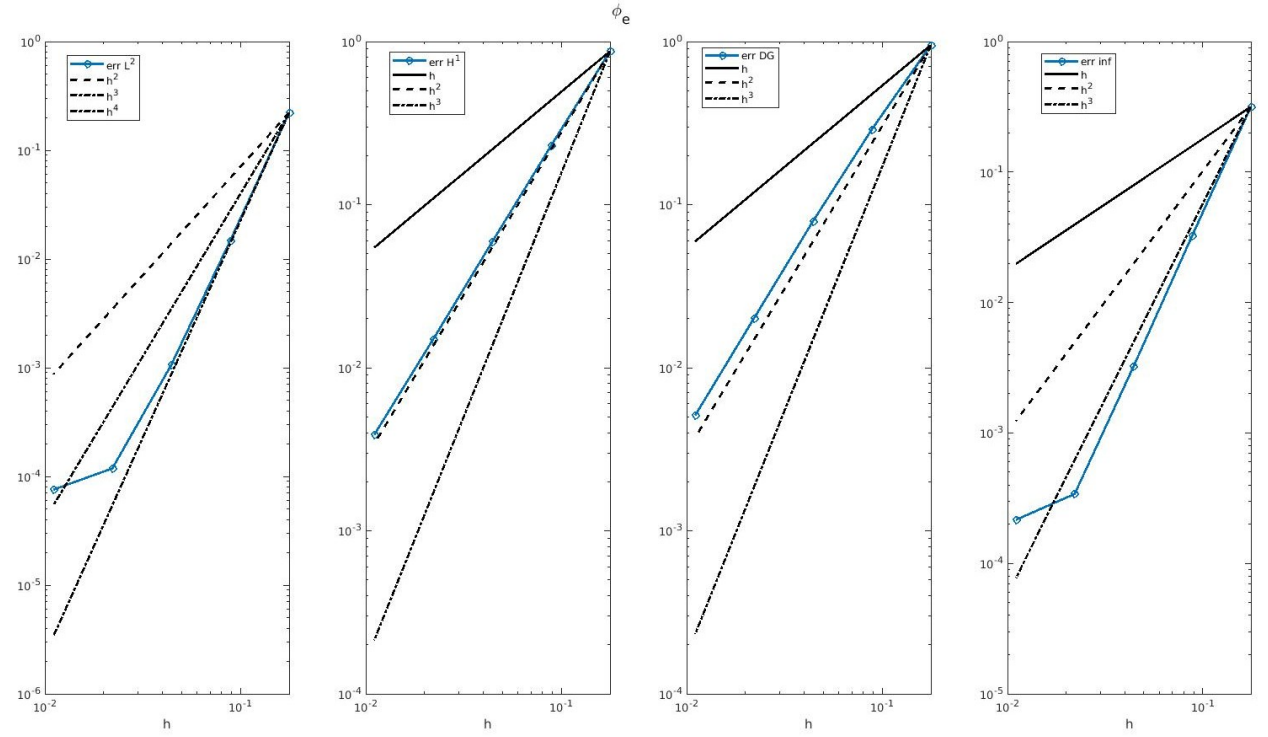


(b) Intracellular potential (ϕ_i) with $p = 2$ Lagrangian hat functions

Figure 11: Comparison of the extracellular potential (ϕ_e)

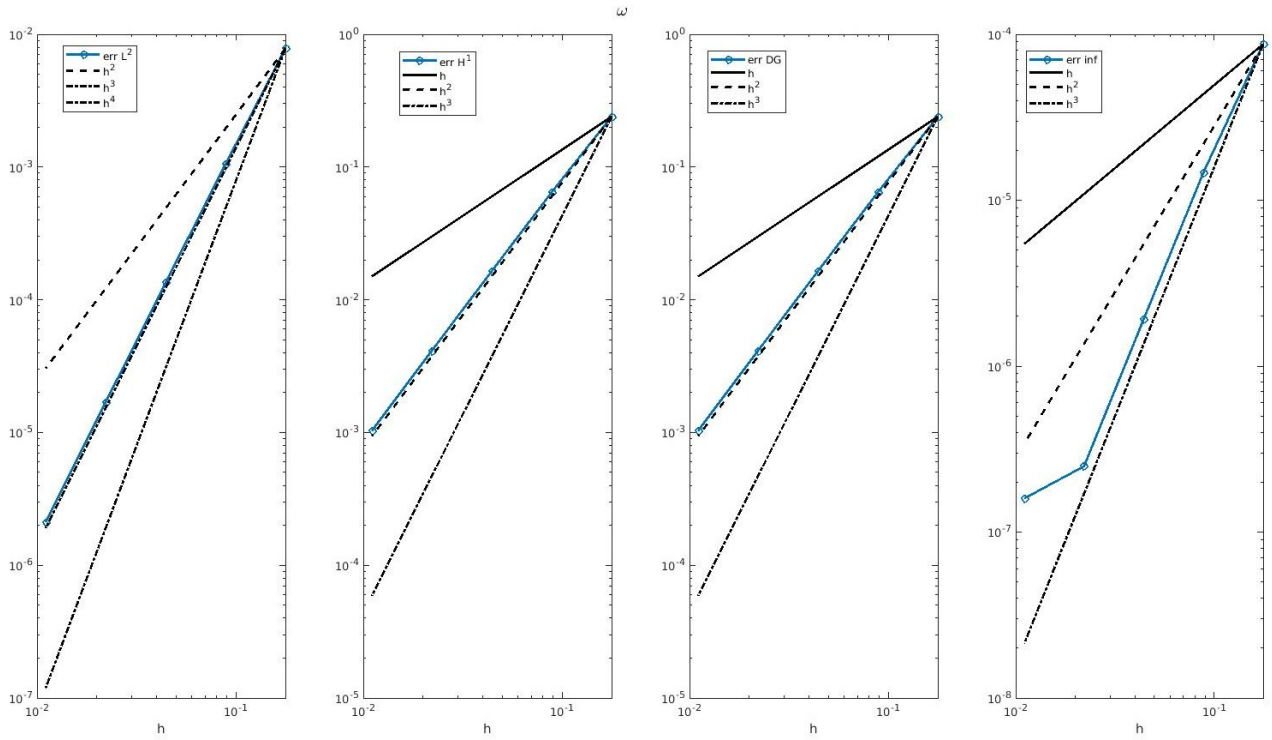


(a) Extracellular potential (ϕ_e) with $p = 2$ Dubiner basis

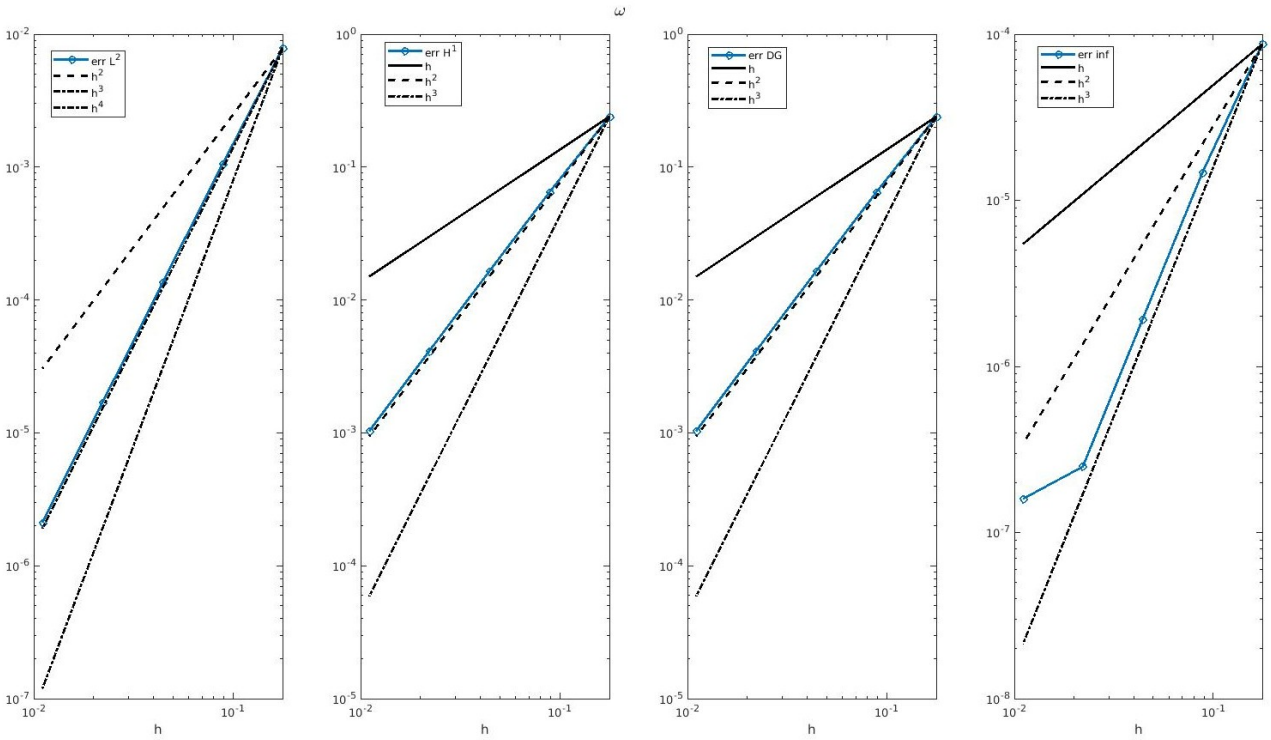


(b) Extracellular potential (ϕ_e) with $p = 2$ Lagrangian hat functions

Figure 12: Comparison of the gating variable (w)



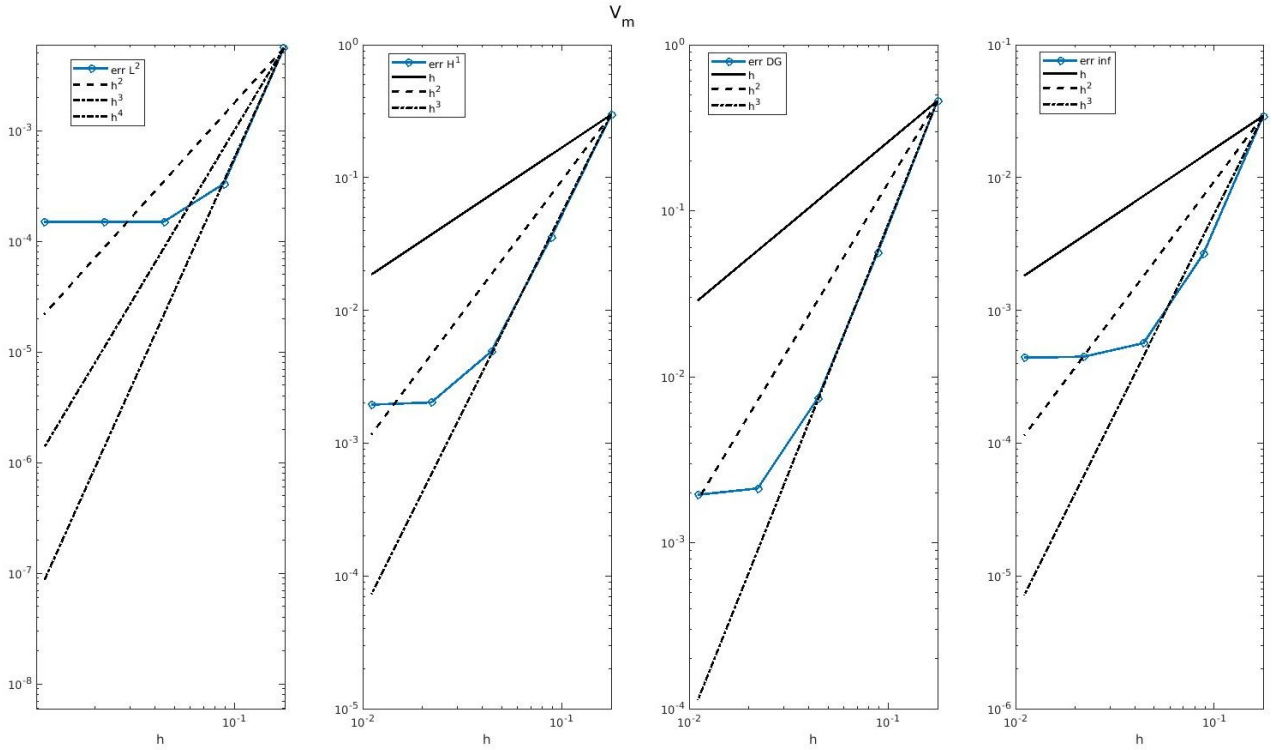
(a) Gating variable (w) with $p = 2$ Dubiner basis



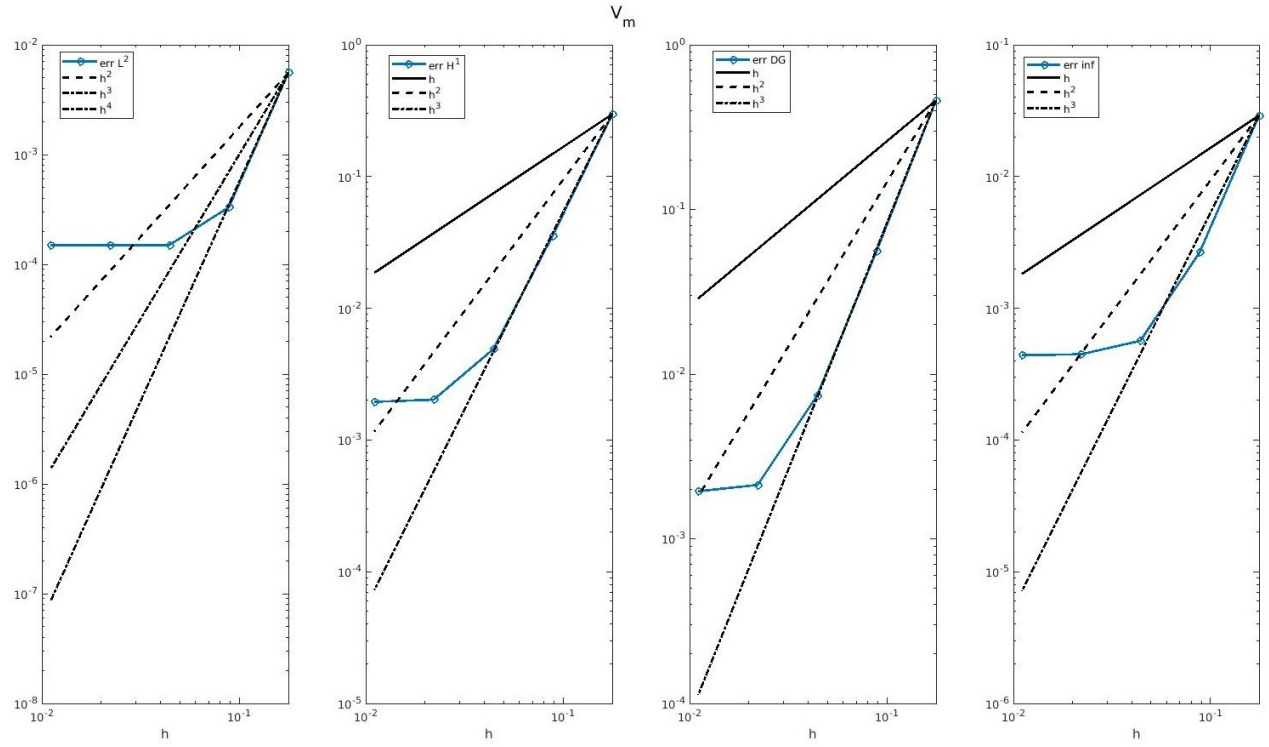
(b) Gating variable (w) with $p = 2$ Lagrangian hat functions

Computed errors for Dubiner and FEM with third order polynomials

Figure 13: Comparison of the trans-membrane potential (V_m)

