

# A $p$ -adaptive polytopal discontinuous Galerkin method for high-order approximation of brain electrophysiology\*

Caterina B. Leimer Saglio<sup>1</sup>, Stefano Pagani<sup>1</sup>, and Paola F. Antonietti<sup>1</sup>

<sup>1</sup>*MOX-Dipartimento di Matematica, Politecnico di Milano, Piazza Leonardo da Vinci 32, Milan, 20133, Italy*

May 5, 2025

## Abstract

Multiscale mathematical models have shown great promise in computational brain electrophysiology but are still hindered by high computational costs due to fast dynamics and complex brain geometries, requiring very fine spatio-temporal resolution. This paper introduces a novel  $p$ -adaptive discontinuous Galerkin method on polytopal grids (PolyDG) coupled with Crank–Nicolson time integration to approximate such models efficiently. The  $p$ -adaptive method enhances local accuracy via dynamic, element-wise polynomial refinement/de-refinement guided by a-posteriori error estimators. A novel clustering algorithm automatizes the selection of elements for adaptive updates, further improving efficiency. A wide set of numerical tests, including epileptic seizure simulations in a sagittal section of a human brain stem, demonstrate the method’s ability to reduce computational load while maintaining the accuracy of the numerical solution in capturing the dynamics of multiple wavefronts.

## 1 Introduction

Biophysical modeling and numerical simulation are pivotal in improving our understanding of biological processes. In silico approaches now offer tools to accelerate medical research and translate it into the clinic with novel personalized therapies [1]. Brain modeling constitutes a fundamental pillar of computational neuroscience, where mathematical laws, particularly partial differential equations, serve as essential tools for describing the spatiotemporal evolution of interconnected biophysical processes and the intricate interplay with the brain’s complex structures, [2, 3].

The field of computational brain electrophysiology is still advancing toward this objective: despite the availability of mathematical multiscale models and advanced numerical methods [4, 5, 6, 3, 7, 8, 9, 10, 11], computational efficiency presents a significant barrier to broader application. Three factors significantly increase the computational burden: the spatial and temporal constraints associated with the cellular scale of the problem, the non-linearity of the coupled multiscale models, and the geometric complexity. Consequently, the number of degrees of freedom required to accurately discretize the corresponding mathematical problem when considering tissue- or organ-scale simulations increases dramatically. More precisely, the wavefronts of neuronal action potentials are characterized by steep and rapid profiles evolving on the millisecond time scale and localized at the synaptic level [12, 13], i.e., the connections between different neurons. The analysis of physiological and pathological phenomena must consider various scales that characterize neuronal activity, ranging from the membrane scale [14], where chemical interactions take place across the cell membrane, to the entire brain. Furthermore, epileptic seizures can last for hours [15], making the problem multiscale in both space and time.

From a modeling perspective, computational neuronal electrophysiology relies on nonlinear ionic models derived from the Hodgkin–Huxley framework [16, 17, 18], which describe the dynamics of transmembrane potential and the associated ionic concentrations. At the tissue and organ level, these ionic models are coupled with the

---

\***Funding:** PFA is partially funded by the European Union (ERC SyG, NEMESIS, project number 101115663). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. PFA is partially supported by ICSC—Centro Nazionale di Ricerca in High Performance Computing, BigData, and Quantum Computing funded by European Union—NextGeneration EU. CBL’s has been funded by the National Recovery and Resilience Plan (NRRP), Mission 4, Component 1 – Investment 3.4, and Investment 4.1, funded by the European Union. The present research is part of the activities of ”Dipartimento di Eccellenza 2023-2027”. The authors are members of INdAM-GNCS.

monodomain or bidomain equations [3, 19] to model the propagation of transmembrane electric potential. Furthermore, the brain’s geometry is highly complex, and its material properties are strongly anisotropic, which adds further challenges in the discretization step. The numerical approximation of the resulting coupled system of partial and ordinary differential equations typically leads to a large (monolithic or segregated) algebraic system, featuring a prohibitively large number of degrees of freedom. These computational challenges are also common in cardiac electrophysiology, where a wide strand of discretization approaches are available in the literature, see, e.g., [20, 21, 22, 23, 24, 25, 26, 27, 28, 19]; we also refer to the comprehensive monographs [29, 30] and the reference therein. In the specific context of high-order discretizations of electrophysiology models, we mention, e.g., spectral element methods [31, 32, 33], isogeometric analysis [34, 35, 36, 37], Domain Decomposition schemes [38], and high-order parallel-in-time approaches [39]. High-order discontinuous Galerkin discretizations of electrophysiology models [40, 41] offer a highly flexible framework for implementing  $h$ -,  $p$ -, and  $hp$ -adaptivity strategies, thanks to the “local” nature of the discretization spaces. These methods enable enhanced accuracy while reducing the number of degrees of freedom and computational costs. In particular, discontinuous Galerkin methods formulated on polytopal meshes (PolyDG) [42, 43, 44, 45] naturally accommodate complex geometries, benefiting from the inherent flexibility of agglomerated grid structures. For brain tissue electrophysiology, in [11] we proposed a high-order discontinuous PolyDG method to handle inherent geometrical complexities. Nevertheless, the computational costs for these simulations remain substantial, as fine meshes and small time steps are required to currently capture the wavefronts. Given the inherited complexities of the problem at hand,  $p$ - [46] or  $hp$ -adaptive methods can provide an effective approach to lessen simulation costs while preserving the accuracy, see, e.g., [46, 47, 48] for triangular and [49, 42] for polytopal meshes.

In this work, we propose a novel  $p$ -adaptive PolyDG method coupled with Crank–Nicolson time stepping for the numerical discretization of the monodomain equation coupled with the Barreto–Cressman neuronal ionic model. The proposed  $p$ -adaptive algorithm is driven by suitable a posteriori estimators and allows local and dynamic high-order accuracy in the numerical approximation of the solution while efficiently reducing the total number of degrees of freedom. To further enhance computational efficiency we also propose a novel technique based on  $k$ -means centroids for the automatic and dynamic detection of the subset of mesh elements candidate for refinement or coarsening of the local polynomial degree. We demonstrate the practical performance of the proposed  $p$ -adaptive method with extensive numerical experiments, including benchmark test cases and simulations of epileptic seizure propagation in a sagittal section of the human brainstem. Our numerical results confirm the capability of the proposed method to substantially reduce the total number of degrees of freedom, while preserving the accuracy required to capture such complex wavefront dynamics. We point out that, while the proposed approach has been developed and validated within the framework of the monodomain equation coupled with the Barreto–Cressman model, its design is sufficiently general to be extended to other models of (brain) electrophysiology, including different ionic models or alternative macroscopic descriptions.

The remainder of the manuscript is organized as follows: In Section 2, we introduce the mathematical model of brain electrophysiology under investigation. Section 3 presents the semi discrete formulation within the PolyDG framework and the fully discrete method based on employing the Crank–Nicolson time marching scheme. Section 4 introduces and discusses in detail the proposed  $p$ -adaptive method, including a detailed discussion on implementation details. In Section 5, we demonstrate the practical performance of the proposed  $p$ -adaptive method through a series of benchmark test cases. Finally, in Section 6 we illustrate the effectiveness of the proposed  $p$ -adaptive method in simulating physio-pathological scenarios in brain electrophysiology.

## 2 The mathematical model

In this section, we present the mathematical formulation of the brain electrophysiology model. Specifically, we consider the monodomain equation [50, 51] coupled with suitable neuronal ionic models [52, 18].

Given an open, bounded, polygonal domain  $\Omega \in \mathbb{R}^d$ , ( $d = 2, 3$ ) and a final time  $T > 0$ , we introduce the transmembrane potential  $u = u(\mathbf{x}, t)$  with  $u : \Omega \times [0, T] \rightarrow \mathbb{R}$ , and the vector  $\mathbf{y} = \mathbf{y}(\mathbf{x}, t)$  with  $\mathbf{y} : \Omega \times [0, T] \rightarrow \mathbb{R}^n, n \geq 1$ , containing the ion concentrations and gating variables of the ionic model. The coupled multiscale problem reads as follows:

For any time  $t \in (0, T]$ , find  $u = u(\mathbf{x}, t)$  and  $\mathbf{y} = \mathbf{y}(\mathbf{x}, t)$  such that:

$$\begin{cases} \chi_m C_m \frac{\partial u}{\partial t} - \nabla \cdot (\boldsymbol{\Sigma} \nabla u) + \chi_m f(u, \mathbf{y}) = I_{\text{ext}} & \text{in } \Omega \times (0, T], \\ \frac{\partial \mathbf{y}}{\partial t} + \mathbf{m}(u, \mathbf{y}) = \mathbf{0} & \text{in } \Omega \times (0, T], \\ \boldsymbol{\Sigma} \nabla u \cdot \mathbf{n} = 0 & \text{on } \partial\Omega \times (0, T], \\ u(0) = u^0, \mathbf{y}(0) = \mathbf{y}^0 & \text{in } \Omega. \end{cases} \quad (1)$$

Here,  $\boldsymbol{\Sigma}$  represents the conductivity tensor, assumed to be constant in time and piecewise in space. Under the isotropic assumption, it is defined as  $\boldsymbol{\Sigma} = \sigma \mathbb{1}$ , where  $\sigma$  is the parameter defining the conductivity magnitude. In the first two test cases, we employ a fully isotropic conductivity tensor to focus only on the  $p$ -adaptive algorithm. Nonetheless, in the context of brain electrophysiology (test case 3), the 2D conductivity tensor is usually defined as:

$$\boldsymbol{\Sigma} = \sigma_a \mathbf{l} \otimes \mathbf{l} + \sigma_n \mathbf{n} \otimes \mathbf{n} = \sigma_l \mathbb{1} + (\sigma_n - \sigma_l) \mathbf{n} \otimes \mathbf{n}, \quad (2)$$

where  $\sigma_l$  represents the conductivity along the principal axonal direction,  $\sigma_n$  the conductivity in the orthogonal direction,  $\mathbf{l} = \mathbf{l}(x)$  the axonal fibers direction in the point  $x \in \Omega$  and  $\mathbf{n} = \mathbf{n}(x)$  its normal. We applied our  $p$ -adaptive algorithm to this context in the last test case. In all the test cases, we assume homogeneous Neumann boundary conditions and, finally, we introduce suitable initial conditions  $u^0$  and  $\mathbf{y}^0$ .

For the modeling of ionic currents, we employ the Barreto-Cressman ionic model, a conductance-based model that describes a neuron's membrane potential and the dynamic interactions between intra- and extracellular ion concentrations [18]. Through parameters' variations, it can effectively represent neuronal bursting, a phase of epilepsy marked by rapid transmembrane potential spiking, as well as the classical action potential. A detailed derivation of this model can be found in [17, 18]. The variables of the models are three different ionic concentrations, namely intracellular sodium  $s(t) = [\text{Na}]_i(t)$ , extracellular potassium  $k(t) = [\text{K}]_o(t)$ , and intracellular calcium  $c(t) = [\text{Ca}]_i(t)$ , and three gating variables  $g^s, g^k, g^c$  that control the opening and closing of ion channels:

$$\mathbf{y}(\mathbf{x}, t) = [s(\mathbf{x}, t), k(\mathbf{x}, t), c(\mathbf{x}, t), g^s(\mathbf{x}, t), g^k(\mathbf{x}, t), g^c(\mathbf{x}, t)]^\top, \quad (3)$$

with  $\mathbf{y}(\mathbf{x}, t) : \Omega \times [0, T] \rightarrow \mathbb{R}^6$ . The ionic reaction term of the model in Equation (1) is defined as follows:

$$f(u, \mathbf{y}) = I_{\text{Na}}(u, \mathbf{y}) + I_{\text{K}}(u, \mathbf{y}) + I_{\text{Cl}}(u, \mathbf{y}), \quad (4)$$

where the currents of sodium ( $I_{\text{Na}}$ ), potassium ( $I_{\text{K}}$ ), and chloride ( $I_{\text{Cl}}$ ) ions are:

$$\begin{aligned} I_{\text{Na}}(u, \mathbf{y}) &= (G_{\text{NaL}} + G_{\text{Na}}(g^s(t))^3 g^k(t)) (u(t) - E_{\text{Na}}), \\ I_{\text{K}}(u, \mathbf{y}) &= \left( G_{\text{K}}(g^c(t))^4 + G_{\text{AHP}} \frac{c(t)}{1 + c(t)} + G_{\text{KL}} \right) (u(t) - E_{\text{K}}), \\ I_{\text{Cl}}(u, \mathbf{y}) &= G_{\text{CIL}}(u(t) - E_{\text{Cl}}). \end{aligned}$$

Finally, the right-hand side of the ionic model is given by:

$$\mathbf{m}(u, \mathbf{y}) = \begin{bmatrix} \frac{c}{80} + G_{\text{Ca}} \frac{0.002(u - E_{\text{Ca}})}{1 + \exp\left(-\frac{25+u}{2.5}\right)} \\ \frac{1}{\tau} (I_{\text{diff}} - 14I_{\text{pump}} - I_{\text{glia}} + 7\gamma I_{\text{K}}) \\ \frac{1}{\tau} (\gamma I_{\text{Na}} - 3I_{\text{pump}}) \\ 3 \frac{g - g_\infty}{\tau_g} \end{bmatrix}.$$

For the weak formulation of the problem, we introduce the Sobolev space  $V = H^1(\Omega)$ , and we employ a standard definition of the scalar product in  $L^2(\Omega)$ , denoted by  $(\cdot, \cdot)_\Omega$ . The induced norm is denoted by  $\|\cdot\|$ . We remind that for vector-valued and tensor-valued functions, the definition extends component-wise [53]. Starting from Equation (1), we introduce the ionic component and the dynamics component of the ionic model as:

$$\begin{aligned} a(u, v) &= (\boldsymbol{\Sigma} \nabla u, \nabla v)_\Omega, & r_{\text{ion}}(u, \mathbf{y}, v) &= (f(u, \mathbf{y}), v)_\Omega \quad \forall v \in V, \\ r_{\text{m}}(u, \mathbf{y}, \mathbf{w}) &= (\mathbf{m}(u, \mathbf{y}), \mathbf{w})_\Omega \quad \forall \mathbf{w} \in [V]^n. \end{aligned} \quad (5)$$

We assume that the forcing terms, physical parameters, and initial conditions in Equation (1) are sufficiently regular, *i.e.*:  $f(u, \mathbf{y}) \in L^2(0, T; L^2(\Omega))$ ,  $I^{\text{ext}}(u, \mathbf{y}) \in L^2(0, T; L^2(\Omega))$ ,  $\mathbf{m}(u, \mathbf{y}) \in [L^2(0, T; L^2(\Omega))]^n$ ,  $\chi_m$  and  $C_m \in L^{\infty}_+(\Omega)$  where  $L^{\infty}_+(\Omega) := \{v \in L^{\infty}(\Omega) : v \geq 0 \text{ a.e. in } \Omega\}$ ,  $u^0 \in L^2(\Omega)$  and  $\mathbf{y}^0 \in [L^2(\Omega)]^n$ . The weak formulation of the problem (1) reads:  $\forall t \in (0, T]$  find  $u(t) \in V$ ,  $\mathbf{y}(t) \in V^n$  such that:

$$\begin{cases} \chi_m C_m \left( \frac{\partial u(t)}{\partial t}, v \right)_{\Omega} + a(u(t), v) + \chi_m r_{\text{ion}}(u(t), v) = (I^{\text{ext}}, v)_{\Omega} & \forall v \in V, \\ \left( \frac{\partial \mathbf{y}(t)}{\partial t}, \mathbf{w} \right)_{\Omega} + r_m(u(t), \mathbf{y}(t), \mathbf{w}) = 0 & \forall \mathbf{w} \in V^n, \\ u(0) = u^0, \mathbf{y}(0) = \mathbf{y}^0 & \text{in } \Omega. \end{cases} \quad (6)$$

### 3 PolyDG semi-discretization and fully-discrete formulation

We now present the PolyDG semi-discrete formulation of the problem described in Equation (6). Let  $\mathcal{T}_h$  represent a polytopal mesh partition of the domain  $\Omega$ , consisting of disjoint elements  $K$ . For each element  $K$ , we define its diameter as  $h_K$  and set  $h = \max_{K \in \mathcal{T}_h} h_K < 1$ . The interfaces are defined as the intersections of the  $(d - 1)$ -dimensional facets of neighboring elements. For  $d = 2$ , the interfaces of a given element  $K$  are always segments. We denote by  $\mathcal{F}_h^I$  the union of all interior faces contained within  $\Omega$  and by  $\mathcal{F}_h^N$  those lying on the boundary  $\partial\Omega$ . In the following we assume that the underlying grid is *uniformly polytopal shape regular* in the sense of [54].

We define  $\mathbb{P}^p(K)$  as the space of polynomials of degree at most  $p \geq 1$  over the element  $K$  and the discontinuous finite element space as:

$$V_h^{\text{DG}} = \{v_h \in L^2(\Omega) : v_h|_K \in \mathbb{P}^{p_K}(K) \quad \forall K \in \mathcal{T}_h\},$$

with local degree  $p_K \geq 1$  for any  $K \in \mathcal{T}_h$ . Let  $F \in \mathcal{F}_h^I$  be the face shared by the elements  $K^{\pm}$ , and let  $\mathbf{n}^{\pm}$  denote the normal unit vectors on the face  $F$  pointing outward to  $K^{\pm}$ . For regular enough scalar-valued function  $v$  and a vector-valued function  $\mathbf{q}$ , the trace operators are defined as follows [55]:

$$\begin{aligned} \llbracket v \rrbracket &= \frac{1}{2}(v^+ + v^-), & \llbracket v \rrbracket &= v^+ \mathbf{n}^+ + v^- \mathbf{n}^-, & \text{on } F \in \mathcal{F}_h^I, \\ \llbracket \mathbf{q} \rrbracket &= \frac{1}{2}(\mathbf{q}^+ + \mathbf{q}^-), & \llbracket \mathbf{q} \rrbracket &= \mathbf{q}^+ \cdot \mathbf{n}^+ + \mathbf{q}^- \cdot \mathbf{n}^-, & \text{on } F \in \mathcal{F}_h^I. \end{aligned}$$

Here, the superscripts  $\pm$  indicate the traces of these functions on  $F$  taken in the interiors of  $K^{\pm}$ , respectively. Moreover, we assume for the derivation of the formulation that the local polynomial degree vector  $\mathbf{p} = \{p_K\}_{K \in \mathcal{T}_h}$  is given. This assumption allows us to give the following consistent definition of the penalization parameter  $\eta : \mathcal{F}_h^I \cup \mathcal{F}_h^D \rightarrow \mathbb{R}_+$ :

$$\eta = \eta(\mathbf{p}, h, \Sigma) = \eta_0 \begin{cases} \{\Sigma_K\}_A \frac{\{p_K\}_A}{\{h_K\}_H} & \text{on } F \in \mathcal{F}_h^I, \\ \Sigma_K \frac{p_K}{h_K} & \text{on } F \in \mathcal{F}_h^D, \end{cases} \quad (7)$$

which depends explicitly on both the local degrees and the mesh size. In Equation (7) we set  $\Sigma_K = \|\sqrt{\Sigma}|_K\|_{L^2(K)}^2$  and we consider both the harmonic average operator  $\{\cdot\}_H$ , that is  $\{h\}_H = \frac{2h_+h_-}{h_+ + h_-}$ , and the arithmetic average operator  $\{\cdot\}_A$ , that is  $\{p\}_H = \frac{p_+ + p_-}{2}$  on  $F \in \mathcal{F}_h^I$ . When applying our  $p$ -adaptive method (described in the following section), the polynomial degree vector  $\mathbf{p} = \{p_K\}_{K \in \mathcal{T}_h}$  is updated at each time step. The penalization parameter  $\eta$  is adjusted accordingly, while  $\eta_0$  is chosen large enough to ensure stability. This setting enable us to introduce the following bilinear form  $\mathcal{A}(\cdot, \cdot) : V_h^{\text{DG}} \times V_h^{\text{DG}} \rightarrow \mathbb{R}$ :

$$\mathcal{A}(u, v) = \int_{\Omega} \Sigma \nabla_h u \cdot \nabla_h v \, dx + \sum_{F \in \mathcal{F}_h^I} \int_F (\eta \llbracket u \rrbracket \cdot \llbracket v \rrbracket - \llbracket \Sigma \nabla u \rrbracket \cdot \llbracket v \rrbracket - \llbracket u \rrbracket \cdot \llbracket \Sigma \nabla v \rrbracket) \, d\sigma \quad \forall u, v \in V_h^{\text{DG}}, \quad (8)$$

where  $\nabla_h$  is the element-wise gradient. The semi-discrete formulation of problem in Equation (1) reads:

For any  $t \in (0, T]$ , find  $(u_h(t), \mathbf{y}_h(t)) \in V_h^{\text{DG}} \times [V_h^{\text{DG}}]^n$  such that:

$$\begin{cases} \chi_m C_m \left( \frac{\partial u_h(t)}{\partial t}, v_h \right)_\Omega + \mathcal{A}(u_h(t), v_h) + \chi_m (f(u_h(t), \mathbf{y}_h(t)), v_h)_\Omega = (I_h^{\text{ext}}, v_h)_\Omega & \forall v_h \in V_h^{\text{DG}}, \\ \left( \frac{d\mathbf{y}_h(t)}{dt}, \mathbf{w}_h \right)_\Omega + (\mathbf{m}(u_h(t), \mathbf{y}_h(t)), \mathbf{w}_h)_\Omega = 0 & \forall \mathbf{w}_h \in [V_h^{\text{DG}}]^n, \\ u_h(0) = u_h^0, \mathbf{y}_h(0) = \mathbf{y}_h^0 & \text{in } \Omega. \end{cases} \quad (9)$$