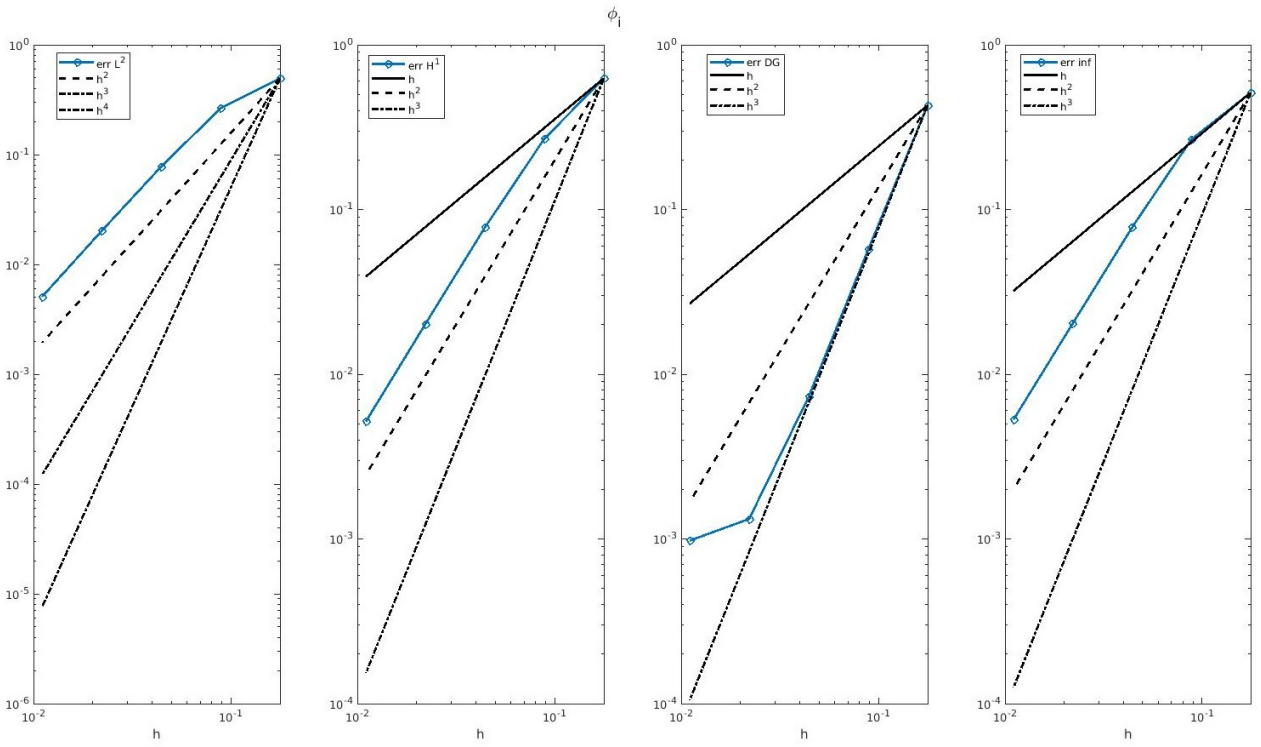


(a) Trans-membrane potential (V_m) with $p = 3$ Dubiner basis

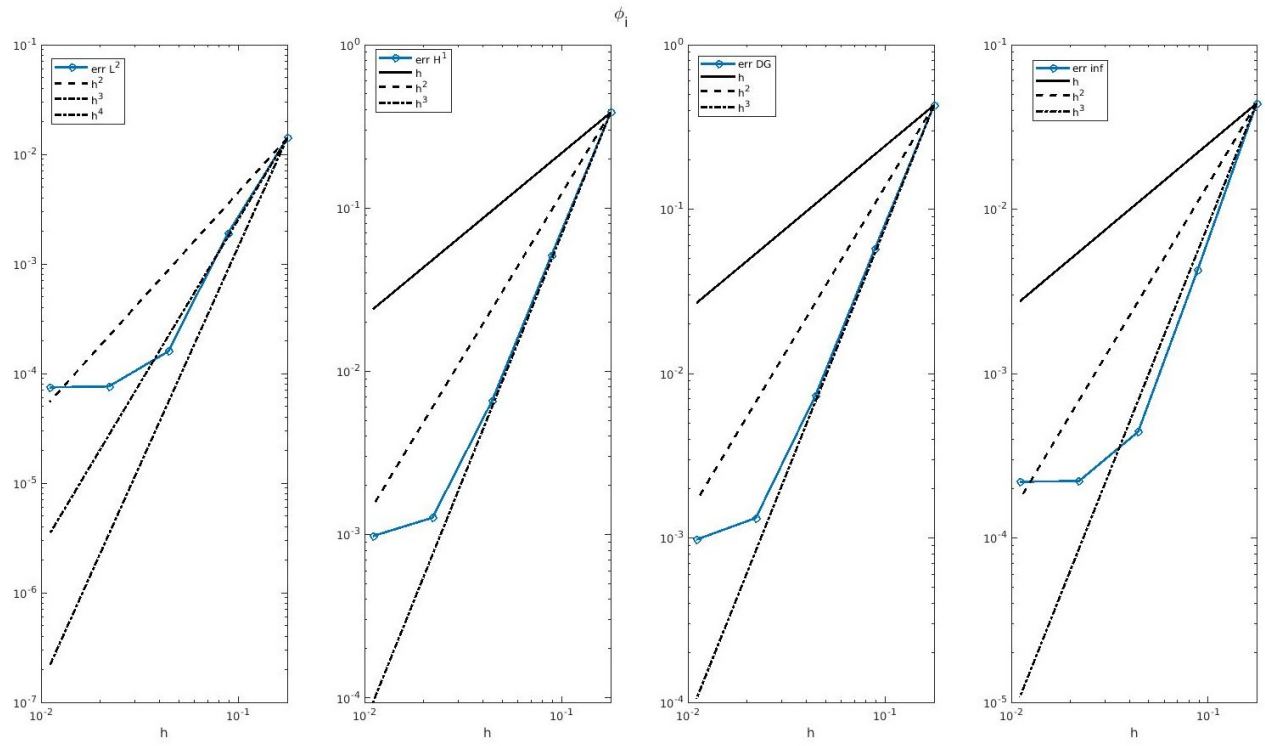


(b) Trans-membrane potential (V_m) with $p = 3$ Lagrangian hat functions

Figure 14: Comparison of the intracellular potential (ϕ_i)

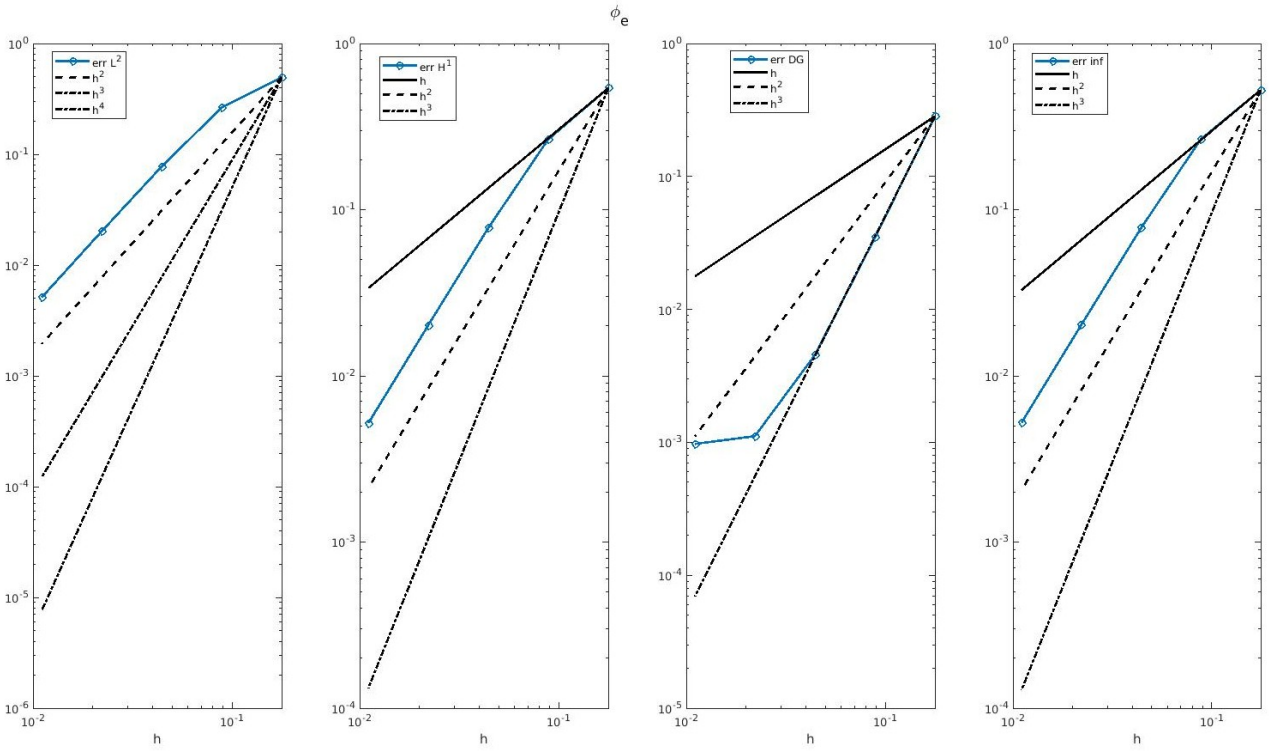


(a) Intracellular potential (ϕ_i) with $p = 3$ Dubiner basis

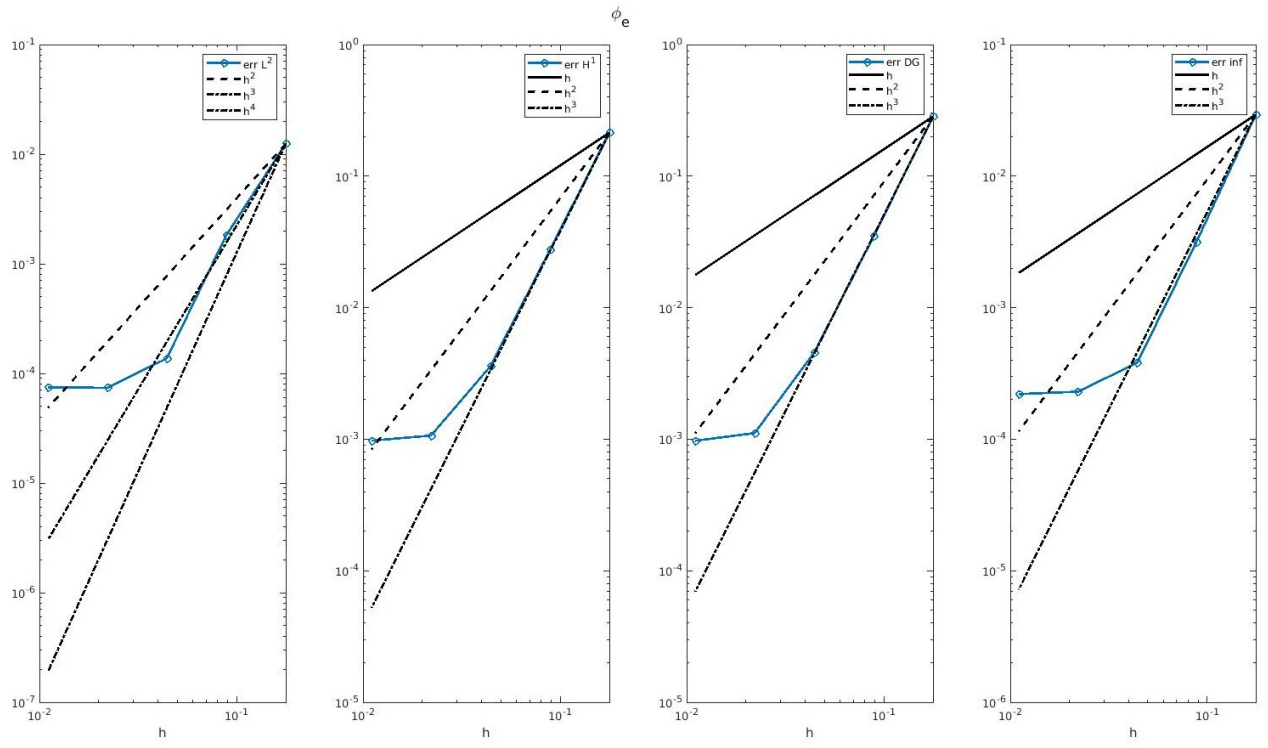


(b) Intracellular potential (ϕ_i) with $p = 3$ Lagrangian hat functions

Figure 15: Comparison of the extracellular potential (ϕ_e)