As a consequence of our assumptions, we denote the dimension of the discrete space as  $N_h(\mathbf{p})$ , to make explicit its dependence on the vector of local polynomial distribution. Let  $N_h(\mathbf{p})$  be the dimension of  $V_h^{\text{DG}}$  and let  $(\varphi_j)_{j=0}^{N_h(\mathbf{p})}$  be a suitable basis for  $V_h^{\text{DG}}$ , then  $u_h(t) = \sum_{j=0}^{N_h(\mathbf{p})} U_j(t) \varphi_j$  and  $y_l(t) = \sum_{j=0}^{N_h(\mathbf{p})} Y_j^l(t) \varphi_j$  for all  $l = 1, \dots, n$ . We denote  $\mathbf{U} \in \mathbb{R}^{N_h(\mathbf{p})}$ ,  $\mathbf{Y}_l \in \mathbb{R}^{N_h(\mathbf{p})}$  for all  $l = 1, \dots, n$  and  $\mathbf{Y} = [\mathbf{Y}_1, \dots, \mathbf{Y}_n]^\top$ . We define the matrices:

$$\begin{aligned} [\mathbf{M}]_{ij} &= (\varphi_i, \varphi_j)_\Omega, & \text{(Mass matrix), } & i, j = 1, \dots, N_h(\mathbf{p}) \\ [\mathbf{F}]_j &= (I^{\text{ext}}, \varphi_j)_\Omega, & \text{(Forcing term), } & j = 1, \dots, N_h(\mathbf{p}) \\ [\mathbf{I}(u, \mathbf{y})]_j &= (f(u, \mathbf{y}), \varphi_j)_\Omega, & \text{(Non-linear ionic forcing term), } & j = 1, \dots, N_h(\mathbf{p}) \\ [\mathbf{G}_l(u, \mathbf{y})]_j &= (\mathbf{m}_l(u, \mathbf{y}), \varphi_j)_\Omega, & \text{(Dynamics of the ionic model), } & j = 1, \dots, N_h(\mathbf{p}), l = 1, \dots, n \\ [\mathbf{A}]_{ij} &= \mathcal{A}(\varphi_i, \varphi_j) & \text{(Stiffness matrix), } & i, j = 1, \dots, N_h(\mathbf{p}). \end{aligned} \quad (10)$$

We are now ready to introduce the fully-discrete formulation. We partition the interval  $[0, T]$  into  $N$  sub-intervals  $(t^{(k)}, t^{(k+1)})$ , each of length  $\Delta t$ , such that  $t^{(k)} = k\Delta t$  for  $k = 0, \dots, N-1$ . For temporal discretization, we adopt the second-order Crank-Nicolson scheme [56], which handles the linear part (diffusion term) in a semi-implicit way, while the ion term in a fully explicit way. Given the initial conditions  $\mathbf{U}_0$  and  $\mathbf{Y}_0$ , the discrete scheme is: Find  $\mathbf{U}^{(k+1)} = \mathbf{U}(t^{(k+1)})$  and  $\mathbf{Y}^{(k+1)} = \mathbf{Y}(t^{(k+1)})$  for  $k = 0, \dots, N-1$ , such that:

$$\begin{cases} \left( \chi_m C_m \mathbf{M} + \frac{\Delta t}{2} \mathbf{A} \right) \mathbf{U}^{(k+1)} = \left( \chi_m C_m \mathbf{M} - \frac{\Delta t}{2} \mathbf{A} \right) \mathbf{U}^{(k)} - \chi_m \Delta t \mathbf{I}^{(k+1)} + \Delta t \mathbf{F}^{(k+1)}, \\ \mathbf{Y}^{(k+1)} = \mathbf{Y}^{(k)} - \Delta t \mathbf{G}^{(k)}, \\ \mathbf{U}^0 = \mathbf{U}_0, \mathbf{Y}^0 = \mathbf{Y}_0, \end{cases}$$

where the ionic current at step  $k+1$  is computed as follows:

$$\mathbf{I}^{(k+1)} = \frac{3}{2} \mathbf{I}^{(k)} - \frac{1}{2} \mathbf{I}^{(k-1)}. \quad (11)$$

## 4 $p$ -adaptive PolyDG framework

In practical applications, the solution of the problem in Equation (1) is characterized by action potentials with extremely steep and fast wavefronts that necessitate very fine spatial and temporal discretizations for their accurate numerical discretization. However, at a specific time instant, these traveling action potentials are localized in a specific portion of the domain, favoring adaptive numerical discretizations. These latter can accurately track the wavefront while removing unnecessary degrees of freedom, retaining the accuracy of the solution while reducing computational efforts. This section presents a spatial  $p$ -adaptive strategy based on the PolyDG method that locally adapts the polynomial order, using high-order discretization only in specific domain regions where wavefront tracking is needed. Unlike spatial adaptivity methods that rely on re-meshing, which demands significant computational resources and integration with a mesh generator,  $p$ -adaptivity avoids re-meshing during computation. An element-wise error indicator guides this flexibility in local polynomial degree selection [57] (described in Section 4.1). The modification of the polynomial degree is then automatically driven by this local indicator and a suitable threshold numerically determined at the beginning of the simulation (see Section 4.1.1). In Section 4.2, we present the complete algorithm that automatically uses higher-order discretizations only where needed to reduce the computational costs and reduce the total number of degrees of freedom while maintaining the accuracy required for accurate wavefront tracking.

### 4.1 $A$ -posteriori indicator

In this Section, we discuss the  $a$ -posteriori error indicator to determine/update the local polynomial order. We employ the comprehensive  $a$ -posteriori indicator proposed in [44] and defined for each time  $t$  and on each element

$K \in \mathcal{T}_h$  as:

$$\tau_K^2(\cdot, \cdot) = \tau_{K,r}^2(\cdot, \cdot) + \tau_{K,n}^2(\cdot) + \tau_{K,j}^2(\cdot) + \tau_{K,t}^2(\cdot) + \mathcal{O}_{K,r}^2, \quad (12)$$

where the components represent the norm of the residual ( $\tau_{K,r}^2$ ), the jump component of the normal gradient ( $\tau_{K,n}^2$ ), the jump component ( $\tau_{K,j}^2$ ), and the tangential component of the gradient ( $\tau_{K,t}^2$ ) of the numerical solution, respectively. The local data oscillations on the element  $K$ , defined for the forcing term ( $\mathcal{O}_{K,r}^2$ ) is instead given by the error generated by the  $L_2$ -projection of the forcing term. In the following, we define all the terms appearing in Equation (12) and refer to [58] for more details.

Let  $u_h^{(k)} \in V_h^{DG}$  and  $\mathbf{y}_h^{(k)} \in [V_h^{DG}]^n$  be the approximated solution at the time  $t^{(k)}$ ,  $k = 1, 2, \dots$ . Then, the first four components of the indicator (12) are defined as

$$\tau_{K,r}(u_h^{(k)}, \mathbf{y}_h^{(k)}) = \|h_K \mathcal{R}(u_h^{(k)}, \mathbf{y}_h^{(k)})\|_{L^2(K)}, \quad (13)$$

$$\tau_{K,n}(u_h^{(k)}) = \left( \|\sqrt{h_K} [(\boldsymbol{\Sigma} \nabla_h u_h^{(k)}) \cdot \mathbf{n}_K]\|_{L^2(\partial K) \cap \mathcal{F}_h^I}^2 + \|\sqrt{h_K} (\boldsymbol{\Sigma} \nabla_h u_h^{(k)}) \cdot \mathbf{n}_K\|_{L^2(\partial K) \cap \mathcal{F}_h^N}^2 \right)^{\frac{1}{2}}, \quad (14)$$

$$\tau_{K,j}(u_h^{(k)}) = \|\sqrt{\eta} [u_h^{(k)}]\|_{L^2(\partial K) \cap \mathcal{F}_h^I}^2, \quad (15)$$

$$\tau_{K,t}(u_h^{(k)}) = \|\sqrt{h_K} [(\boldsymbol{\Sigma} \nabla_h u_h^{(k)}) \cdot \mathbf{t}_K]\|_{L^2(\partial K) \cup \mathcal{F}_h^I}, \quad (16)$$

where  $h_K$  is the diameter of  $K \in \mathcal{T}_h$ ,  $\Pi$  is the  $L^2$  orthogonal projector onto the space of polynomials of degree  $p_K$  on  $K$ ,  $\eta$  is the penalty coefficient defined in Equation (7), and  $\mathbf{n}_K$  and  $\mathbf{t}_K$  are the unit normal and tangential vectors to  $\partial K$ . Moreover, in Equation (13) the residual  $\mathcal{R}(\cdot, \cdot)$  of the monodomain problem is given by

$$\mathcal{R}(u_h^{(k)}, \mathbf{y}_h^{(k)}) = \chi_m f(u_h^{(k)}, \mathbf{y}_h^{(k)}) - \nabla_h \cdot (\boldsymbol{\Sigma} \nabla_h u_h^{(k)}) + \chi_m C_m \frac{u_h^{(k)} - u_h^{(k-1)}}{\Delta t} - \Pi I_h^{\text{ext}}.$$

Finally, for any  $t^{(k)}$  and any  $K \in \mathcal{T}_K$ , the terms incorporating the oscillations with respect to the data are defined as follows:  $\mathcal{O}_{K,r} = \|h_K (\Pi I_h^{\text{ext}} - I_h^{\text{ext}})\|_{L^2(K)}$ . We remark that in the case of non-homogeneous Neumann boundary conditions, the indicator has to be suitably modified so as to take into account the oscillation term on the boundary datum.

To assess the effectiveness of the *a-posteriori* error indicator, we considered a benchmark test case for electrophysiology based solely on the monodomain equation with a nonlinear reaction term [27] defined by

$$f(u) = a(u - V_{\text{rest}})(u - V_{\text{thres}})(u - V_{\text{depol}})$$

where  $V_{\text{rest}} \leq V_{\text{thres}} \leq V_{\text{depol}}$  and  $a > 0$ . Under this assumption, we can find an analytical solution, as shown later in Section 5.1. In this setting, we evaluate the error exploiting the energy norm that we derived in [11] and defined as follows: for any time  $t$ :

$$\|v\|_E^2 = \|v\|^2 + \int_0^t \frac{2\mu}{C_m \chi_m} \|v\|_{\text{DG}}^2 ds + \int_0^t \frac{a}{C_m} \|v\|_{L^4(\Omega)}^4 ds \quad \forall v \in H^1(\mathcal{T}_h). \quad (17)$$

In Equation (17),  $\|\cdot\|_{\text{DG}}^2$  is given by

$$\|v\|_{\text{DG}}^2 = \|\nabla_h v\|^2 + \|\eta^{\frac{1}{2}} [v]\|_{\mathcal{F}_h^I}^2 \quad \forall v \in H^1(\mathcal{T}_h), \quad (18)$$

where  $\|\cdot\| = \|\cdot\|_{L^2(\Omega)}$  and  $\|\cdot\|_{\mathcal{F}} = \left( \sum_{F \in \mathcal{F}} \|\cdot\|_{L^2(F)}^2 \right)^{\frac{1}{2}}$ . Here,  $\eta$  is defined in Equation (7) and  $H^s(\mathcal{T}_h)$  is the space of piecewise  $H^s$  functions with  $s \geq 0$ . For further details see [11], where we provided the *a-priori* stability, and the convergence analysis with respect to the energy norm defined in (17) for the semi-discrete formulation of the problem in Equation (9).