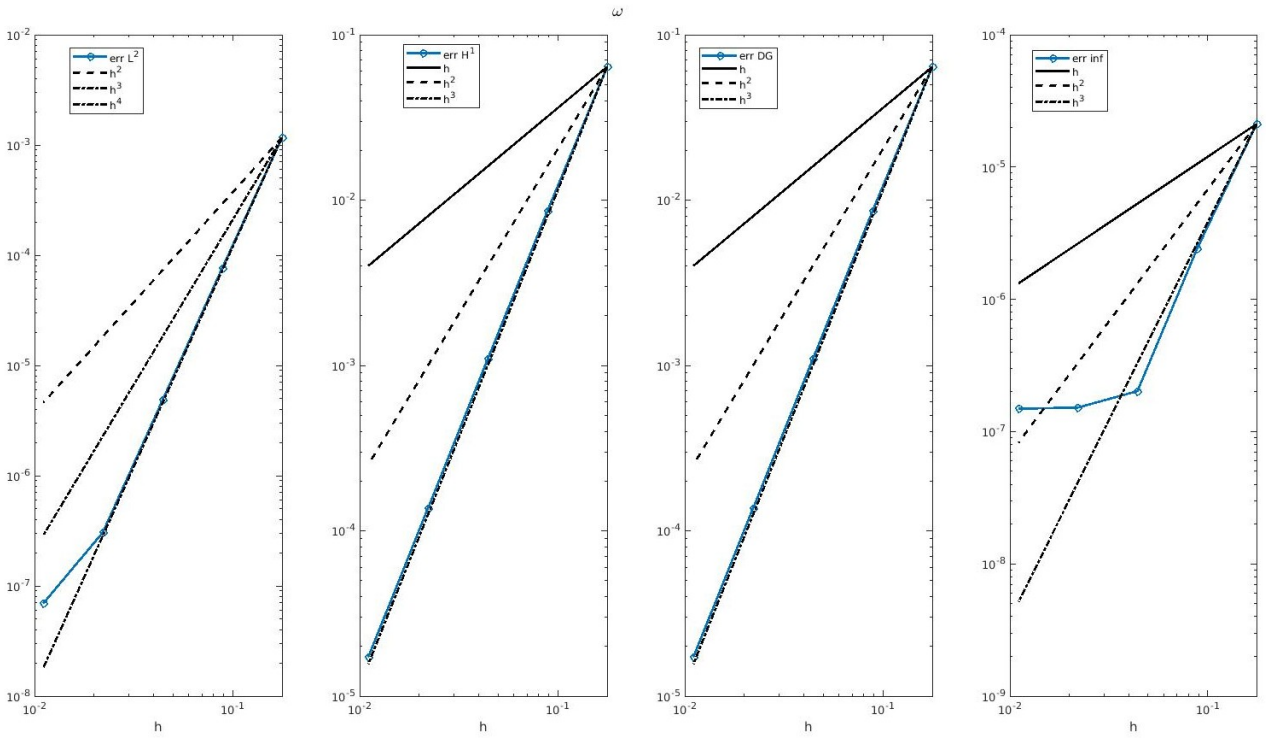


(a) Extracellular potential (ϕ_e) with $p = 3$ Dubiner basis

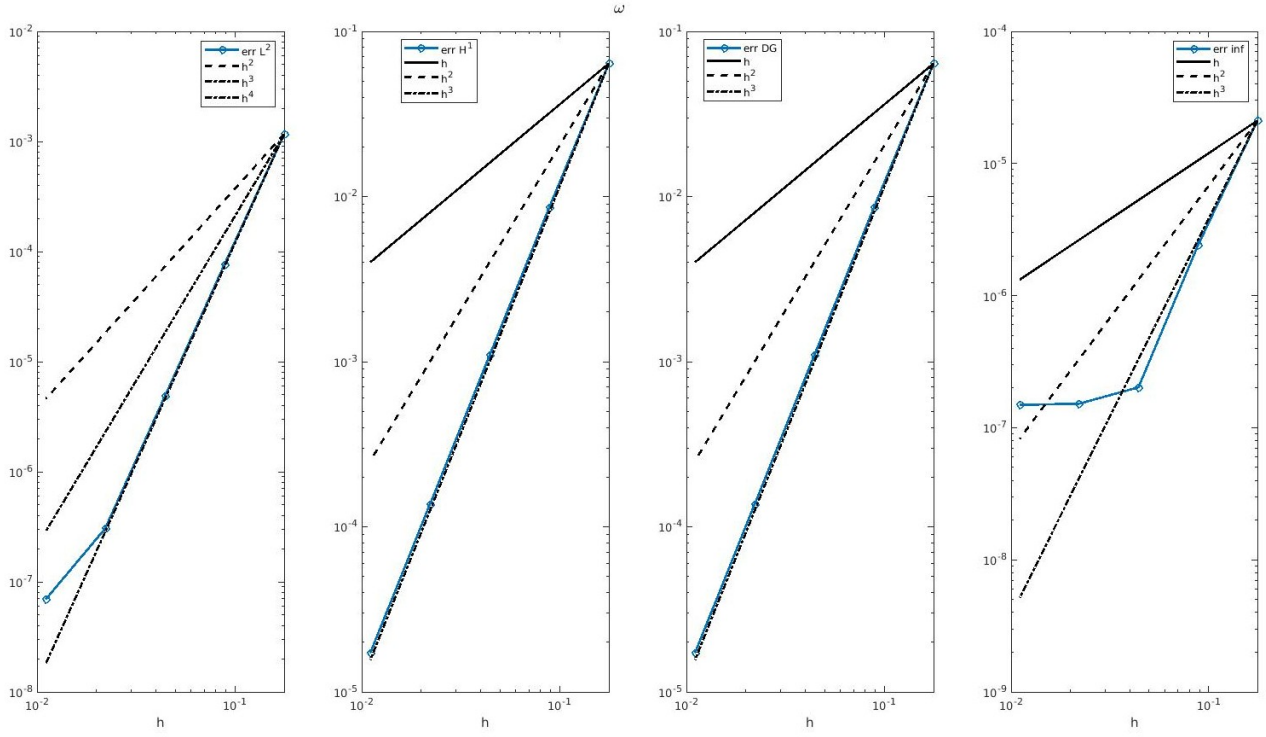


(b) Extracellular potential (ϕ_e) with $p = 3$ Lagrangian hat functions

Figure 16: Comparison of the gating variable (w)



(a) Gating variable (w) with $p = 3$ Dubiner basis



(b) Gating variable (w) with $p = 3$ Lagrangian hat functions

6.1.3 Comparison between different time discretization methods

In Section 4 three methods for the time discretization have been proposed and discussed. An error analysis comparison has been carried out and gives the graphical results shown in Fig. 17, Fig. 18, Fig. 19 and Fig. 20, we adopted for every plot the Dubiner basis with first polynomial order. It is clear that the three methods show the same convergence rate, as expected. This is expected since the time discretization method choice should not influence the space error order. Moreover, this error analysis confirms they are all consistent and work in the same way. Unfortunately, we could not have a time discretization error analysis (that would have been much more interesting) because of the presence of the spacial error that is strongly bigger than the temporal error. The main difference between the three is the explicit/implicit choice for the different terms that, in general, gives different results only in terms of stability and not of convergence order. Moreover, it is noteworthy to remember that the non-linear term is treated in explicit way for all the methods and that no temporal strategy is completely implicit. For this reason, it may be that the three methods have very similar behavior under the stability aspect too.

6.1.4 Comparison between methods for uniqueness of ϕ_i and ϕ_e

Referring to the Section 5, we have presented and explained two different methods to impose uniqueness of the cellular potentials. The simplest one, adopted for the previous error analysis, imposes the value of the function in a specific point. On the other hand, the second imposes the mean value at zero. The comparison is displayed in Fig. 21, Fig. 22, Fig. 23 and Fig. 24.
First of all, we notice that the behavior for V_m and w are the same. This fact follows from the theory since the imposition method affects only the values of the two potentials. However, for what concerns ϕ_i and ϕ_e, we see very similar results (even if not identical, the second method seems slightly more regular in the L^2 norm). These plots, then, confirm that the two methods are consistent and equivalent.

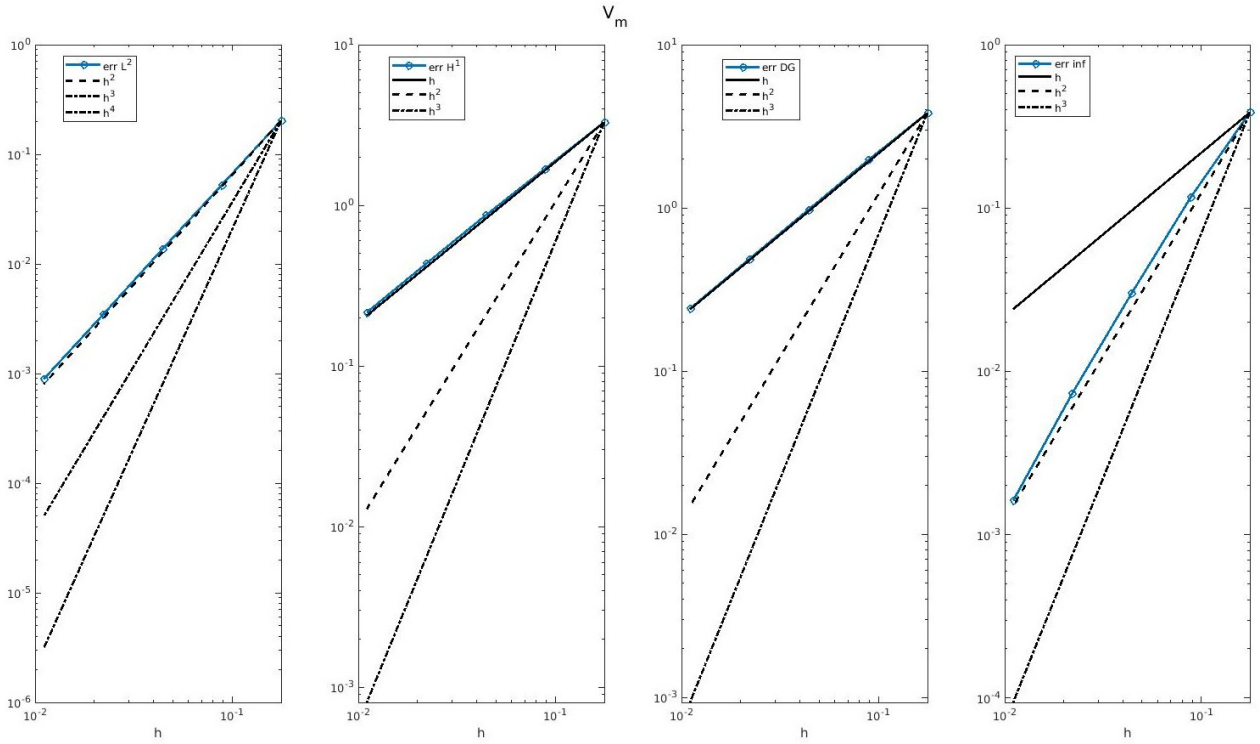
On the other hand, it has already been stated that some peculiar problems can be solved only using the second method. Indeed, the first method may have an overshooting effect that unbalances the solution. These problems will be considered in Section 6.2 regarding physical simulations. However, since there is no known analytical solution for these problems, we cannot make now an error analysis like the previous one.

In conclusion, we successfully imposed the potentials uniqueness directly into the system thanks to these two strategies. For what concerns our two initial objectives (Section 5.2), we achieved both them since:

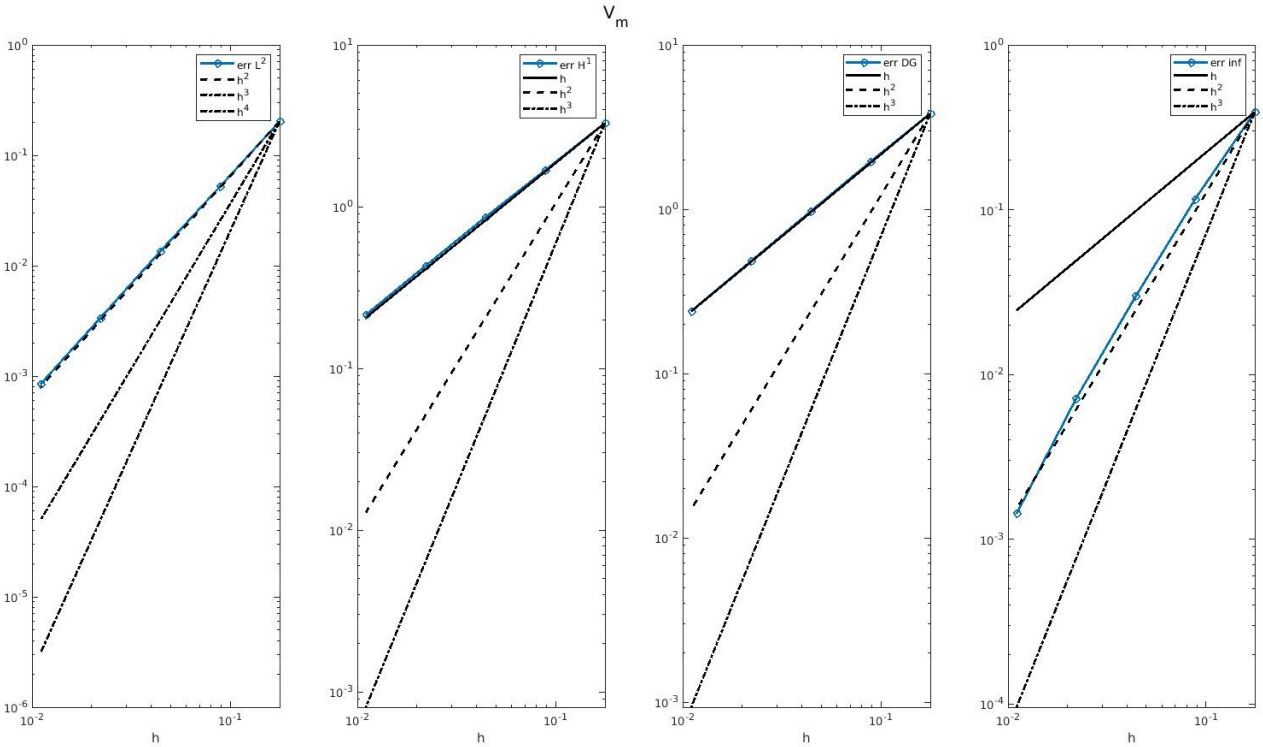
- Error plots demonstrate that the ϕ_i and ϕ_e converge to the exact potentials as V_m and w do. Moreover, plots show that all the error orders are clearly achieved.
- Conditional numbers turn to be considerably decreased. Our measures proved that they passed from $\approx 10^{17}$ to $\approx 10^7$ for both the strategies.

Computed errors for different time-discretization schemes

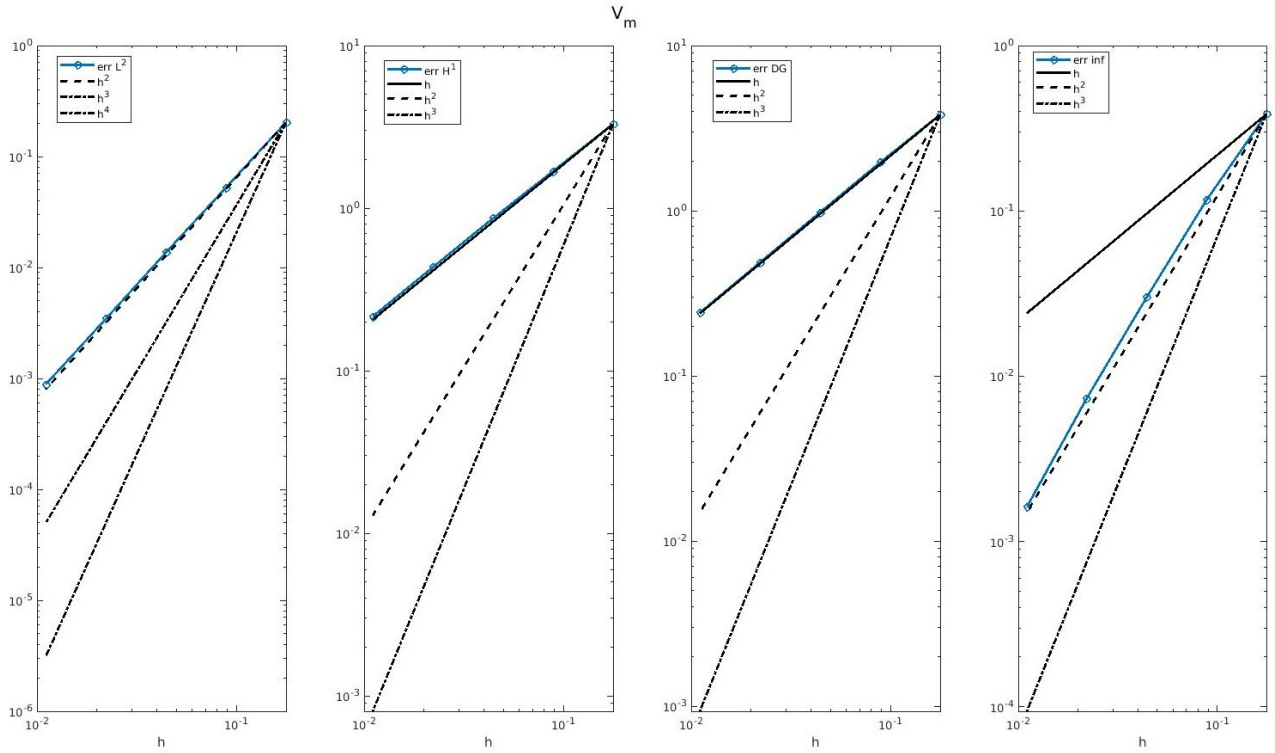
Figure 17: Comparison of the trans-membrane potential (V_m)



(a) V_m: Semi-implicit method with $p = 1$ Dubiner basis

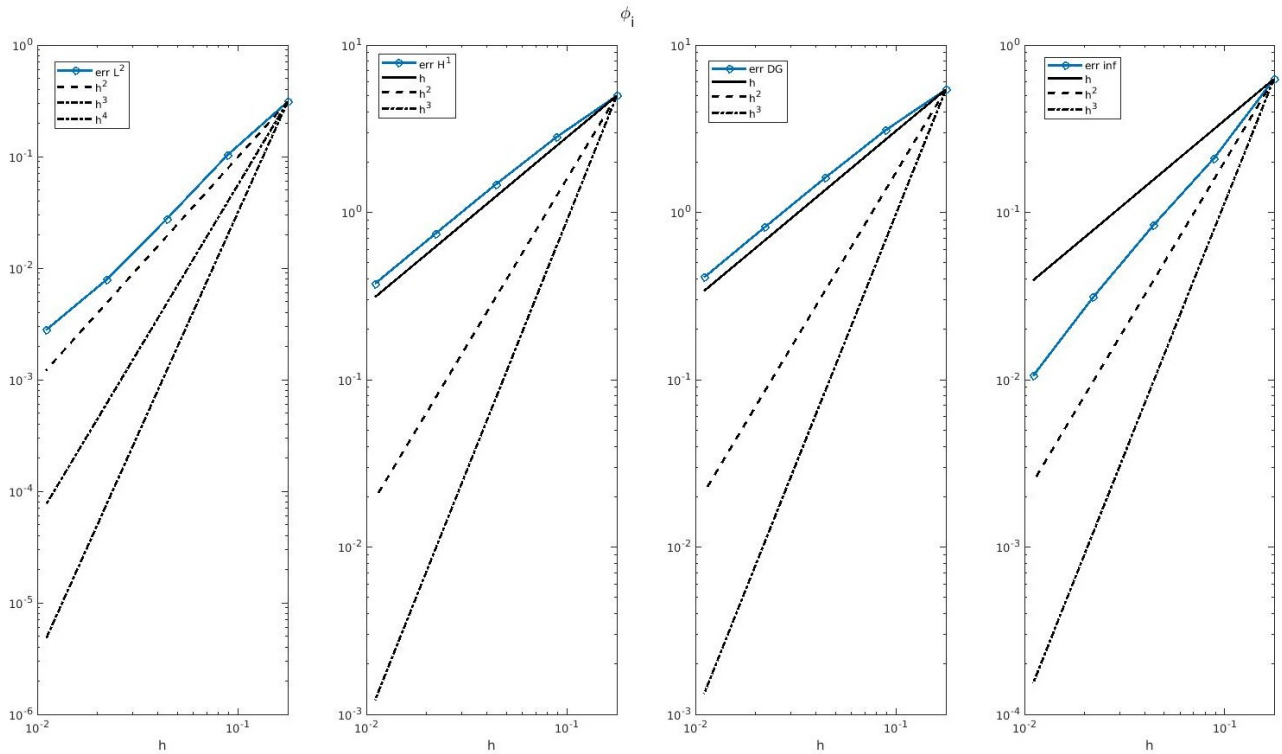


(b) V_m: Godunov operator-splitting method with $p = 1$ Dubiner basis

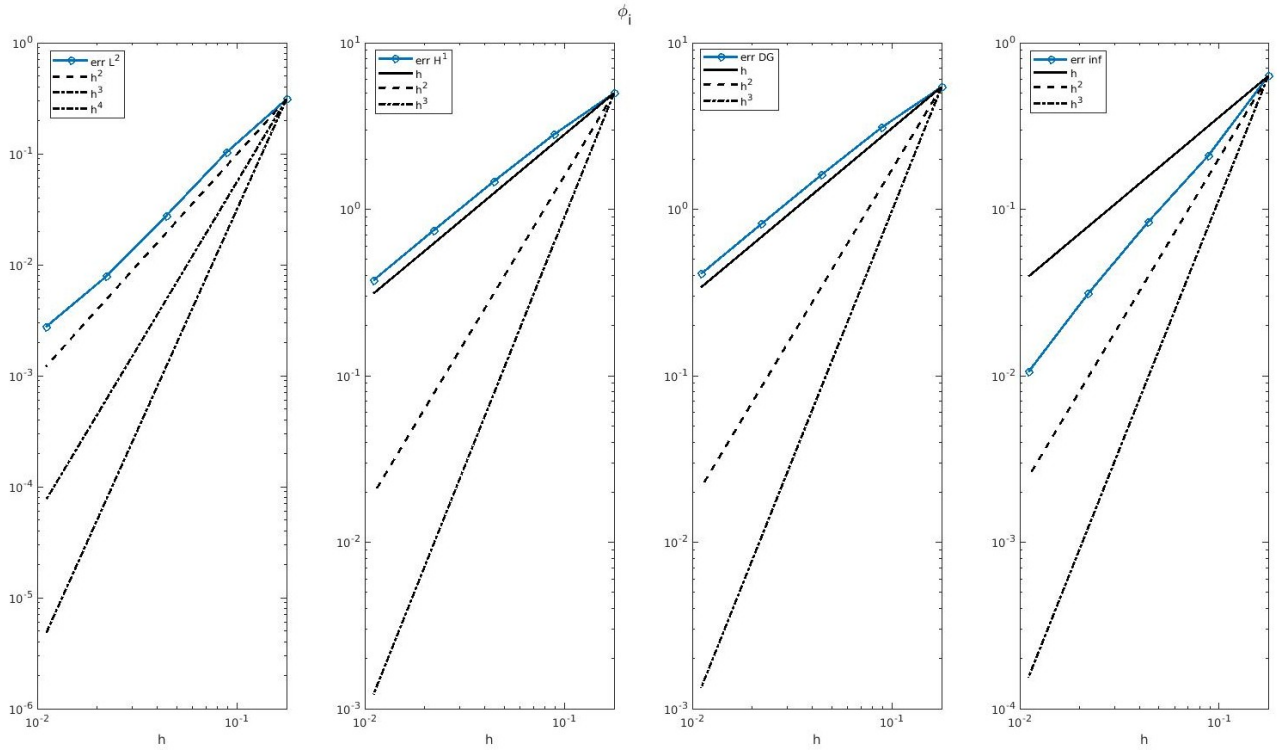


(c) V_m: Quasi-implicit operator-splitting method with $p = 1$ Dubiner basis

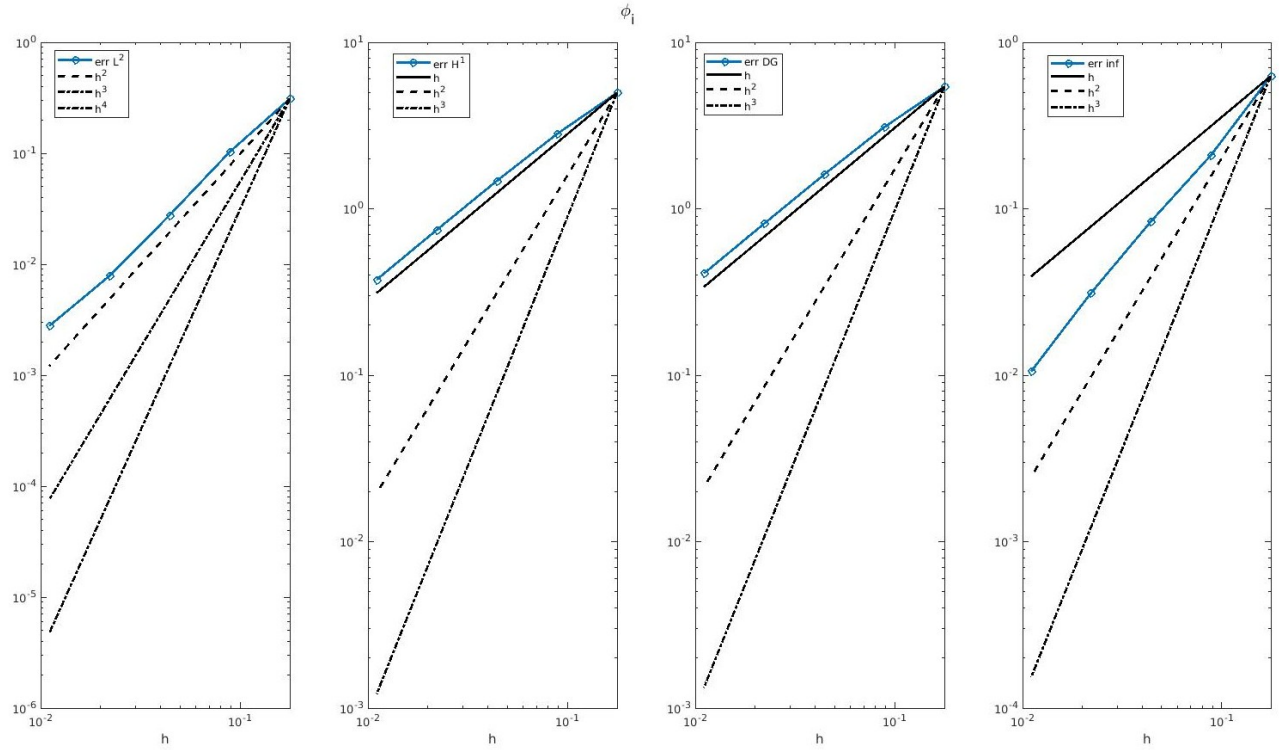
Figure 18: Comparison of the intracellular potential (ϕ_i)



(a) ϕ_i: Semi-implicit method with $p = 1$ Dubiner basis

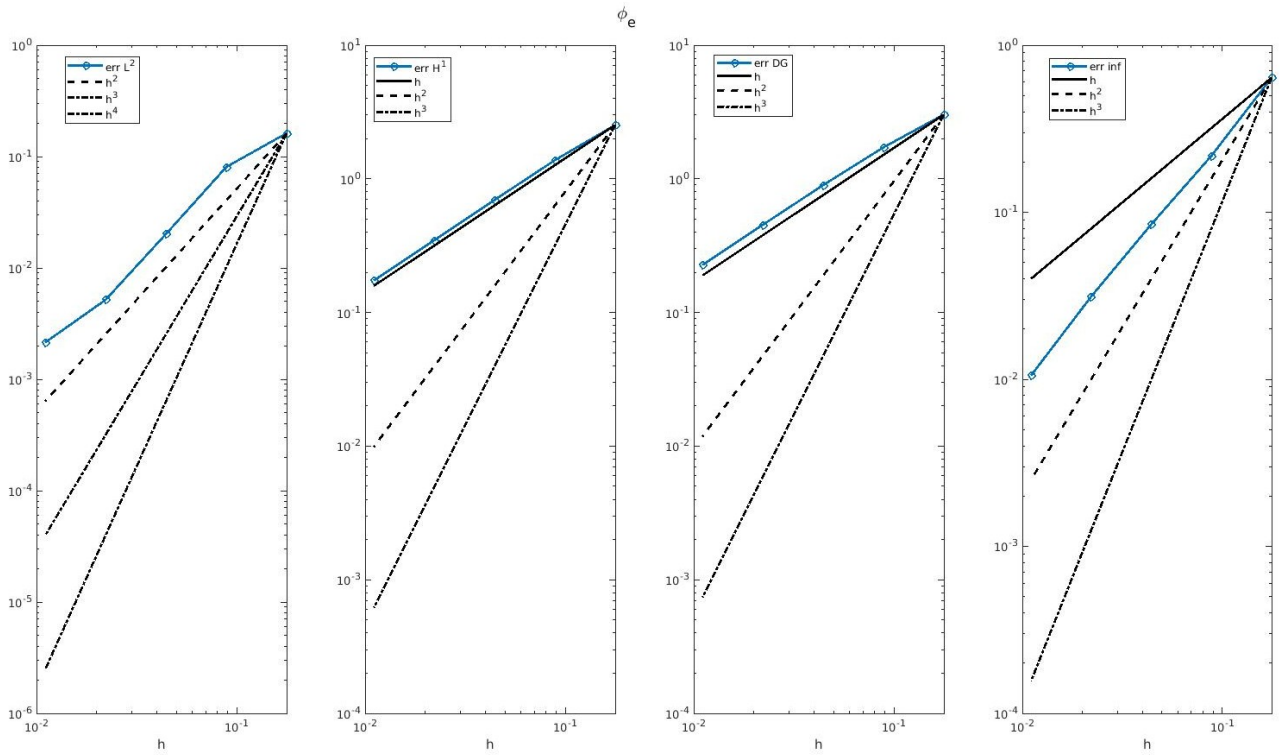


(b) ϕ_i: Godunov operator-splitting method with $p = 1$ Dubiner basis

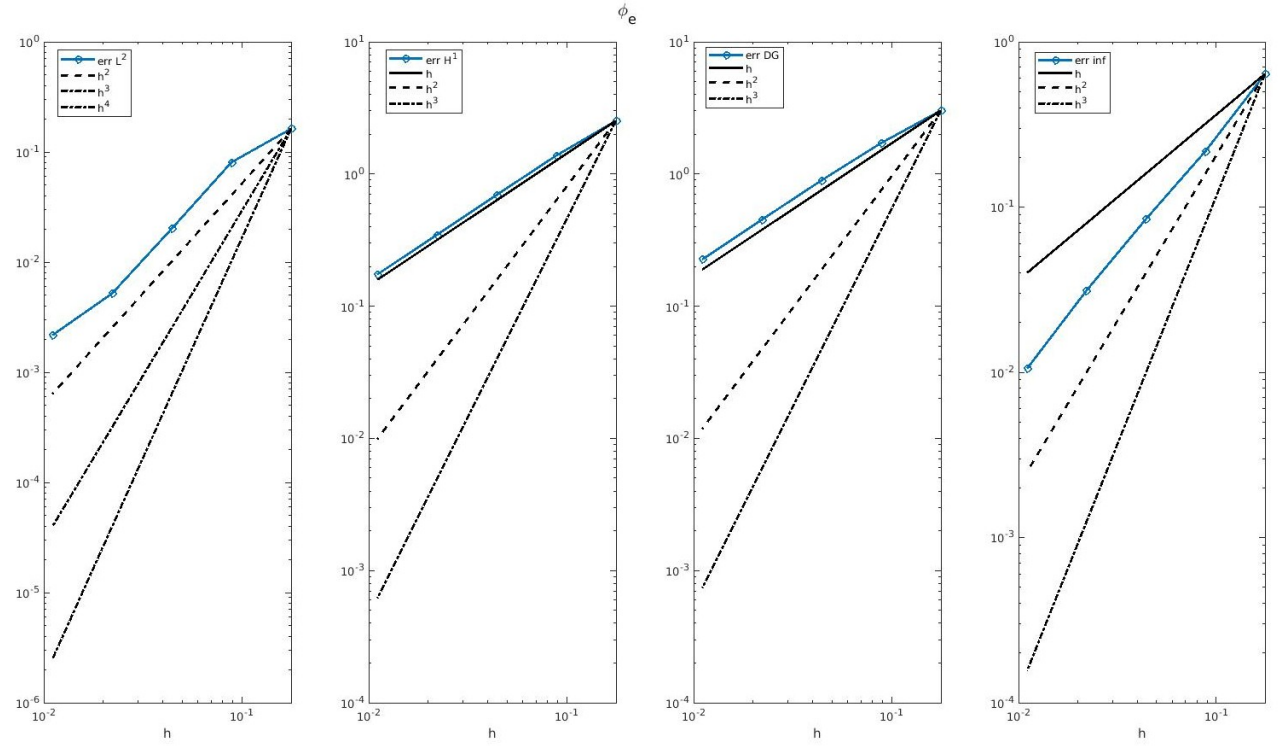


(c) ϕ_i: Quasi-implicit operator-splitting method with $p = 1$ Dubiner basis

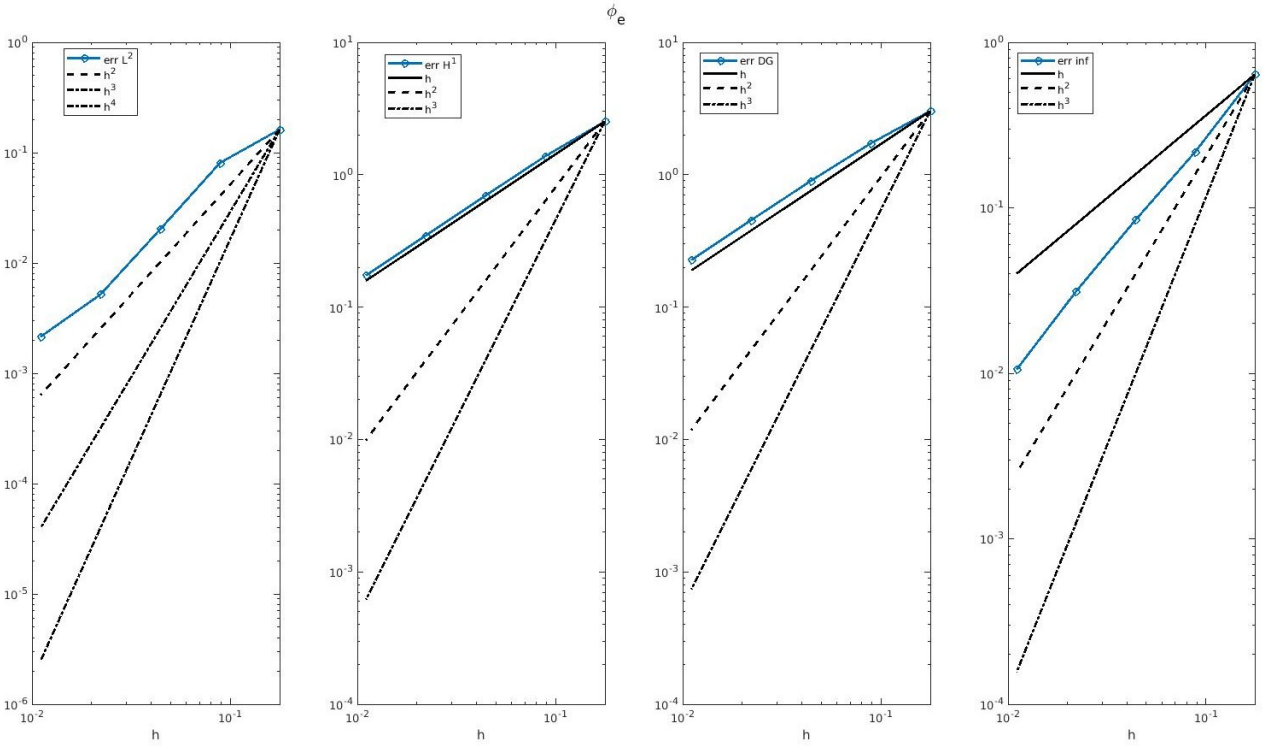
Figure 19: Comparison of the extracellular potential (ϕ_e)