#### 4.1.1 Update of the polynomial degree

Let  $u_h^{(k)} \in V_h^{DG}$  and  $\mathbf{y}_h^{(k)} \in [V_h^{DG}]^n$  be the numerically approximated solution at time  $t^{(k)}$ ,  $k = 1, 2, \dots$  and let  $\tau_K = \tau_K(u_h^{(k)}, \mathbf{y}_h^{(k)})$  the local error indicator on  $K \in \mathcal{T}_h$  defined as in (12). We can update the polynomial degree on each element  $K \in \mathcal{T}_h$  exploiting the following function:

$$p_K^{\arctan} = \left\lceil p^{\max} \frac{2}{\pi} \arctan \left( \frac{\tau_K}{\tau_{\text{threshold}}} \right) \right\rceil, \quad (19)$$

where  $\lceil \cdot \rceil$  is the ceiling function,  $p^{\max}$  is the maximum polynomial degree to be used in the simulation, and  $\tau_{\text{threshold}}$  corresponds to a suitable threshold to be defined. To avoid large element-wise variations in the local polynomial degree, we opt for a smooth scaling, taking into account not only the information obtained from the evaluation of the adaptive function in Equation (19) but also the polynomial degree  $p_K^{\text{old}}$  at the previous time step. Therefore, for each element, we compute the new element-wise polynomial degree  $p_{\text{new}}$  as:

$$p_K^{\text{new}} = \mathbb{1}_{\{p_K^{\text{old}} \leq p_K^{\text{arctan}}\}} \min(p_K^{\text{old}} + 1, p_K^{\text{arctan}}) + \mathbb{1}_{\{p_K^{\text{old}} > p_K^{\text{arctan}}\}} \max(p_K^{\text{old}} - 1, p_K^{\text{arctan}}). \quad (20)$$

We remark that in the  $p$ -adaptive function defined in Equation (19),  $\tau_{\text{threshold}}$  must be identified to select the subset of elements for which the polynomial degree should be updated (either increased or decreased). The selection of this threshold is typically subject to problem-specific tuning according to the mesh size of the underlying partition  $\mathcal{T}_h$ . To allow an automatic problem-specific selection of  $\tau_{\text{threshold}}$ , we propose a procedure relying once more on the physics of the problem. Since the action potential wavefront is localized in space, the error indicators should be subdivided into two clusters: respectively, the subsets of elements in close proximity to the action potential wavefront should be associated with higher error indicators, while the elements distant from the wavefront should correspond to lower error indicators. For this reason, we employ a  $k$ -means algorithm to cluster the local error indicators into two subsets at each time step (see Figure 1).

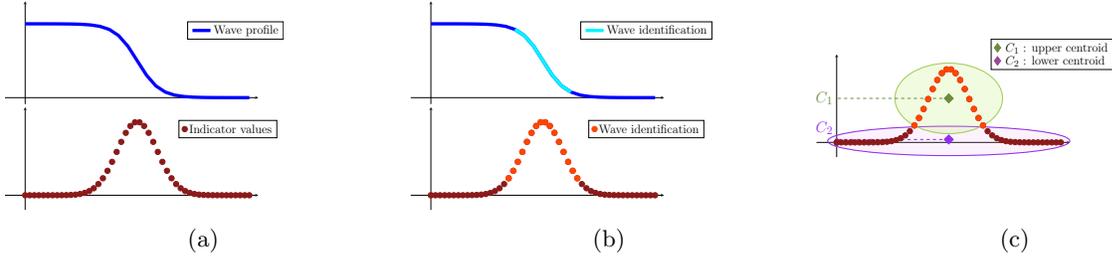


Figure 1: Example of the clustering process for identifying the initial wave in a one-dimensional configuration. (1a) represents an example of a travelling wave and the corresponding distribution of the indicator  $\tau_K$  defined as in Equation (12). (1b) clustering of the local indicator values  $\tau_K$ , based on employing the  $k$ -means algorithm for the identification of the wavefront and the quiescence regions. (1c) definition of the upper centroid ( $C_1$ ) and lower centroid ( $C_2$ ) resulting from the  $k$ -means clustering.

The clustering is performed only at the beginning of the simulation according to the value of the local indicator ( $\tau_K$ ), obtained with an initial condition approximated using the maximum degree. Here  $p_{\max}$  is assigned to all the elements so that it is possible to straightforwardly reduce the polynomial degree far from the wavefront. More in detail, we consider two possible thresholds defined by exploiting the clustering centroids  $C = [C_1, C_2]$  as follows:

$$\tau_{\text{threshold}}^{\min} = \min(C_1, C_2), \quad \tau_{\text{threshold}}^{\text{mean}} = \frac{C_1 + C_2}{2}. \quad (21)$$

Note that, in this way, the threshold for the  $p$ -adaptive strategy is fixed at the beginning of the simulation and it is not updated during time evolution.

## 4.2 $p$ -adaptive algorithm

The polynomial degree is adapted locally and independently for each element using (20). To avoid an excessive computational overhead generated by the calculation of the local error indicator  $\tau_K$  on each element  $K \in \mathcal{T}_h$  at each time step, we design an updating strategy specifically tailored for the problem which involves the evolution of the action potential wavefront. The structure of the adaptive algorithm is summarized in the pseudo-code reported in Algorithm 1 and detailed below. Specifically, for every time step  $t^{(k)}$ , the indicator  $\tau_K(u_h^{(k)}, \mathbf{y}_h^{(k)})$  is not updated on all the mesh elements but only on a specific subset  $\mathcal{S}^{(k)}$  of mesh elements defined as  $\mathcal{S}^{(k)} = \mathcal{S}_K^{(k-1)} \cup \mathcal{S}_{\text{neigh}}^{(k-1)}$ .

In Figure 2, we show how to construct  $\mathcal{S}^{(k)}$ . Specifically, we start from  $\mathcal{S}^{(k-1)}$ , which is defined as the union of the elements  $K \in \mathcal{T}_h$  with a modified polynomial degree at the previous time step and the elements with maximum polynomial degree, i.e.,

$$\mathcal{S}^{(k-1)} = \{K \in \mathcal{T}_h \mid p_K^{(k-1)} \text{ has been updated at the previous time step } t^{(k-1)} \text{ or } p_K^{(k-1)} = p_{\max}\}.$$

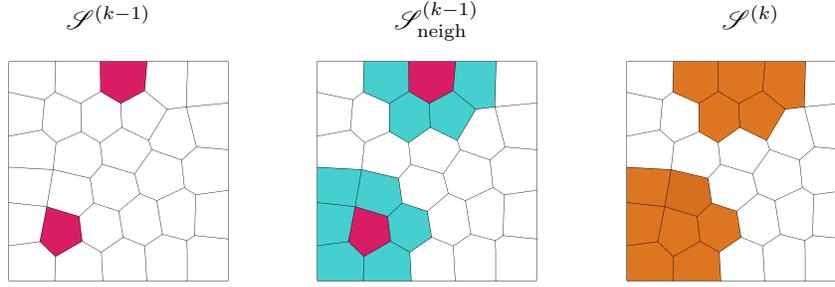


Figure 2: Example of subset of elements  $\mathcal{S}^{(k)}$  where the indicator defined in (12) is updated at the  $k$ -th time step. In red are depicted the elements whose polynomial degree was either updated at the  $(k-1)$ -th time step or was at the maximum allowed value; in blue are depicted the neighbors of such elements. Finally, the complete set of elements for computing the indicator is in orange.

Then,  $\mathcal{S}_{\text{neigh}}^{(k-1)}$  is defined as the subset of neighboring elements to any  $K \in \mathcal{S}^{(k-1)}$ , i.e.,

$$\mathcal{S}_{\text{neigh}}^{(k-1)} = \{K \in \mathcal{T}_h \mid K \text{ is a neighboring element to } K' \in \mathcal{S}^{(k-1)}\}.$$

At the beginning of the simulation, the initial configuration is  $\mathcal{S}^{(0)} = \mathcal{T}_h$ , that is all elements of the mesh are analyzed at the first iteration. Once the subset  $\mathcal{S}^{(k)}$  has been defined, we update the indicator  $\tau_K(u_h^{(k)}, \mathbf{y}_h^{(k)})$  for all the elements in  $\mathcal{S}^{(k)}$ , while for the other elements, we consider the indicator and the local polynomial degree at the previous time step.

---

#### Algorithm 1 $p$ -adaptive algorithm

---

- 1: **Input:** Initial conditions  $\{u_h^{(0)}, \mathbf{y}_h^{(0)}\}$ , Matrices:  $\mathbf{A}, \mathbf{M}$
  - 2: **for**  $k = 1, \dots, N$  **do**
  - 3:   Compute  $u_h^{(k)}, \mathbf{y}_h^{(k)}$  the with the Crank-Nicholson scheme
  - 4:   **if**  $k = 1$  **then**
  - 5:     Exploit the  $k$ -means clustering and the compute threshold  $\tau_{\text{threshold}}$ .
  - 6:   **end if**
  - 7:   Compute  $\mathcal{S}^{(k)}$
  - 8:   Update the indicator  $\tau_K(u_h^{(k)}, \mathbf{y}_h^{(k)})$  for all  $K \in \mathcal{S}^{(k)}$ , for all  $K \notin \mathcal{S}^{(k)}$  consider the indicator at the previous time-step
  - 9:   Update the polynomial degree distribution  $\mathbf{p}^k$  based on employing the adaptive function in Eq. (20)
  - 10:   Update the matrices  $\mathbf{A}, \mathbf{M}$  with respect to the new degree distribution
  - 11:   Update the forcing and ionic terms  $\mathbf{I}_{\text{ext}}^{(k)}, \mathbf{I}_{\text{ion}}^{(k)}$  with respect to the new degree distribution
  - 12:   Update the solution  $u_h^{(k)}, \mathbf{y}_h^{(k)}$  with respect to the new degree distribution
  - 13: **end for**
- 

**Remark 1.** To further reduce the computational costs, we remark that the indicator  $\tau_K(u_h^{(k)}, \mathbf{y}_h^{(k)})$  could also be updated not at every time step but only every  $\bar{k} > 1$  temporal iterations (here  $\bar{k}$  should depend on the velocity of the wavefront and the choice of  $\Delta t$  and  $h$ ).

### 4.3 Implementation details

From a practical implementation perspective, selecting an appropriate set of basis functions is essential for dynamically adapting the polynomial degree without reassembling the mass and stiffness matrices at each time step. In this work, we consider a set of hierarchical basis functions to avoid the reassembling phase at each time step due to the local adaptation algorithm. The initial condition is constructed by employing  $p_K = p^{\max}$  for any  $p_K \in \mathcal{T}_h$ . At the first step, we assemble  $\mathbf{A}, \mathbf{M}$  by choosing  $p_K = p^{\max}$  for any  $p_K \in \mathcal{T}_h$ ; we refer to (10) and Section 3 for their definition. Then, at each time step, we can efficiently and locally update  $\mathbf{A}, \mathbf{M}$  according to the local adaptation of the polynomial degree driven by the  $p$ -adaptive algorithm. The update is trivial and massively parallel for the mass matrix because, in the DG setting, the mass matrix is block diagonal. In Figure 3, we show an example of the

dynamic update of the stiffness matrix  $\mathbf{A}$  for a two-dimensional example. We suppose to have a polytopal mesh with  $n_{\text{el}}$  elements, and we assume, for simplicity and clarity of the example, to have  $p_{\text{max}} = 2$ . If the element  $K_1$  is selected by our adaptive algorithm to decrease the local polynomial approximation degree from  $p_{K_1} = 2$  to  $p_{K_1} = 1$ , the global stiffness matrix has to be locally modified as described in Figure 3. Let  $\mathbf{A}_1 \in \mathbb{R}^{6 \times 6}$  be the block diagonal part of  $\mathbf{A}$  corresponding to the (six) degrees of freedom associated with  $K_1$ . The update of  $\mathbf{A}_1$  into  $\hat{\mathbf{A}}_1 \in \mathbb{R}^{3 \times 3}$  is depicted in Figure 3 (top panel). Let now  $\mathbf{A}_i \in \mathbb{R}^{6 \times 6}$ ,  $i = 2, \dots, n_{\text{el}}$ , the off-diagonal blocks corresponding to the degrees of freedom associated with  $K_1$ . The update of  $\mathbf{A}_j$  into  $\hat{\mathbf{A}}_j \in \mathbb{R}^{3 \times 6}$  is depicted in Figure 3 (bottom panel). We remark that particular attention has to be given to the jump-jump term of the stiffness matrix, i.e.,  $\sum_{F \in \mathcal{F}_h^I} \int_F \eta \llbracket u_h \rrbracket \cdot \llbracket v_h \rrbracket d\sigma \quad \forall u_h, v_h \in V_h^{\text{DG}}$ . Since the penalization parameter  $\eta$  defined in (7) depends on the