(a) ϕ_e: Semi-implicit method with $p = 1$ Dubiner basis

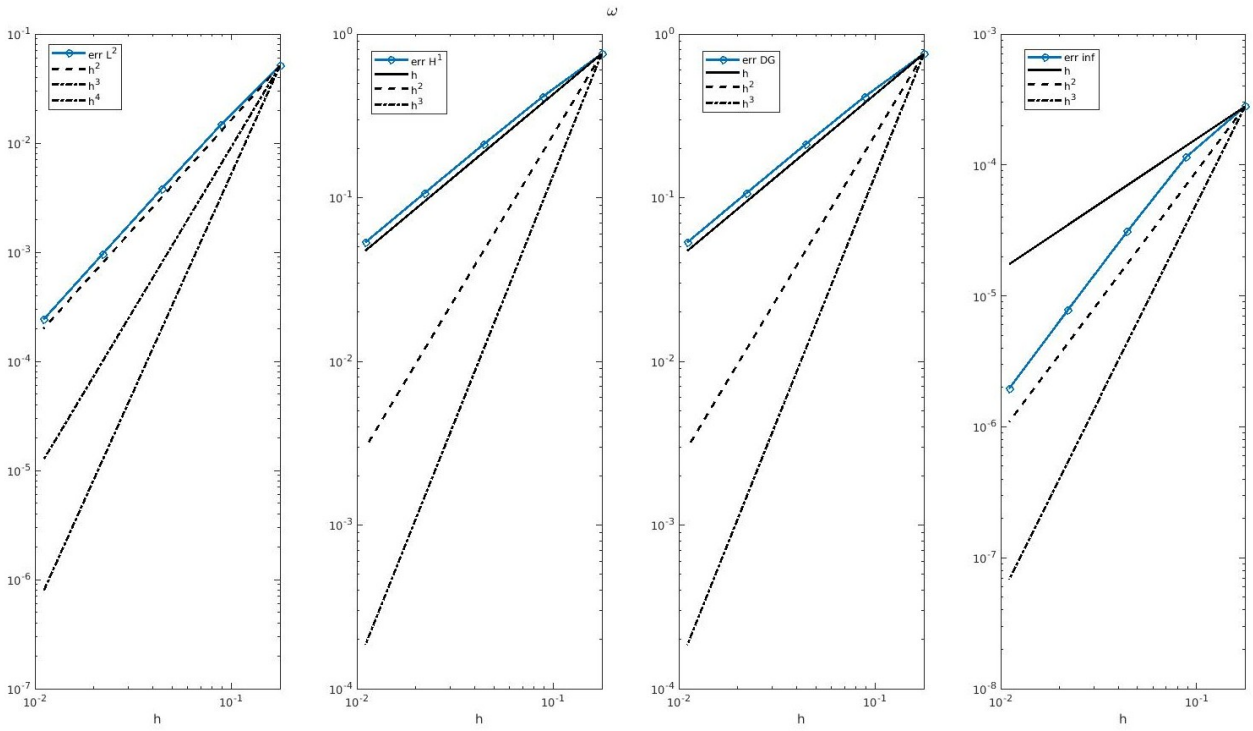


(b) ϕ_e: Godunov operator-splitting method with $p = 1$ Dubiner basis

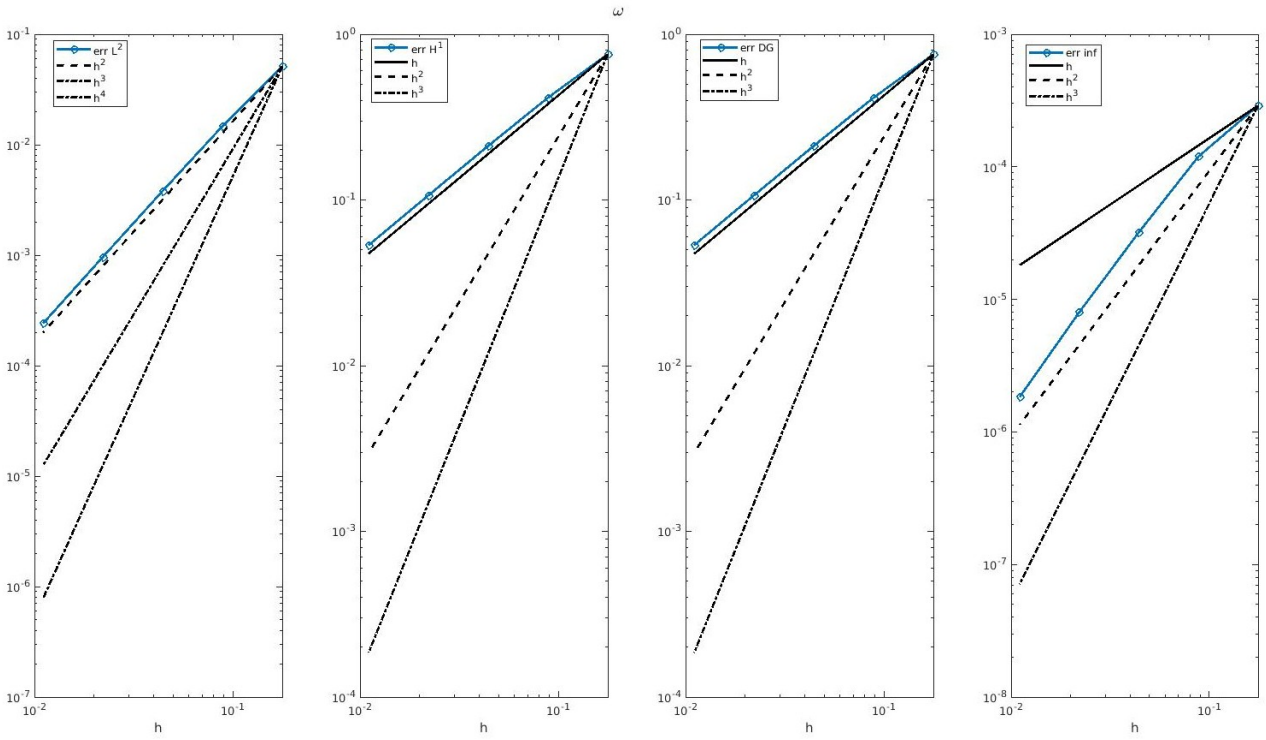


(c) ϕ_e: Quasi-implicit operator-splitting method with $p = 1$ Dubiner basis

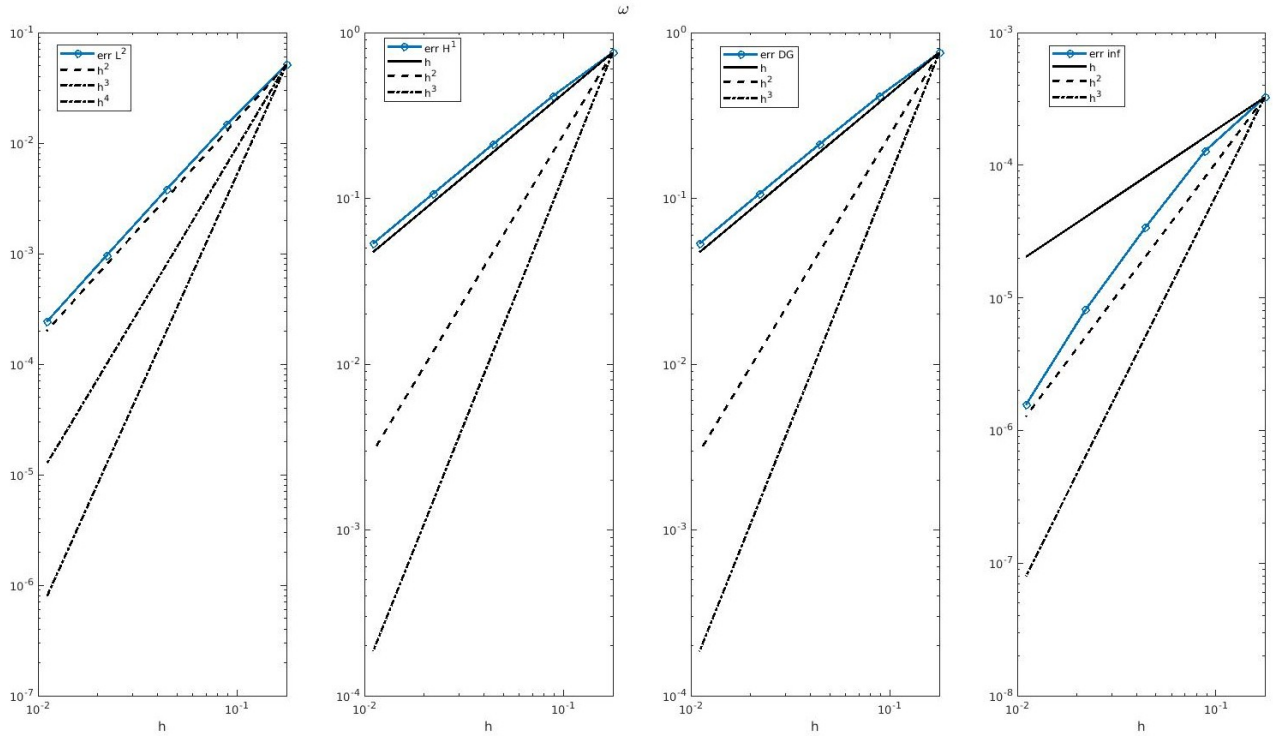
Figure 20: Comparison of the gating variable (w)



(a) w: Semi-implicit method with $p = 1$ Dubiner basis



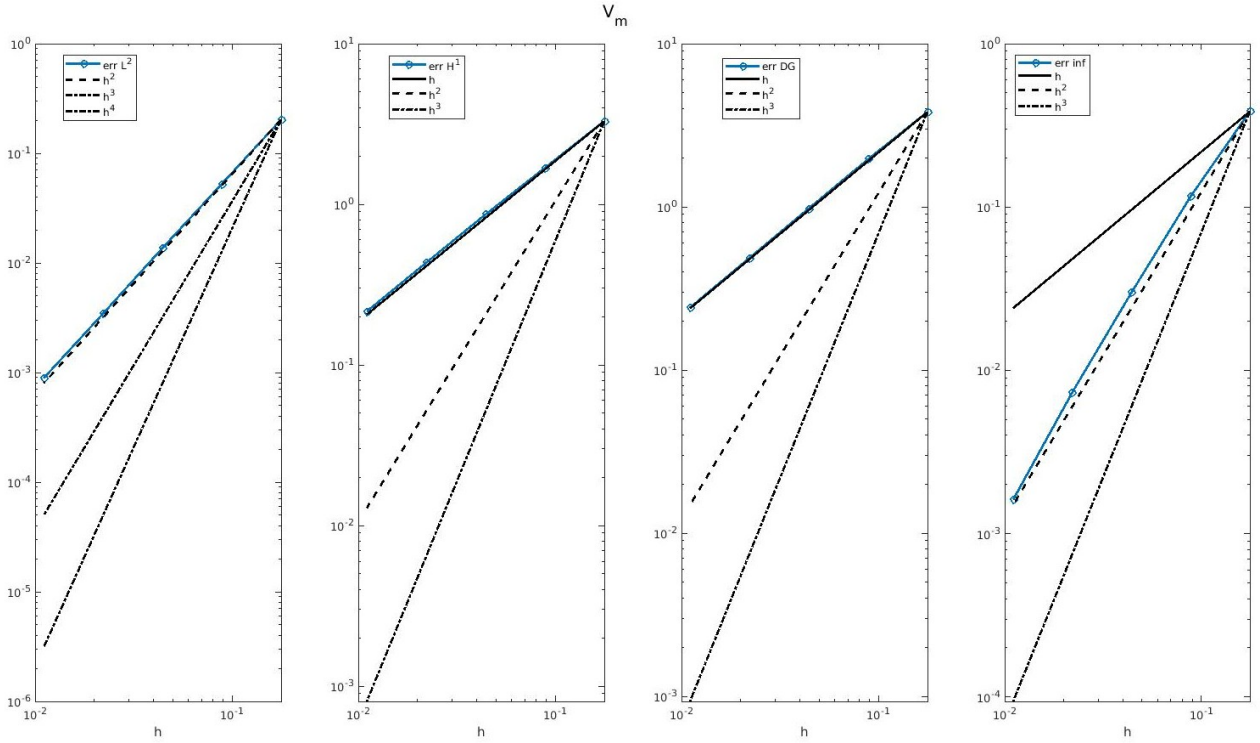
(b) w: Godunov operator-splitting method with $p = 1$ Dubiner basis



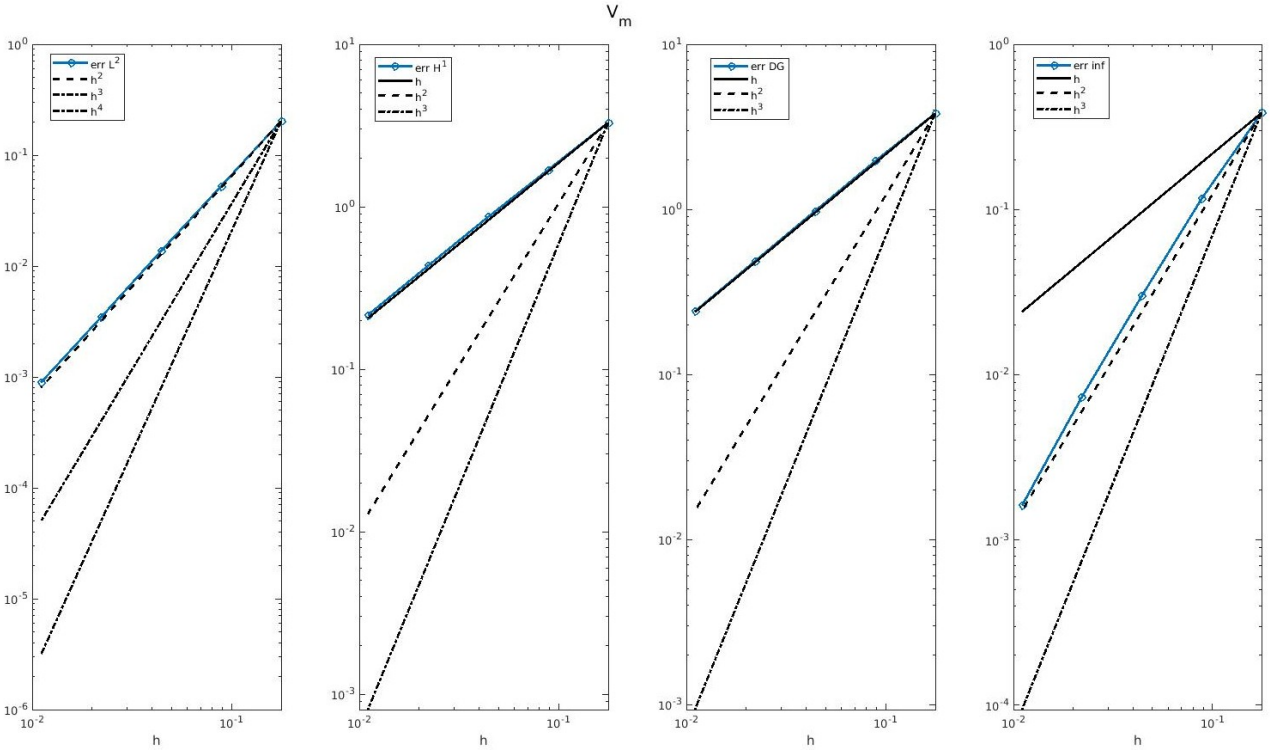
(c) w: Quasi-implicit operator-splitting method with $p = 1$ Dubiner basis

Computed errors for the uniqueness imposition strategies

Figure 21: Comparison of the trans-membrane potential (V_m)

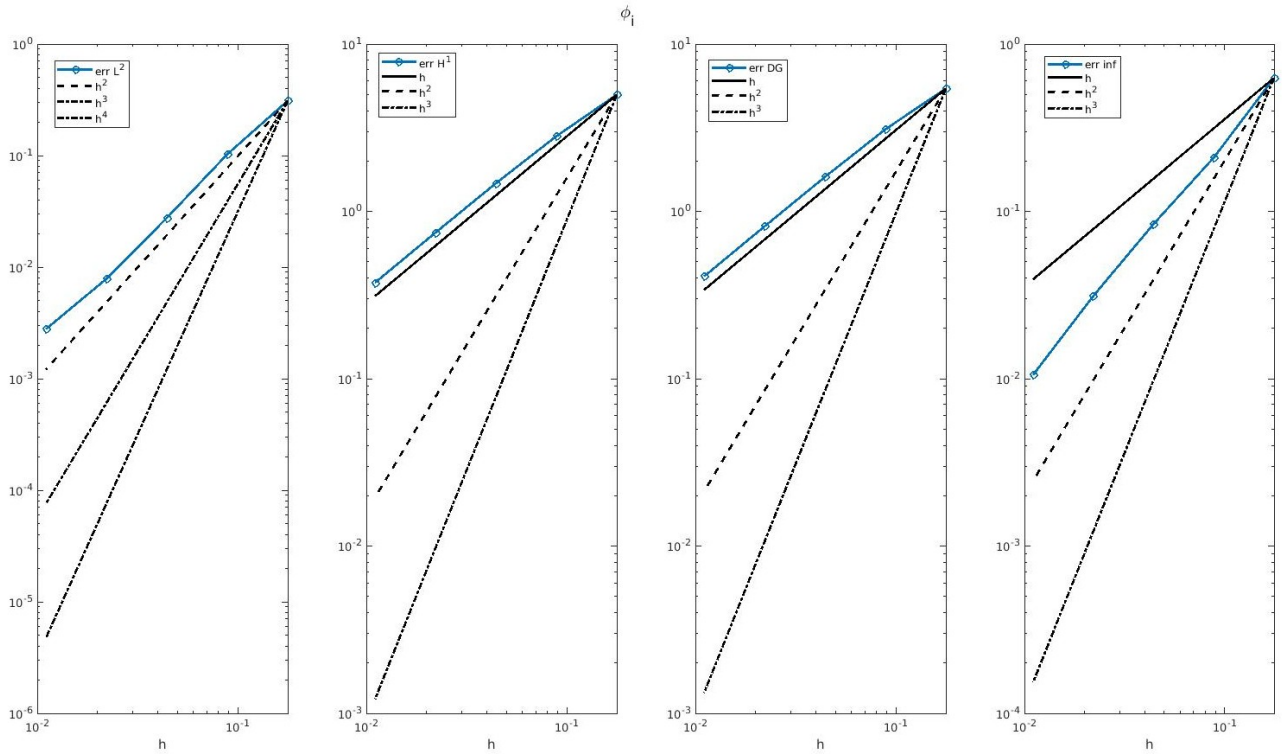


(a) V_m: imposing the first coefficient with $p = 1$ Dubiner basis

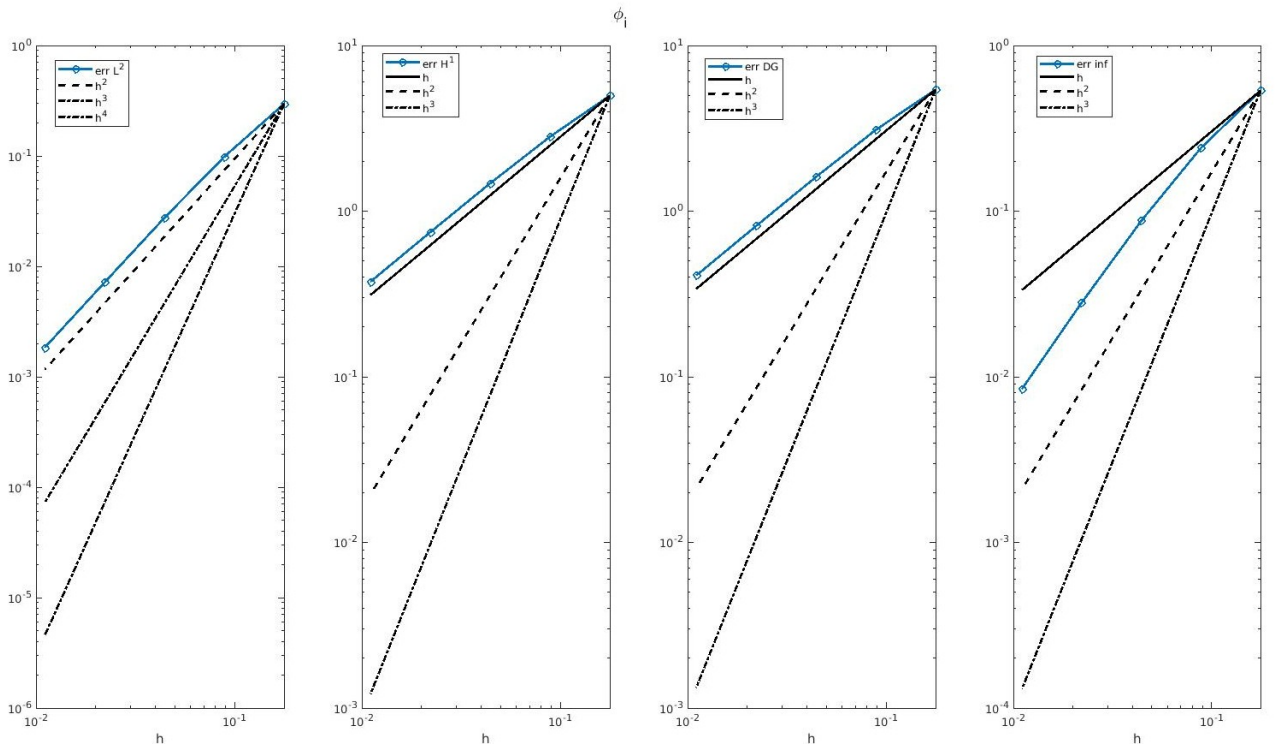


(b) V_m: imposing the zero mean with $p = 1$ Dubiner basis

Figure 22: Comparison of the intracellular potential (ϕ_i)

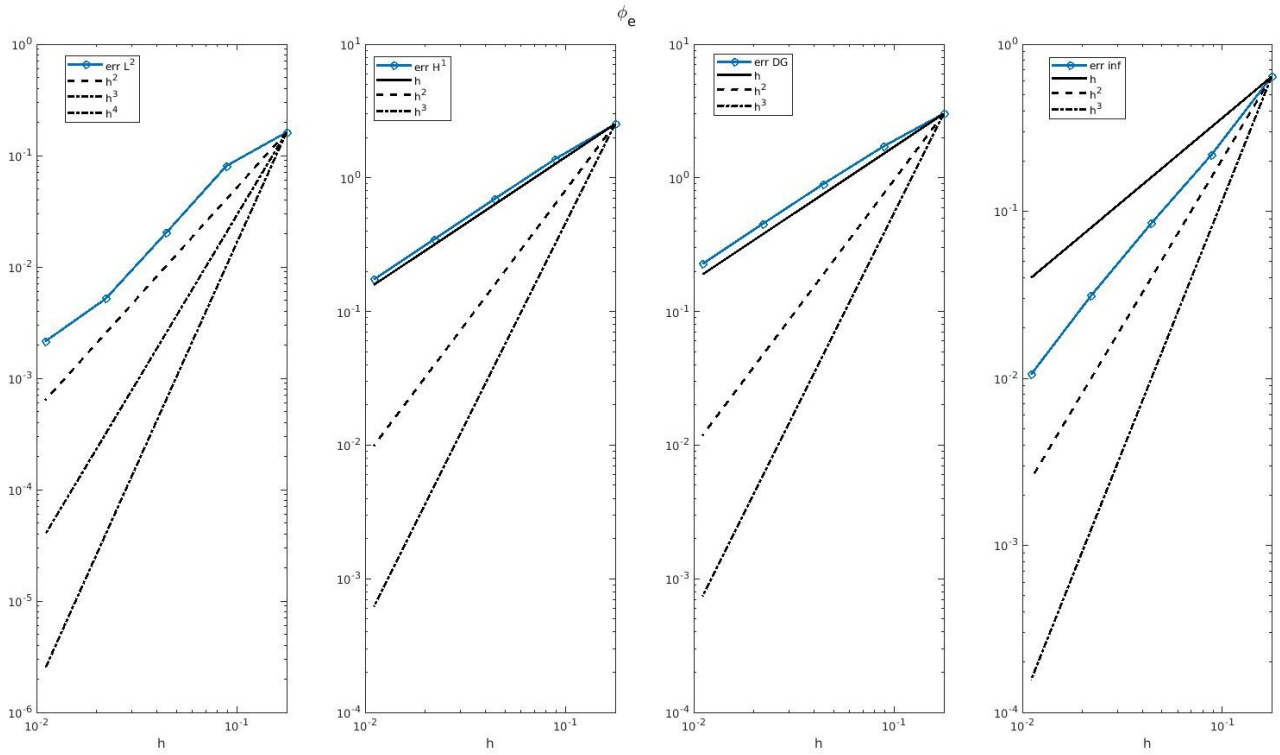


(a) ϕ_i: imposing the first coefficient with $p = 1$ Dubiner basis

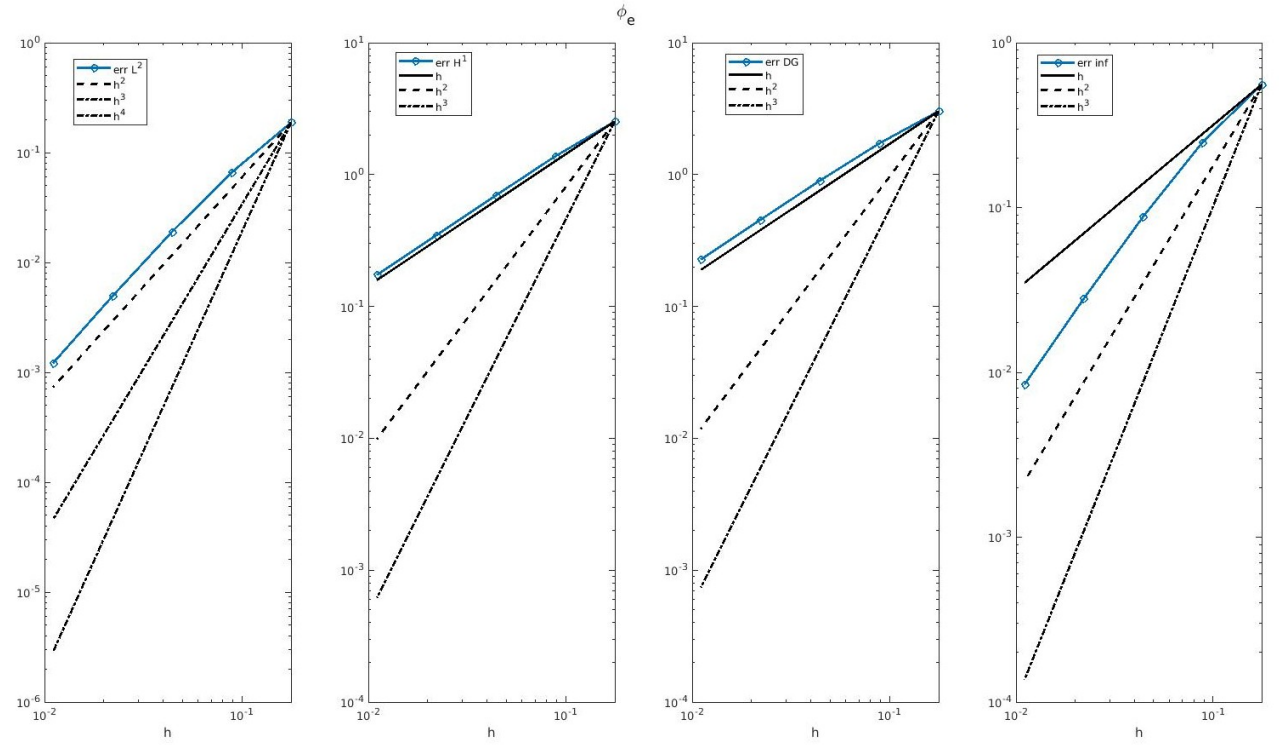


(b) ϕ_i: imposing the zero mean with $p = 1$ Dubiner basis

Figure 23: Comparison of the extracellular potential (ϕ_e)