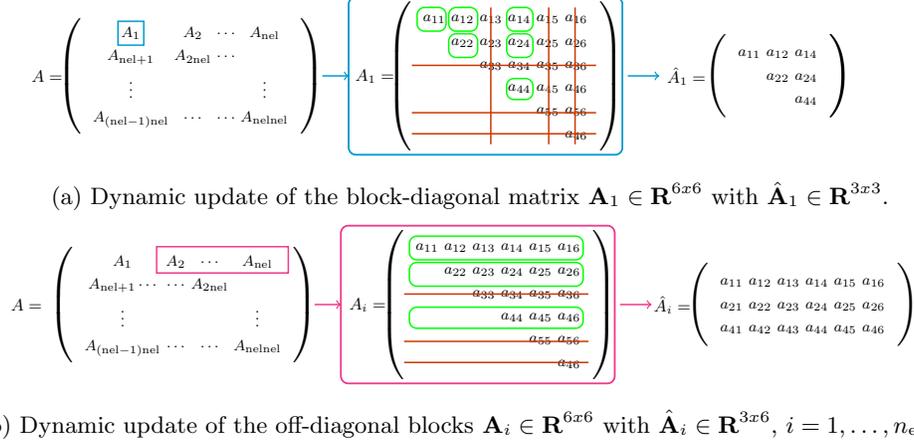


Figure 3: Example of the dynamic update of the stiffness matrix  $\mathbf{A}$  in two dimensions, supposing that the element  $K_1$  is selected to decrease the local polynomial approximation degree from  $p_{K_1} = 2$  to  $p_{K_1} = 1$  whereas all the other mesh elements  $p_K = 2$ .

element-wise polynomial degree, this matrix has to be reassembled at each adaptive step. We can therefore write, exploiting the definition in (8):  $\mathbf{A} = \tilde{\mathbf{A}} + \mathbf{S}$  where  $\mathbf{S}$  is the matrix representation of the jump-jump term (cf. (10) and (7), last term) and  $\tilde{\mathbf{A}}$  is the matrix representation of the first three terms in the bilinear form defined as in (7). In the  $p$ -adaptive algorithm  $\tilde{\mathbf{A}}$  is updated as described above and illustrated in Figure 3 while the matrix  $\mathbf{S}$  is reassembled.

## 5 Numerical results

In this section, we present various numerical tests to evaluate the performance of the proposed  $p$ -adaptive method. The numerical results have been obtained with a numerical code based the open source Lymph library [59], which implements the PolyDG method for multiphysics differential problems.

### 5.1 Test case 1: traveling wave solution

The first test case consists of a traveling wavefront benchmark. For numerical verification, we consider a simplified model that describes an analytical traveling-wave, where the non-linear ionic term is defined by the following non-linear function:  $f(u) = a(u - V_{\text{rest}})(u - V_{\text{thres}})(u - V_{\text{depol}})$  where  $V_{\text{rest}} \leq V_{\text{thres}} \leq V_{\text{depol}}$  and  $a > 0$ . Here,  $V_{\text{rest}}$  is the resting potential,  $V_{\text{thres}}$  represents the threshold potential, and  $V_{\text{depol}}$  represents the depolarization potential. Starting from [27], where the convergence for the one-dimensional problem is analyzed, the solution is extended to the two-dimensional case by introducing  $u_{\text{ex}}$  as the exact solution defined as follows:

$$u_{\text{ex}}(\mathbf{x}, t) = \frac{V_{\text{dep}} - V_{\text{rest}}}{2} \left[ 1 - \tanh \left( \frac{\mathbf{x} - \mathbf{c}t}{\epsilon} \right) \right] + V_{\text{rest}},$$

where  $\epsilon$  characterizes the thickness of the wavefront, while  $\mathbf{c}$  is the propagation speed of the wave.

Table 1: Test case 1. Values of the model parameters.

Parameter	Value	Unit	Parameter	Value	Unit
$\sigma_n$	0.17	$\text{mS} \cdot \text{mm}^{-1}$	$a$	$1.4e-5$	$\text{mS} \cdot \text{mm}^{-2} \cdot \text{mV}^{-2}$
$\sigma_t$	0.62	$\text{mS} \cdot \text{mm}^{-1}$	$\chi_m$	140	$\text{mm}^{-1}$
$V_{\text{depol}}$	30	mV	$C_m$	0.01	$\mu\text{F} \cdot \text{mm}^{-2}$
$V_{\text{rest}}$	-85	mV	$c$	0.5	$\text{mm} \cdot \text{ms}^{-1}$
$V_{\text{thres}}$	-57.6	mV	$\epsilon$	0.2	mm

### 5.1.1 Test case 1a: convergence analysis

For the first test case, we consider the domain  $\Omega = (-1, 2) \times (-0.5, 0.5)$ , discretized with a mesh constructed by PolyMesher [60]. We define the conductivity tensor as  $\Sigma = 0.1336\mathbb{1}[\text{mS} \cdot \text{mm}^{-1}]$ ,  $\mathbb{1}$  being the identity tensor. The model's parameters are chosen as reported in Table 1, cf. from [27]. In Figure 4, we report the computed errors in the energy norm defined in (17) at the final time  $T = 1e-4$  [s] with  $\Delta t = 1e-6$ . These results have been obtained considering a uniform polynomial degree for every element of the mesh, i.e.,  $p_K = p_{\text{un}}$  for any  $K \in \mathcal{T}_h$ , with  $p_{\text{un}} = 1, 2, 3, 4$  and considering a sequence of uniformly refined meshes with mesh-size  $h$  (the corresponding number of elements is equals to 70, 250, 800, 3000, 8000). We observe that the theoretical convergence rates are achieved, consistently with the theoretical analysis proved in [11].

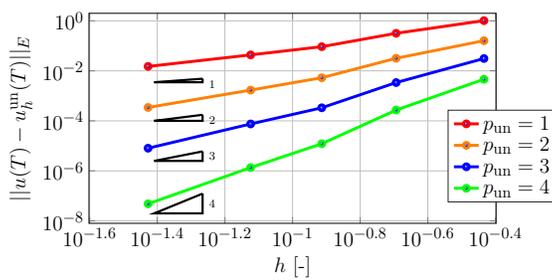


Figure 4: Test case 1a. Computed errors in the energy norm defined in Equation (17) at the final time  $T = 1e-4$  [s] as a function of the mesh size (loglog scale) for different choices of the polynomial degree  $p_{\text{un}} = 1, 2, 3, 4$ .

We next present the numerical results obtained based on employing our  $p$ -adaptive algorithm and compare them with the corresponding ones obtained under uniform  $p$ -refinement. Figure 5 reports the computed errors in the energy norm defined in (17) at the final time  $T = 1e-4$  as a function of total number of degrees of freedom (NDoF) for adaptive and uniform polynomial distributions (loglog scale). More precisely, in Figure 5 we show the computed errors obtained with a uniform polynomial degree  $p_{\text{un}} = 2, 3, 4$  and compare them with the corresponding results obtained by employing the proposed  $p$ -adaptive method (denoted as  $p_{\text{ad}}$ ).

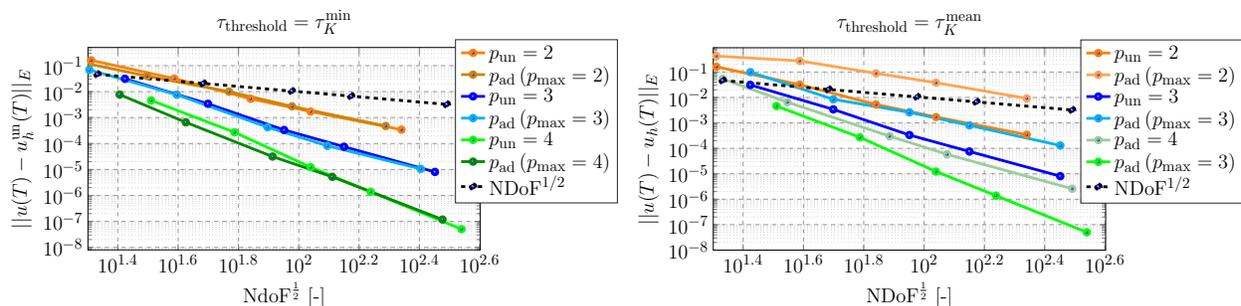


Figure 5: Test case 1a. Computed errors in the energy norm defined in (17) at the final time  $T = 1e-4$  [s] as a function of total number of degrees of freedom NDoF (loglog scale) for uniform ( $p_{\text{un}}$ ) and adaptive ( $p_{\text{ad}}$ ) polynomial degree and different choices of  $\tau_{\text{threshold}} = \tau_{\text{threshold}}^{\min}, \tau_{\text{threshold}}^{\text{mean}}$ , cf. (21). Here  $p_{\text{max}}$  represents the maximum polynomial degree in the adaptive algorithm.

These results are based on two different threshold parameters: Figure 5 (left) show the results obtained setting  $\tau_{\text{threshold}} = \tau_{\text{threshold}}^{\min}$ , i.e., the threshold is defined as the minimum of the centroids of the  $k$ -means clustering (cf.

also (21)), while Figure 5 (right) show the results obtained setting  $\tau_{\text{threshold}} = \tau_{\text{threshold}}^{\text{mean}}$ , i.e., the mean of the centroids obtained from the  $k$ -means clustering (cf. also (21)). In this set of numerical experiments, we impose the maximum polynomial degree in our  $p$ -adaptive algorithm as  $p_{\text{max}} = 2, 3, 4$ . From the results reported in Figure 5, we can conclude that choosing as threshold the mean of the centroids seems to lead to larger errors, while exploiting a threshold based on the minimum of the centroids seems to provide better results and lead to approximations that are comparable to those obtained with a uniform polynomial degree. Given these numerical results, the simulation of the following sections will be carried out using the minimum threshold.

### 5.1.2 Test case 1b: degrees of freedom analysis

We next analyze the performance of our  $p$ -adaptive scheme to correctly track the evolution of the wavefront. We set  $\Omega = (-2, 3) \times (-0.5, 0.5)$ , and consider a polytopal mesh with 1500 elements, cf. Figure 6b. Here, the conductivity tensor is defined as  $\Sigma = 0.00811 \text{ [mS} \cdot \text{mm}^{-1}]$ , and all the other parameters are defined as in Table 1. For the  $p$ -adaptive scheme, the initial polynomial degree is uniform and equal to  $p_{\text{un}} = 5$ , which corresponds to a total number of degrees of freedom equal to  $\text{NDoF} = 3.15e + 4$ . The initial condition for the transmembrane potential is reported in Figure 6b.



Figure 6: Test case 1b. (6a) Polytopal grid with 1500 elements ( $h = 0.1170$ ); (6b) initial condition of the transmembrane potential.

In Figure 7, we report the results obtained at two time snapshots  $t = 5 \text{ [ms]}$  and  $t = 20.3 \text{ [ms]}$ . The first two rows of Figure 7 report the exact solution together with the corresponding numerical solution. In the third and fourth rows of Figure 7, we report the element-wise computed errors in the  $DG$  and  $L^2$  norms. The element-wise

Table 2: Test case 1b. Computed errors in the  $L2$  and  $DG$  norms for different time snapshots  $t = 5$  and  $t = 20.3 \text{ [ms]}$ . Here  $u_h^{\text{ad}}$  is the numerical solution computed with the  $p$ -adaptive scheme,  $u_h^{\text{un}}$  is the one computed with a uniform polynomial degree  $p_{\text{un}} = 5$ ).