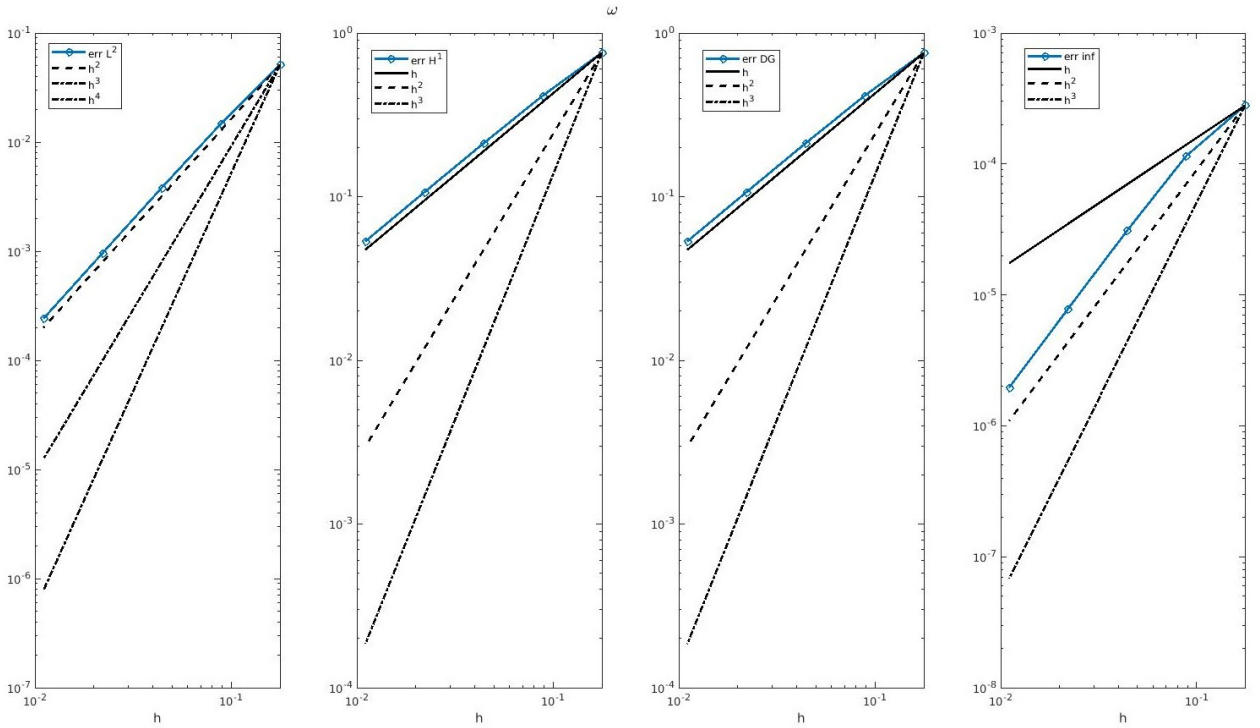


(a) ϕ_e: imposing the first coefficient with $p = 1$ Dubiner basis

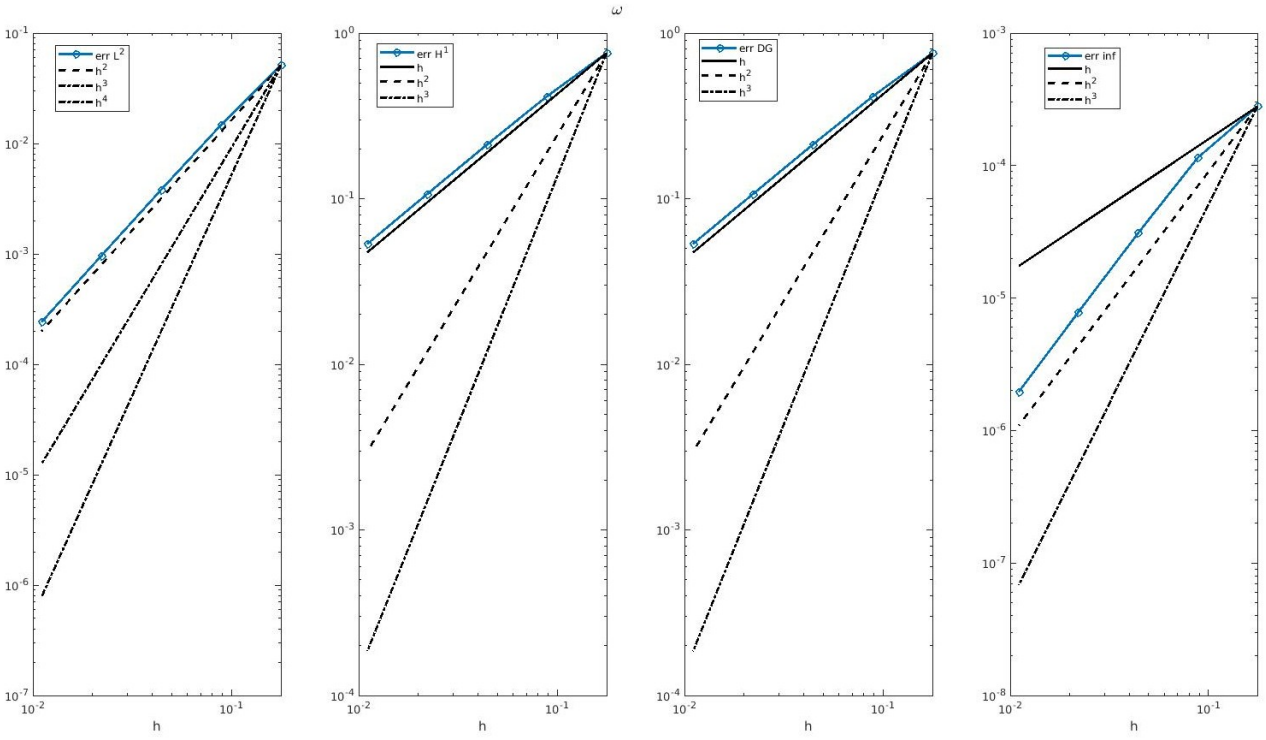


(b) ϕ_e: imposing the zero mean with $p = 1$ Dubiner basis

Figure 24: Comparison of the gating variable (w)



(a) w: imposing the first coefficient with $p = 1$ Dubiner basis



(b) w: imposing the zero mean with $p = 1$ Dubiner basis

6.2 Toward a realistic simulation

Our final goal is to exploit the strategies and tools presented in the previous sections to run simulations that can mimic the cardiac electrophysiology phenomena. It is noteworthy to warn that no exact solutions are known for these kinds of problems. Moreover, we point out that there are some limitations to the complete resemblance to the realistic phenomena, for instance the domain shape. Our general choices for all the test-cases are reported in Table 2.

Table 2: Parameters for pseudo-realistic simulations

Domain (m)	$\Omega = (-0.025, 0.035)^2$
Temporal scheme	Semi-implicit
Polynomials space	D1
dt (s)	0.0001
$nREF$	5
Initial condition for V_m (V)	0
Initial condition for w (Am^{-2})	0
I_i^{ext} (Am^{-3})	$I \cdot 10^3 \, \chi_{[0.001,0.002]}(t) \, \chi_{[0.0045,0.0055]}(x) \, \chi_{[0.0045,0.0055]}(y)$
I_e^{ext} (Am^{-3})	$I \cdot 10^3 \, \chi_{[0.001,0.002]}(t) \, \chi_{[0.0045,0.0055]}(x) \, \chi_{[0.0045,0.0055]}(y)$
b_i (Am^{-2})	0
b_e (Am^{-2})	0
χ_m (m^{-1})	10^5
C_m (Fm^{-2})	10^{-2}
Σ_i (Sm^{-1})	$\begin{bmatrix} 0.34 & 0 \\ 0 & 0.06 \end{bmatrix}$
Σ_e (Sm^{-1})	$\begin{bmatrix} 0.62 & 0 \\ 0 & 0.24 \end{bmatrix}$

In Table 2, I is a positive value to be chosen depending on the test case. We remind from Section 5.2.4 that the mean value imposition is always chosen for realistic simulations.

Moreover, we observe that the main differences with the previous parameters are the square domain that is no more unitary and the anisotropy of the diffusion tensors. The latter choice is motivated from the real utility of the *Bidomain Model* if compared to the *Monodomain Model*, where the two tensors are assumed to be equal or proportional. Parameter values are taken from literature, see [12].

Physically, this setting represents a square section of the surface of the heart that is electrically isolated (homogeneous boundary conditions) and an external current that is applied in a central region for a limited interval of time. It could be seen as a simulation of the action of a defibrillator.
To conclude, we observe that the compatibility condition (equation (1)) for the existence of the solution is satisfied since boundary conditions are null and external currents are the same in the intracellular and extracellular regions.

For what concerns the *FitzHugh-Nagumo* parameters, there are no general values as for the previous ones. To generalize this aspect, we defined two different test-cases with two different sets of parameters, as shown in Table 3.

Table 3: FitzHugh-Nagumo Parameters for pseudo-realistic simulations

	Test-case 1	Test-case 2
k	19.5	1
ϵ	1.2	0.2232
γ	0.1	4.0322
a	$13 \cdot 10^{-3}$	0.004

The first test-case data employs the values already adopted for Section 6.1 and taken from past projects [4],[2], [10]. Meanwhile, the second test-case parameters are taken from [1].

6.2.1 First test-case

For both the test-cases, we aim to show two different situations depending on the external current intensity: first, for too weak currents, the electrical activation should miss and this implies that the potentials are not capable to hold up (*underflow* case). Second, 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:

- $I = 500 \cdot 10^3 Am^{-3}$ is a suitable value for the *underflow* case.
- $I = 700 \cdot 10^3 Am^{-3}$ is a suitable value for the *overflow* case.

In Fig. 25 and in Fig. 26, we report the computed solutions for both choices of I using first polynomial degree Dubiner functions.

Figure 25: Test-case 1, computed snapshots for underflow case ($I = 500 \cdot 10^3 Am^{-3}$)

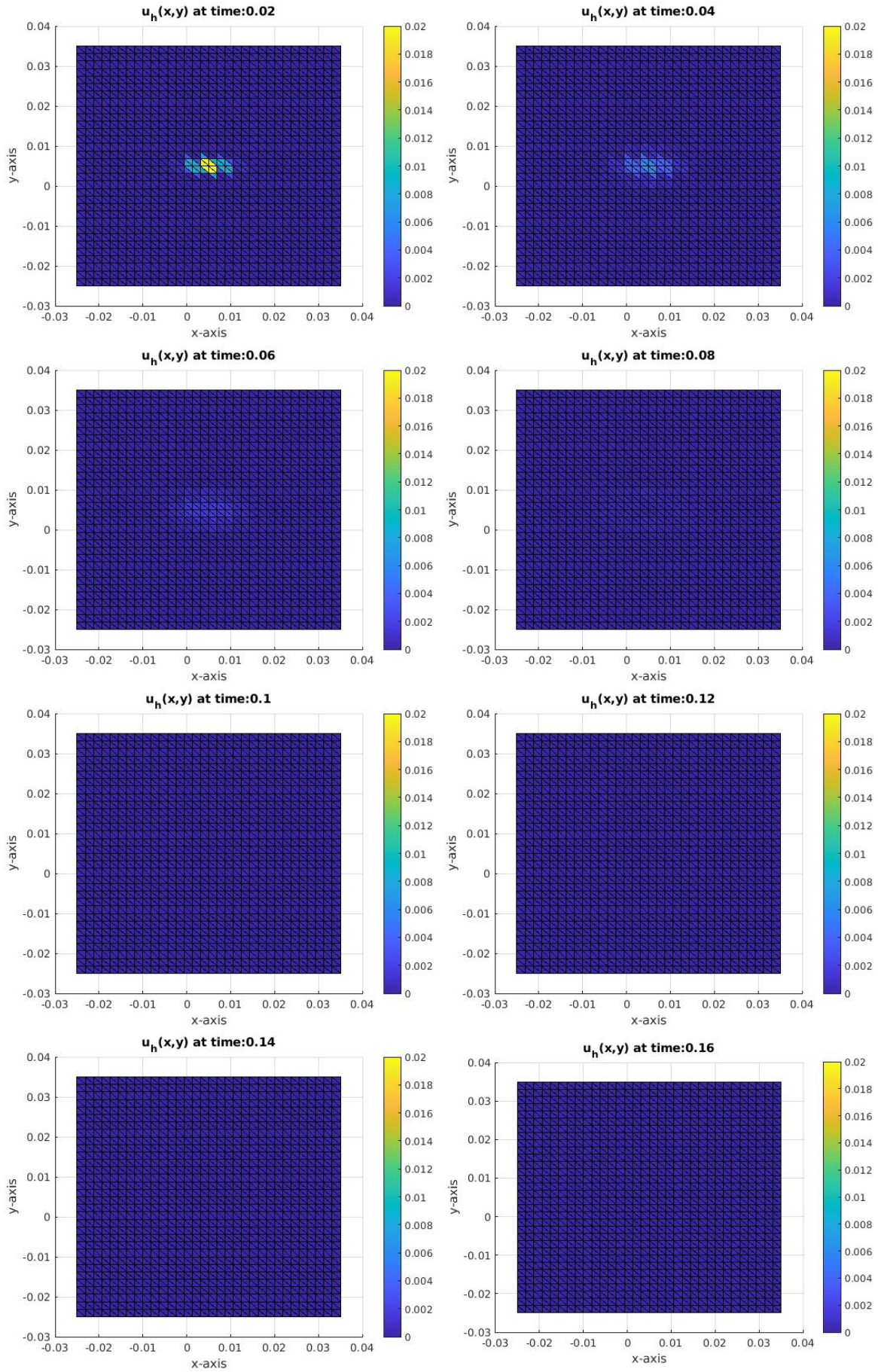
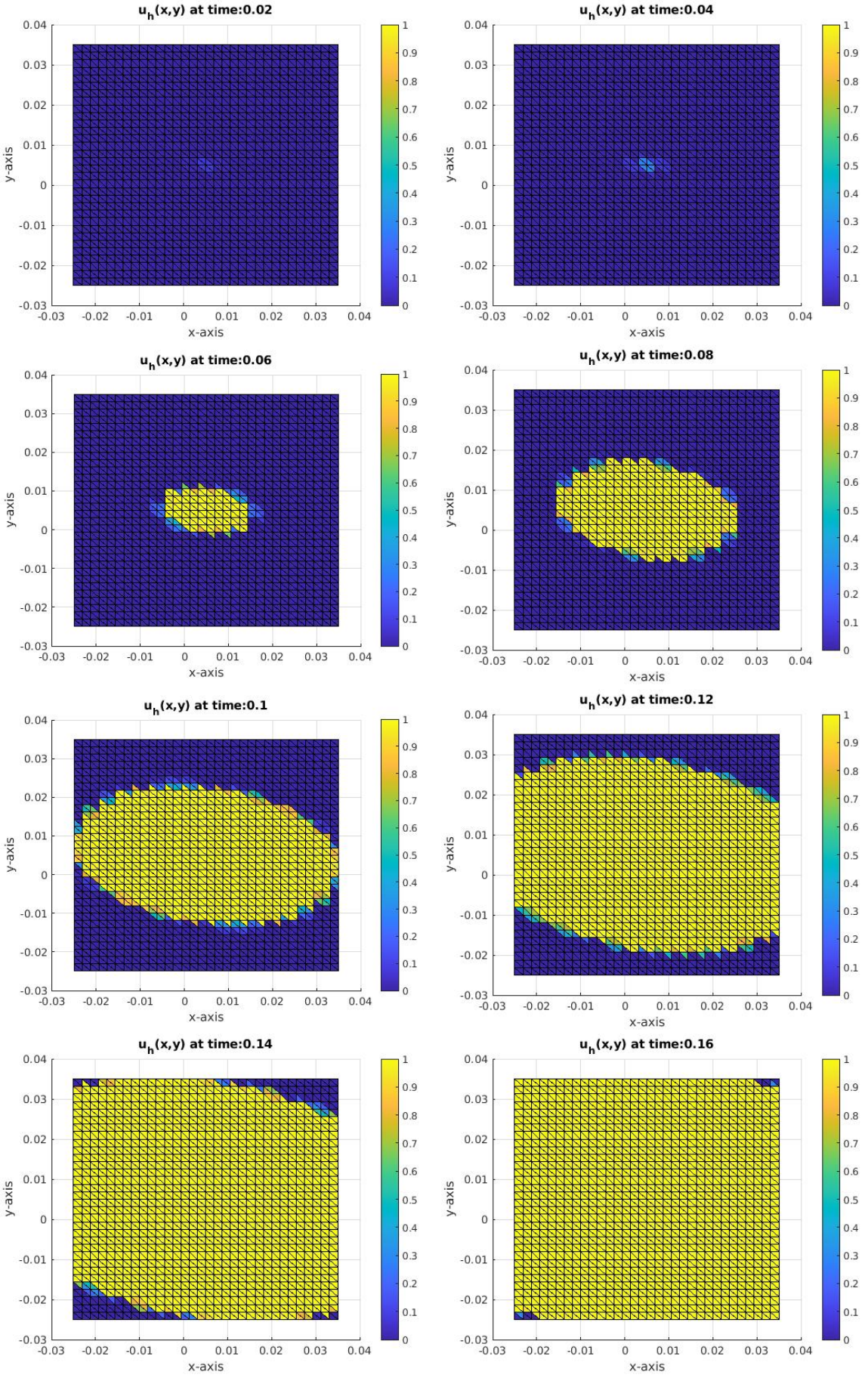


Figure 26: Test-case 1, computed snapshots for overflow case ($I = 700 \cdot 10^3 Am^{-3}$)



6.2.2 Second test-case

For this test case, the threshold for the activation is different. We choose:

- $I = 15000 \cdot 10^3 Am^{-3}$ as a suitable value for the *underflow* case.
- $I = 20000 \cdot 10^3 Am^{-3}$ as a suitable value for the *overflow* case.

The results related to this test-case are reported in Fig. 27 and Fig. 28.

6.2.3 Comments on the results

As expected, we can distinguish two different phenomena in both the test-cases:

- In *underflow* simulations, propagation is visible and the wave height (that is always below 1) decreases at every time-step. Then, the overall phenomena seems to be only diffusive.
- On the contrary, in *overflow* simulations, propagation is visible and the wave height keeps constant with value ≈ 1.