	$\ u_h^{\text{un}}(t) - u_{\text{ex}}(t)\ _{L^2(\Omega)}$	$\ u_h^{\text{un}}(t) - u_{\text{ex}}(t)\ _{DG}$	$\ u_h^{\text{ad}}(t) - u_{\text{ex}}(t)\ _{L^2(\Omega)}$	$\ u_h^{\text{ad}}(t) - u_{\text{ex}}(t)\ _{DG}$
$t = 5 \text{ [ms]}$	$1.63e - 3$	$1.15e - 2$	$1.62e - 3$	$1.14e - 2$
$t = 20.3 \text{ [ms]}$	$5.43e - 3$	$2.54e - 2$	$5.50e - 3$	$2.55e - 2$

polynomial degree adapted with our  $p$ -adaptive algorithm ( $\tau_{\text{threshold}} = 3.25e - 3$ ) every 5 time steps of the Crank-Nicholson time advancing scheme is shown in Figure 7 (fifth row) together with corresponding local error indicator defined as in Equation (12) (Figure 7, sixth row). The results show that the proposed  $p$ -adaptive algorithm is able to track the wavefront correctly. The distribution of the local polynomial degree for each element of the mesh over time is consistent with the position of the wavefront. In particular, we observe that high-order polynomial degrees are automatically set by our  $p$ -adaptive algorithm to correctly capture the steep wavefront, while low order elements are employed where the solution is flat.

The evolution of the total NDoF as a function of time resulting from our  $p$ -adaptive algorithm is shown in Figure 8a. Starting at the initial time from a uniform polynomial degree distribution  $p_{\text{max}} = 5$ , the  $p$ -adaptive algorithm is able to significantly reduce the total number of degrees of freedom while preserving the accuracy, resulting in a gain of 78% compared to the uniform approximation. We also observe that, once the wave has passed, all elements are discretized with linear elements, leading to 4500 degrees of freedom, resulting in a gain of 86%. Figure 8b shows the evolution of the total number of modified elements (Nel) over time, while Figure 8c shows the evolution over time of the number of elements with local polynomial degree equal to 1, 2, 3, 4, 5. As expected, most of the elements employ linear elements, except for a few elements in the proximity of the wavefront, where high-order polynomials are employed to correctly describe steep variations of the solution. Exploiting the proposed  $p$ -adaptive algorithm, we observe a consistent reduction of the total number of degrees of freedom needed to discretize the system while maintaining accuracy in the approximation of the solution, as is evident from the

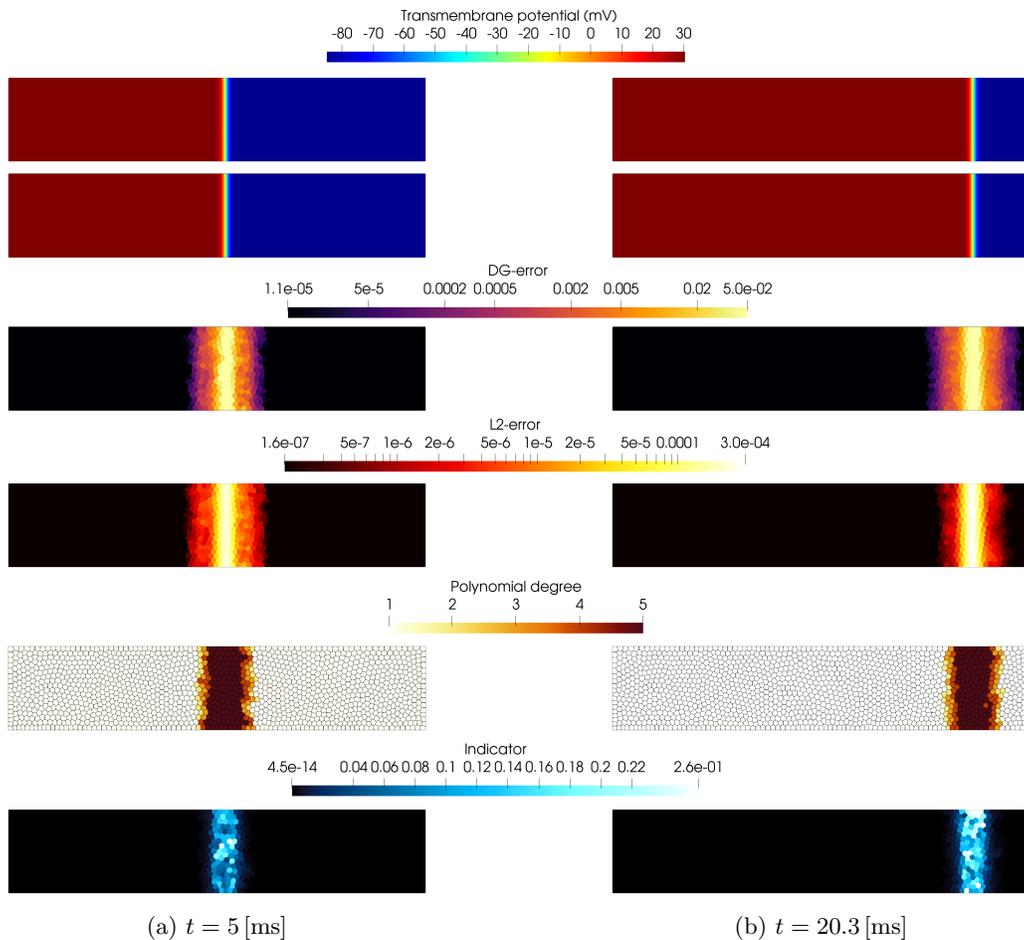


Figure 7: Test case 1b. Exact solution  $u_{ex}$  (first row), numerical solution  $u_h$  (second row), computed errors in the  $DG$  and  $L^2$  norms (third and fourth rows), element-wise polynomial approximation degree (fifth row) and local error indicator (sixth row) at two time instants  $t = 5$  [ms] and  $t = 20.3$  [ms].

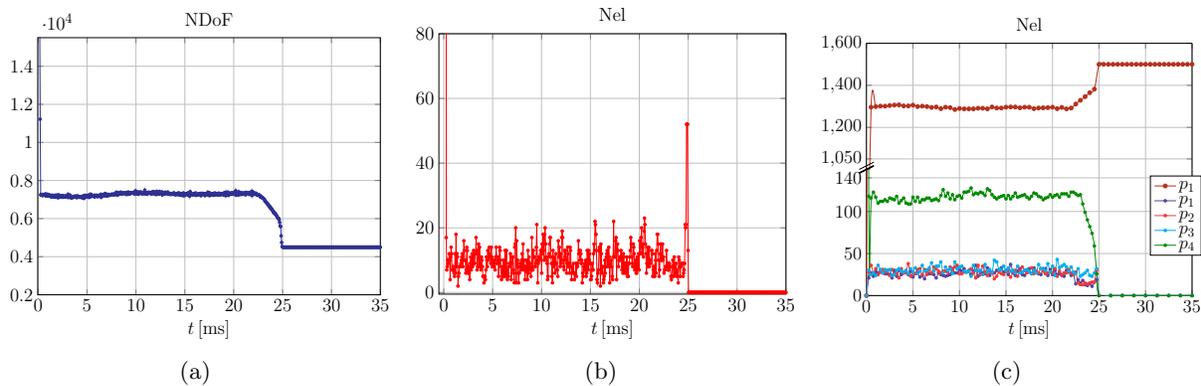


Figure 8: Test case 1b. Left: Evolution of the number of degrees of freedom (NDoF) as a function of time, driven by our  $p$ -adaptive algorithm ( $\tau_{\text{threshold}} = 3.25e - 3$ ). At the initial time  $\text{NDoF} = 3.15e + 4$ . Center: Evolution of number of elements (Nel) where the local polynomial degree is updated over time. Right: Evolution of the number of elements (Nel) with local polynomial degrees equal to 1, 2, 3, 4, 5 over time. As initial condition, all the elements are discretized with  $p_{\text{max}} = 5$ ;

results shown Table 2 where the computed errors in the  $L2$  and  $DG$  norms obtained with the  $p$ -adaptive scheme and with a uniform polynomial degree  $p_{\text{un}} = 5$  are compared at two different time snapshots  $t = 5, 20.3$  [ms].

### 5.1.3 Test case 1c: comparison of different choices for the local error indicators

Next, we investigate different choices of local error indicators in driving our  $p$ -adaptive strategy. Specifically, we compare:

1. the "full" local error indicator  $\tau_K$  as defined in Equation (12),
2. the local error indicator relating only to the residual of the numerical solution, i.e.,  $\tau_{K,r}$ ,
3. the local error indicator related to the jump of the numerical solution, i.e.,  $\tau_{K,j}$ .

Figure 9 shows the time evolution (with final time  $T = 10$  [s]) of the total number of degrees of freedom (left) and of the total number of elements marked for  $p$ -refinement/de-refinement when our adaptive algorithm is driven by the three different local error indicators (right). Our results indicate that the number of degrees of freedom resulting from using the residual indicator seems to be consistently smaller than the ones obtained by exploiting the other two indicators.

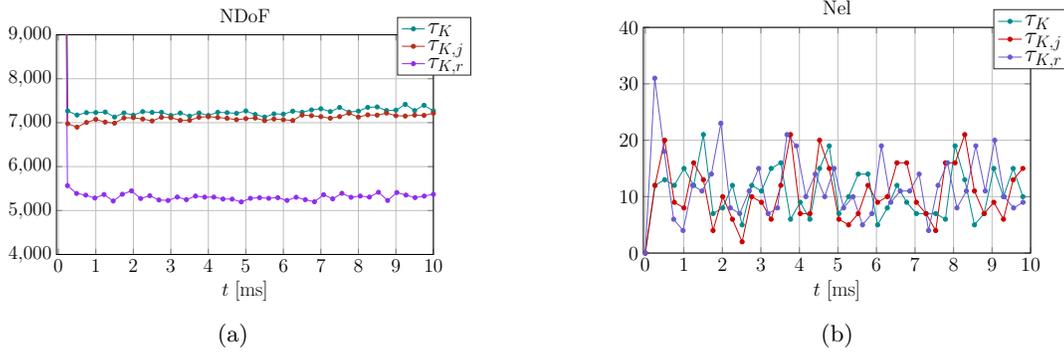


Figure 9: Test case 1c. Time evolution of the total number of degrees of freedom (left) and of the total number of elements marked for  $p$ -refinement/de-refinement (right) for the three different choices of the local error indicators driving the  $p$ -adaptive algorithm.

Figure 10 shows a comparison between the computed solution at time  $t = 2$  [ms] (first row) as well as the element-wise polynomial approximation degree for the three choices of local error indicators (second row). In the same Figure (third and fourth rows) we report the computed errors in the  $L2$  and  $DG$  norms together with the element-wise values of the three error indicators (last row). From these results, we can deduce that the local polynomial degree distribution associated with the local residual indicator  $\tau_{K,r}$  entails a lower number of degrees of freedom than the other two indicators. However, this choice leads to a less accurate numerical solution, as clear from the results reported in the third and fourth rows, where the errors computed based on employing  $\tau_{K,r}$  are consistently larger. This suggests that the residual indicator is underestimating the error and, as a consequence, lowering the local polynomial degree even in at the wavefront, compromising the numerical scheme accuracy. The "full" and jump local error indicators outperform the residual indicator: indeed, we observe that they can correctly identify the wave, maintaining the required accuracy, as shown in from Figures 11. We also observe that for the adaptive method based on the complete indicator  $\tau_K$ , the traveling wave is completely included in the band of elements where high-order approximations are employed, and the elements are then gradually scaled down to linear elements in correspondence with quiescence regions. Furthermore, comparing the results obtained making use of the complete indicator ( $\tau_K$ ) and the jump indicator ( $\tau_{K,j}$ ), the time evolution of the total number of degrees of freedom is comparable and the error obtained by the adaptive algorithm considering  $\tau_{K,j}$  as local error indicator seems to be slightly larger than the corresponding results obtained based on employing  $\tau_K$ .