On the other hand, the main differences due to the choice of the *FitzHugh-Nagumo* parameters turn out to be:

- The threshold intensity (in test-case 2 this value is ten times the first).
- The speed of propagation (test-case 2 needs a lot more time to activate all the domain, 0.6 seconds are still not sufficient).
- The direction of propagation (shape in test-case 2 is more symmetric).

Most of these features were expected and successfully reflect the physical phenomenon. Indeed, a certain current intensity is needed to fully activate the myocytes. Otherwise, cells go soon back to their rest potential. Moreover, the stretched shape of the activated region is consistent with the anisotropy of the diffusion tensor and it physically shows the difference of conductivity between the tangential and normal direction of the fibers.

On the other hand, if we look at the potential cycle shown in Fig. 1, we notice some unexpected mismatches. First of all, the rest and activation values should be $-0.09V$ and $0.02V$ and not $0V$ and $1V$. However, it is probably due to the simplicity of the *FitzHugh-Nagumo* model that we adopted. Just from the analytical formulation is clear that $V_m = 0$ is an equilibrium value and not $V_m = -0.09$, again a refinement to the model is needed to get these more realistic values.
The second incongruence is the missed repolarization after the activation. In other words, the potential keeps the same value and it does not decrease back to 0 even after several time-steps. We think it is in part due to (again) the simplicity of the *FitzHugh-Nagumo* model, fact that is already observed in [6]. Here, it is, indeed, stated that *FitzHugh-Nagumo* model is not able to truly approximate the physical phenomenon, in particular for the *plateau* and *repolarization* phases.

Figure 27: Test-case 2, computed snapshots for underflow case ($I = 15000 \cdot 10^3 Am^{-3}$)

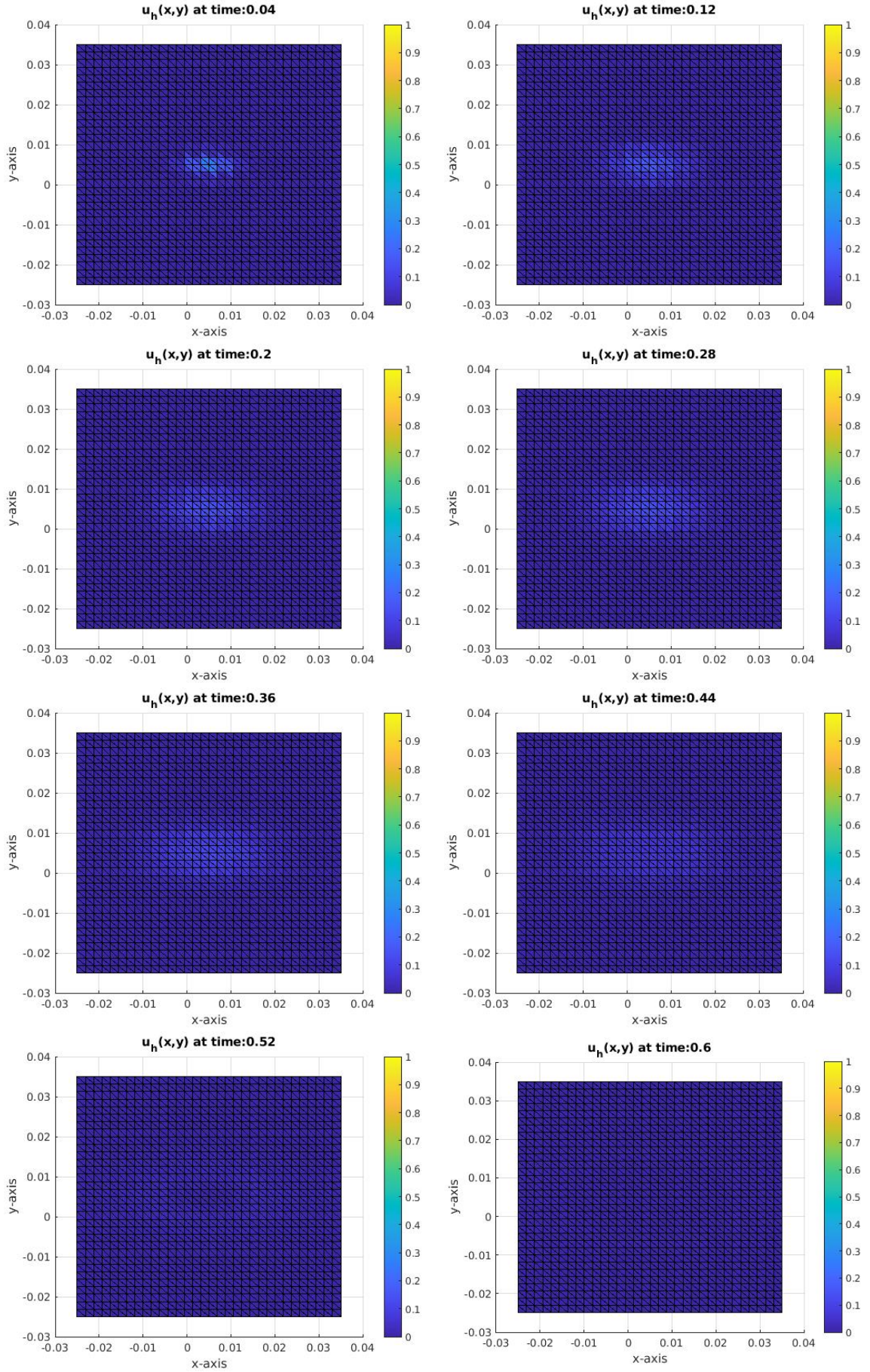
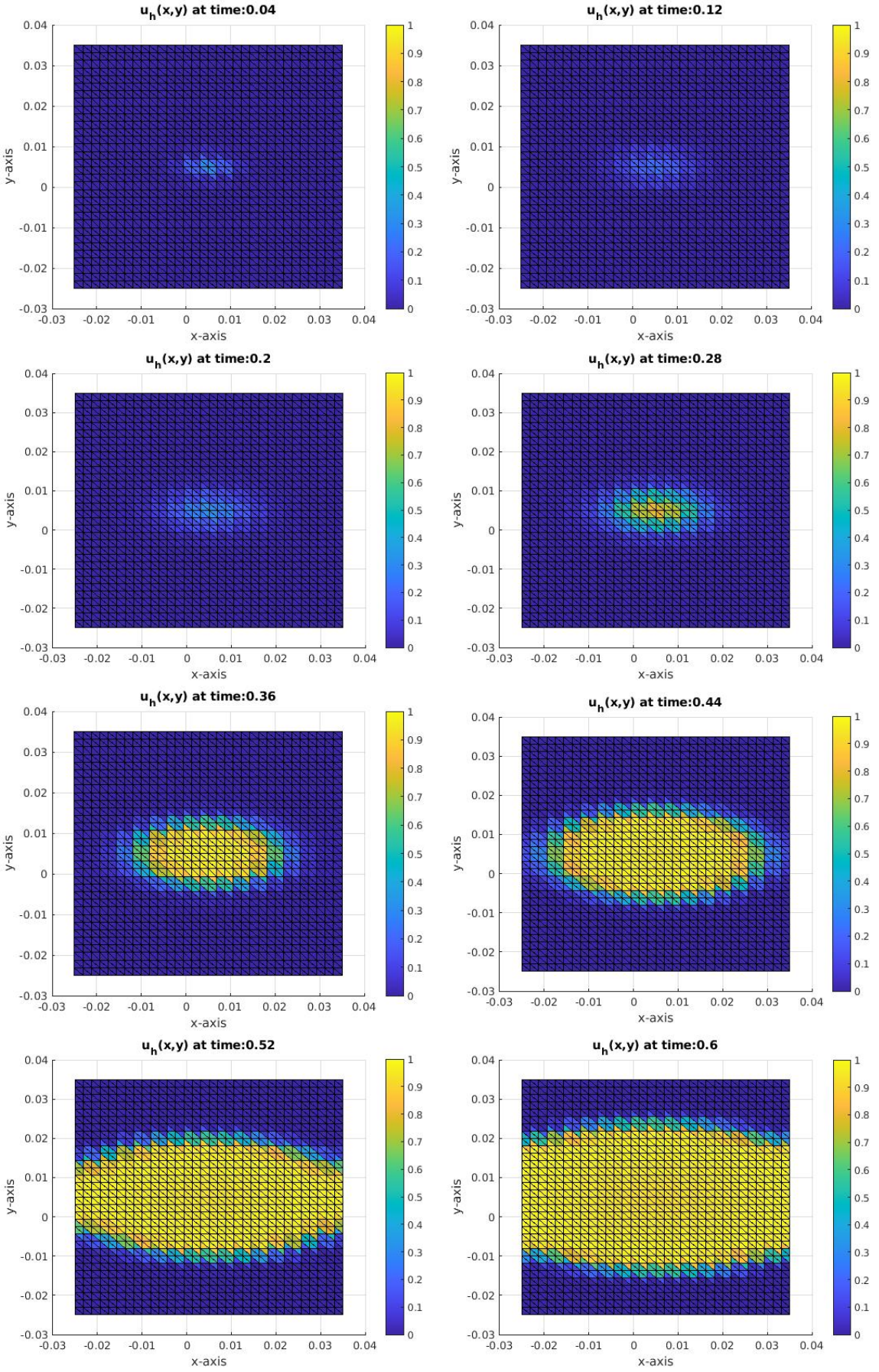


Figure 28: Test-case 2, computed snapshots for overflow case ($I = 20000 \cdot 10^3 Am^{-3}$)



7 Conclusion

In this project we have considered a DG approach for cardiac electro-physiology. After some motivations, we have validated the proposed numerical method through a basic error analysis and a couple of simulations aimed to re-create what happens in the human heart. From the former we obtained excellent validations: every result was indeed consistent with the theory.
We also achieved very good outcomes from the pseudo realistic simulations: they turned to be consistent with the physical phenomenon, except for the missed repolarization. We doubt that it is due to the numerical schemes, because of the excellent results in the error analysis. We instead suppose it is due to the extreme simplicity of the ionic model taken into consideration. Some inconsistencies may also be caused by a too coarse mesh, something that we could not avoid to get results in a reasonable time. Further researches could then consider a mesh-adaptivity study (even if it requires very powerful computers) and/or the adoption of other ionic models. We hope that this last point could be a springboard for future projects aiming at improving and generalize our results.

8 Appendix: numerical codes

8.1 dubiner_to_fem.m

```
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
   nln = femregion.nln;
   index = (ie-1)*nln*ones(nln,1) + [1:nln]';

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

8.2 fem_to_dubiner.m

```
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
   nln = femregion.nln;
   index = (ie-1)*nln*ones(nln,1) + [1:nln]';
   % loop over local degrees of freedom
   for i = 1 : 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 : 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
```

8.3 `main2D.m` (semi-implicit method)

```
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
```

8.4 `main2D.m` (Godunov operator-splitting method)

```
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);
```

```
10
11      [C] = assemble_nonlinear(femregion,Data,Vm0);
12
13      w1 = (1 -epsilon*gamma*dt)*w0 + epsilon*dt*Vm0;
14      B = MASS + [Ai, ZERO; ZERO, -Ae];
15      r = -MASSW*[w0;w0] + ((Cm/dt)*MASSW - [C, ZERO; ZERO, C])
16      *[Vm0;Vm0] + f1;
17
18      Vm0 = u1(1:ll) - u1(ll+1:end);
19      u0 = u1;
20      w0 = w1;
21  end
```

8.5 `main2D.m` (quasi-implicit operator-splitting method)

```
1   ZERO = sparse(ll,ll);
2
3   for t=dt:dt:T
4
5       [C] = assemble_nonlinear(femregion,Data,Vm0);
6       Q  = (ChiM*Cm/dt)*M + C - (epsilon*ChiM*dt)/(1+epsilon*gamma*dt)*M;
7       R  = (ChiM*Cm/dt)*M*Vm0 - (ChiM)/(1+epsilon*gamma*dt)*M*w0;
8
9       fi = assemble_rhs_i(femregion,neighbour,Data,t);
10      fe = assemble_rhs_e(femregion,neighbour,Data,t);
11      f1 = cat(1, fi, -fe);
12
13      B = [Q, -Q; Q, -Q] + [Ai, ZERO; ZERO, -Ae];
14      r = [R;R] + f1;
15
16      u1 = B \ r;
17
18      Vm1 = u1(1:ll)-u1(ll+1:end);
19
20      w1 = (w0 + epsilon*dt*Vm1)/(1+epsilon*gamma*dt);
21
22      f0 = f1;
23      Vm0 = u1(1:ll) - u1(ll+1:end);
24      u0 = u1;
25      w0 = w1;
26  end
```

8.6 `assign_phi_i.m`

```
1   function [A, b] = assign_phi_i (A, b, t, Data, femregion)
2
3   ...
4   ...
5   ...
```

```
6
7  if (Data.fem(1)=='P')
8
9      x = femregion.dof(1,1); % x-coordinate of the first dof point
10     y = femregion.dof(1,2); % y-coordinate of the first dof point
11     exact_coeff = eval(Data.exact_sol_i); % evaluation of exact sol
12
13
14 elseif (Data.fem(1)=='D')
15     x0=femregion.dof(3,1); % bottom-left corner of the first element
16     y0=femregion.dof(3,2);
17     h=femregion.dof(1,1)-femregion.dof(3,1); % length of the element
18
19     exact_coeff = 0;
20     index = 1;
21
22     ...
23     ...
24     ...
25
26     % the first coeff is the L2 scalar product of uh with the first
27     % basis function. To the get the right coeff, we compute the scalar
28     % product between the exact solution and the first basis function
29
30     for k = 1:length(w_2D) % loop over 2D quadrature points
31         %physical coords of the integration point
32         x = x0 + h*node_2D(k,1);
33         y = y0 + h*node_2D(k,2);
34         evalsol = eval(Data.exact_sol_i);
35         exact_coeff = exact_coeff + evalsol*phi_dub(1,k,index).*w_2D(k);
36     end
37
38 end
39
40
41 % we change the system coefficients to impose u(1)=exact_coeff
42 Nh = length(b);
43 b = b - A(:,1)*exact_coeff;
44 b(1) = exact_coeff;
45 A(:,1) = zeros(Nh,1);
46 A(1,:) = zeros(1,Nh);
47 A(1,1) = 1;
```

8.7 assign_null_average.m

```
1  function [A, b] = assign_null_average (A, b, Data, femregion)
2
3  Nh = length(b)/2;
4
5  if (Data.fem(1)=='P')
```

```
6
7       ...
8       ...
9       ...
10
11      for k = 1:length(w_2D)
12          coeff = coeff + phi_fem(1,k,1).*w_2D(k);
13      end
14
15      for i = 1:Nh
16          A(i,2*Nh+1)=coeff;
17          A(2*Nh+1,i)=coeff;
18      end
19
20
21  elseif (Data.fem(1)=='D')
22
23      ...
24      ...
25      ...
26
27      for p = 1:femregion.nln
28          p_int = 0;
29          for k = 1:length(w_2D)
30              p_int = p_int + phi_dub(1,k,p).*w_2D(k);
31          end
32          coeff(p)=p_int;
33      end
34
35      for i = 1:femregion.nln:Nh
36          A(2*Nh+1,i:i+femregion.nln-1)=coeff';
37          A(i:i+femregion.nln-1,2*Nh+1)=coeff;
38      end
39
40  end
41
42  b = [b;0];
```

References

[1] V. Anaya et al. "A Virtual Element Method for a Nonlocal FitzHugh-Nagumo Model of Cardiac Electrophysiology". In: *IMA Journal of Numerical Analysis* 40 (2020), pp. 1544–1576.

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

[3] 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.

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

[5] 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.

[6] P. Colli Franzone and L. F. Pavarino. "A parallel solver for reaction-diffusion systems in computational electrocardiology". In: *Mathematical Models and Methods in Applied Sciences* 14 (2004), pp. 883–911.

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

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

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

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

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

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

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

[14] 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.

[15] 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.