Figure 11 illustrates the correlation between the error computed in the energy norm defined in Equation (17) and three distinct a posteriori indicators. More precisely, In Figure 11 we report the element-wise scatterplot of the local error computed in energy norm ( $y$ -axis) when the  $p$ -adaptive algorithm is driven by the different local errors indicators versus the value of the three local error indicators ( $x$ -axis) for different time snapshots  $t = 1, 5, 10$  [ms]. From the results reported in Figure 11, the complete indicator  $\tau_K$  proves to be the most effective indicator, as it exhibits a clear linear relation with the error computed in the energy norm, with values consistently bounded from below by the corresponding error. This behavior suggests that  $\tau_K$  provides a robust and reliable local error estimator and can be exploited within the  $p$ -adaptive algorithm. In fact, exploiting the complete indicator we obtain a  $L2$  error of  $3.82e - 3$  and  $DG$ -error of  $4.12e - 2$ . Conversely, as expected, the residual indicator  $\tau_{K,r}$  confirms its

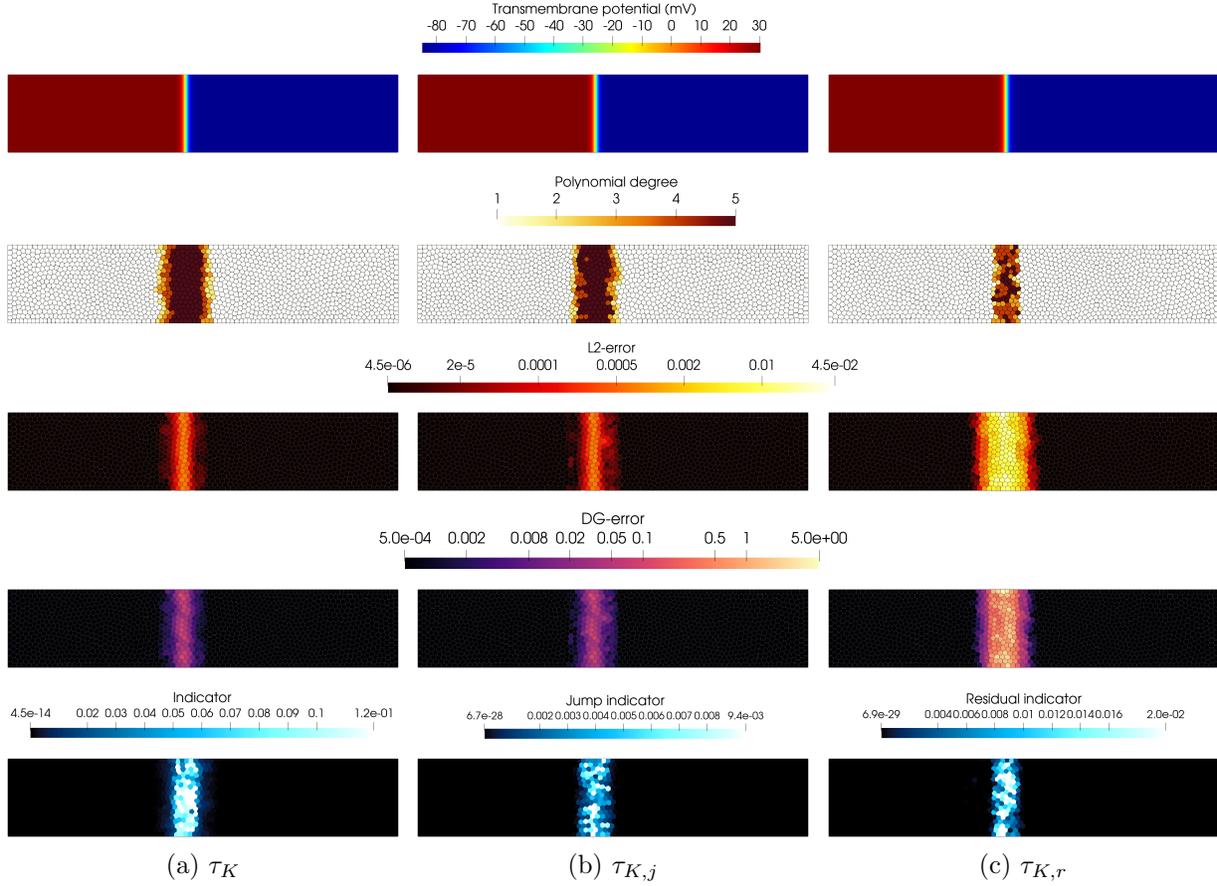


Figure 10: Test case 1c. From left to right: different local error indicators are employed to drive the  $p$ -adaptive algorithm. First row: numerical solution at  $t = 2$  [ms]; Second row: element-wise polynomial degree distribution; third and fourth rows: element-wise errors in the  $L2$  and  $DG$  norms; Fifth row: computed local error indicators.

poor effectiveness: it is characterized by significant variance, lacks a clear correlation with the error, and frequently underestimates it, obtaining a  $L2$  error of  $1.05e - 1$  and  $DG$ -error of  $1.32e + 1$ . Finally, the jump indicator  $\tau_{K,j}$

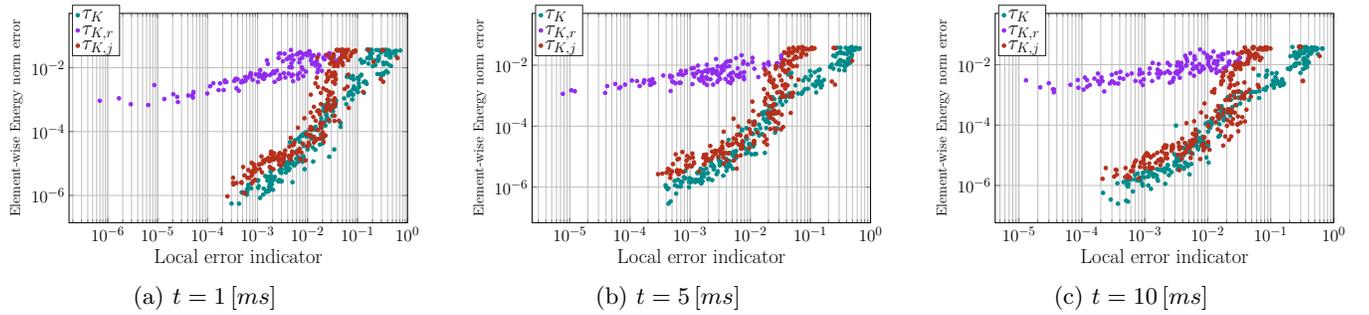


Figure 11: Test case 1c. Scatterplot of the element-wise error computed in energy norm defined in (17) (y-axis) versus the local error indicators (x-axis) for three different time snapshots  $t = 1, 5, 10$  [ms].

exhibits an intermediate behavior, with a discernible positive correlation with the error and moderate variance; the values are generally of the same order of magnitude as  $\tau_K$  but the indicator does not achieve the same level of accuracy and tends to underestimate the error, especially for lower error magnitudes. Here we obtain a  $L2$  error of  $3.85e - 3$  and  $DG$ -error of  $4.74e - 2$ .

## 5.2 Test case 2: "double" traveling wave solution

In the second test case, we numerically investigate the ability of the  $p$ -adaptive algorithm to approximate multiple wavefronts. We consider a domain  $\Omega = (-2, 4) \times (-0.5, 0.5)$  and two starting wavefronts defined through the following initial condition:

$$u^0(\mathbf{x}) = \frac{V_{\text{dep}} - V_{\text{rest}}}{2} \left( \tanh\left(\frac{\mathbf{x} + x_2}{\epsilon_2}\right) - \tanh\left(\frac{\mathbf{x} + x_1}{\epsilon_1}\right) \right) + V_{\text{dep}}. \quad (22)$$

We consider two configurations: a homogeneous case (see Section 5.2.1), in which the wavefronts have the same velocity and steepness, and an heterogeneous case, in which the domain is split by assigning different conductivity values that cause wavefronts to propagate at different velocities (see Section 5.2.2).

### 5.2.1 Test case 2a: homogeneous case

This test case consists of a simulation in which both waves move with the same velocity  $c = 0.1212 [\text{mm} \cdot \text{ms}^{-1}]$ . The mesh is characterized by 1500 polytopal elements ( $h = 0.1105$ ); the initial condition is defined as in Equation (22), with  $x_1 = -3$  and  $x_2 = 1.5$ . We point out that the two waves evolve in opposite directions towards the center of the domain at the same speed. We have repeated the same analysis of test case of Section 5.1.2 considering the complete indicator  $\tau_K$  and  $\tau_{\text{threshold}} = \tau_{\text{threshold}}^{\text{min}}$ . Figure 12 reports the time evolution of the number of degrees of freedom NDoF as a function of time. Starting from  $3.15e + 4$  NDoF (corresponding to the case where all the elements are discretized with  $p_{\text{max}} = 5$ ) we reach around  $8.6e + 3$  NDoF (with a 73% reduction). The corresponding time evolution of the number of elements (Nel) where the local polynomial degree is updated over time is shown in Figure 12 (center) together with the time evolution of the number of elements with local polynomial degrees equal to 1, 2, 3, 4, 5. We also observe that after the two waves collide (approximately at 15.6 [ms]), a resting state is reached with a discretization of linear polynomials and  $4.5e + 3$  NDoF (corresponding to a 86% reduction). In addition, the cost of computing the local error indicator is decreased significantly due to the localized computations of the indicator for the relevant elements only, as shown in Figure 12b. Also in this test case, we observe a significant reduction in the total number of degrees of freedom as a consequence of the  $p$ -adaptive scheme. In Figure 13 (first

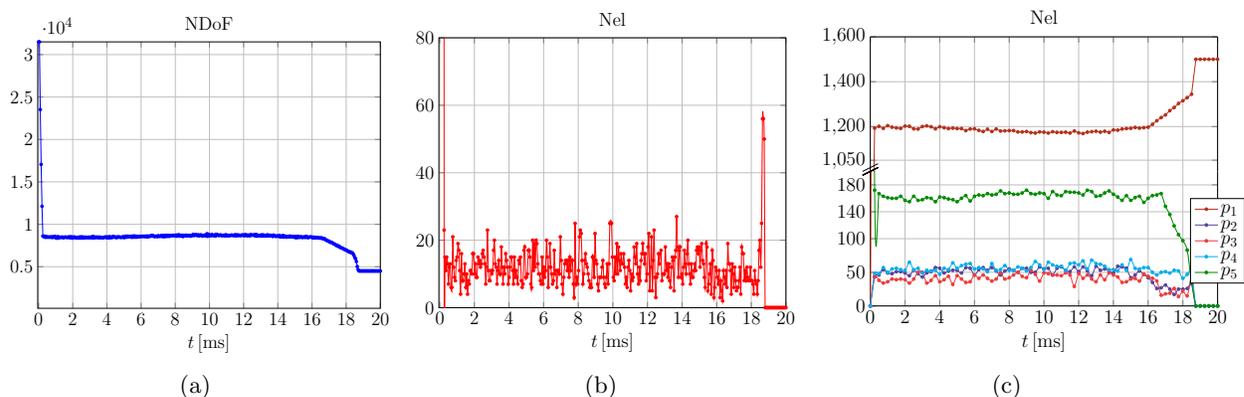


Figure 12: Test case 2a. Left: Evolution of the number of degrees of freedom (NDoF) as a function of time, driven by our  $p$ -adaptive algorithm ( $\tau_{\text{threshold}} = 3.25e - 3$ ). At the initial time NDoF =  $3.15e + 4$ . Center: Evolution of the number of elements (Nel) where the local polynomial degree is updated over time. Right: Time evolution of the number of elements (Nel) with local polynomial degrees equal to 1, 2, 3, 4, 5. At the initial time all the elements are discretized with  $p_{\text{max}} = 5$ .

Table 3: Test case 2a. Computed errors in the  $L2$  and  $DG$  norms for different time snapshots  $t = 7.1$  [s] and  $t = 16.6$  [s]. As exact solution we considered the numerical solution computed with a uniform polynomial degree ( $p_{\text{un}} = 5$ ).

	$t = 7.1$ [s]	$t = 16.6$ [s]
$\ u_h^{\text{ad}}(t) - u_h^{\text{un}}(t)\ _{L^2(\Omega)}$	$7.37e - 5$	$9.12e - 5$
$\ u_h^{\text{ad}}(t) - u_h^{\text{un}}(t)\ _{\text{DG}}$	$1.80e - 3$	$2.04e - 3$

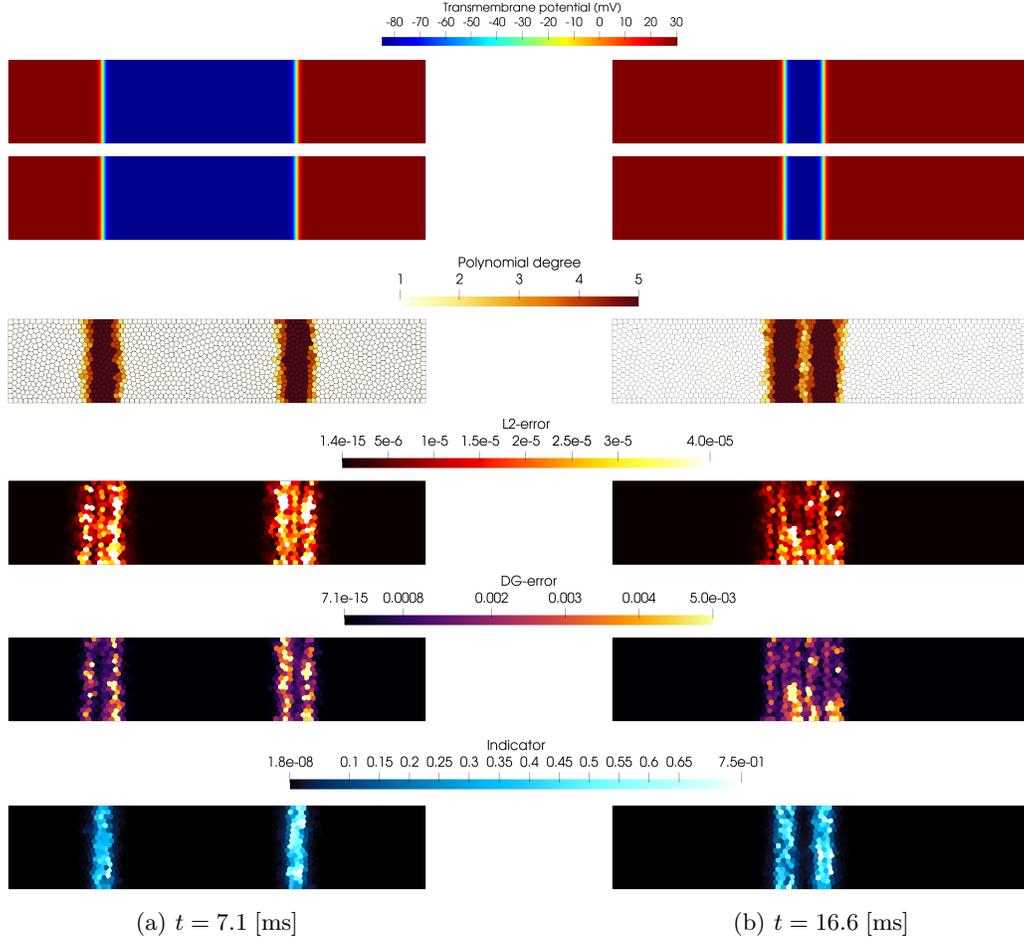


Figure 13: Test case 2a. First and second rows: numerical solution computed based on employing a uniform polynomial degree ( $p_{\text{un}} = 5$ ) and the  $p$ -adaptive strategy (with  $p_{\text{max}} = 5$ ), respectively. Third row: polynomial approximation degree distribution. Third and fourth rows: errors in the  $L2$  and  $DG$  norms computed with the numerical solution obtained with a uniform polynomial degree ( $p_{\text{un}} = 5$ ). Fifth row: local error indicator. The results refer to two different time snapshots  $t = 7.1$  [ms] (left panel) and  $t = 16.6$  [ms] (right panel).

and second rows) we report the numerical solution computed based on employing a uniform polynomial degree ( $p_{\text{un}} = 5$ ) and the  $p$ -adaptive strategy (with  $p_{\text{max}} = 5$ ), respectively, at two different time snapshots  $t = 7.1$  [ms] (left panel) and  $t = 16.6$  [ms] (right panel). We observe that: i) the  $p$ -adaptive algorithm correctly tracks the two wavefronts, with a high-order polynomial degree employed in the proximity of the wavefronts; ii) the polynomial degree distribution is similar for both waves, cf. Figure 13 (third row). This is consistent with what is expected because the two waves have the same slope and speed. Figure 13 (third and fourth rows) also reports the errors in the  $L2$  and  $DG$  norms computed with the numerical solution obtained with a uniform polynomial degree ( $p_{\text{un}} = 5$ ). Comparing these results with the local error indicator reported in Figure 13, last row, we can observe that they exhibit the same behavior, indicating an accuracy of the proposed  $p$ -adaptive algorithm. This is also confirmed by the results reported in Table 3, where we show the errors in the  $L2$  and  $DG$  norms computed for the test case presented, considering as reference solution the numerical solution computed using a uniform polynomial degree ( $p_{\text{un}} = 5$ ).

### 5.2.2 Test case 2b: heterogeneous case

In this test case, we consider a heterogeneous conductivity tensor resulting in the two waves traveling at different speeds and with different slopes. This test was designed to assess the ability of the  $p$ -adaptive algorithm to capture two distinct waves with different dynamics. For this test case, we consider a discontinuous conductivity tensor  $\Sigma(\mathbf{x}) = \Sigma_1 \mathbb{1}_{\Omega_1}(\mathbf{x}) + \Sigma_2 \mathbb{1}_{\Omega_2}(\mathbf{x})$ , where the subdomains  $\Omega_1$  and  $\Omega_2$  are defined as:  $\Omega_1 = (-2, 1) \times (-0.5, 0.5)$  and  $\Omega_2 = (1, 4) \times (-0.5, 0.5)$  (cf. Figure 14a). Imposing the wave speed as  $c_1 = 0.1212$  [mm·ms<sup>-1</sup>] and  $c_2 = 0.3157$  [mm·ms<sup>-1</sup>],

we obtain the conductivity tensors  $\Sigma_1 = 0.00811 [\text{mS} \cdot \text{mm}^{-1}]$  and  $\Sigma_2 = 0.05511 [\text{mS} \cdot \text{mm}^{-1}]$ . This choice results in a faster wave evolving from the right to the left of the domain until it encounters a jump in conductivity at  $x = 1$ ; after that, the wave slows down and becomes steep exactly like the other one, evolving with the same speed. The initial condition profile together with the computational grid is shown in Figure 14b.



Figure 14: Test case 2b. Computational mesh and initial condition for double traveling wavefront (left) and profile of the initial condition (right).

Also in this case we have repeated the same analysis of test case Section 5.1.2. Figure 15 shows the time evolution of NDoF together with the evolution of the number of elements where the local polynomial degree is updated over time and the time evolution of the number of elements with local polynomial degrees equal to 1, 2, 3, 4, 5. With respect to the homogeneous test case analyzed in Section 5.2.1, we observe an increase in the total number of degrees of freedom. The different wave speed induces the  $p$ -adaptive algorithm to increase the bandwidths where high-order polynomials are employed. Moreover, the presence of the discontinuity in the conductivity tensor leads to an additional increase in the total NDoF in the time period when the second wavefront passes through that specific portion of the domain (a zoom in this particular time window is reported in 15a). This increase in NDoFs lasts from 2 [ms] up to 5 [ms]. After that, the number of elements with each polynomial degree stabilizes, since the two waves have the same profile (see Figure 15c). From the numerical results, we can conclude that the proposed  $p$ -adaptive algorithm accurately identifies the two waves in the domain, using different polynomial degree distributions to effectively capture the fronts in a heterogeneous context. Figure 16 shows, for different time

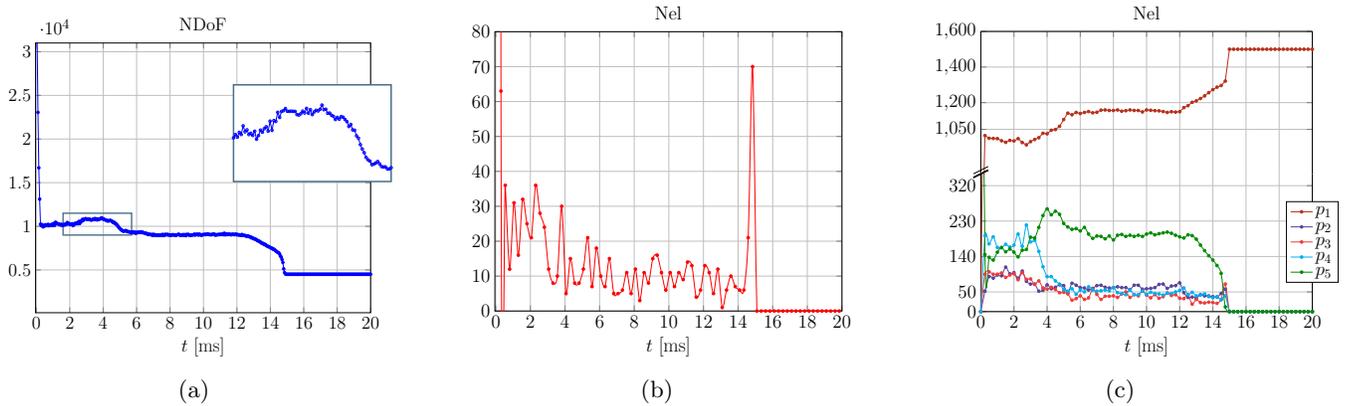


Figure 15: Test case 2b. Left: Evolution of the number of degrees of freedom (NDoF) as a function of time, driven by our  $p$ -adaptive algorithm. Center: Evolution of number of elements (Nel) where the local polynomial degree is updated over time. Right: Evolution of the number of elements (Nel) with local polynomial degrees  $p_K = 1, 2, 3, 4, 5$  over time. At the initial time all the elements are discretized with  $p_{\max} = 5$ .

snapshots  $t = 1.5, 3.6, 7.7, 13$  [ms], the computed solution, the element-wise polynomial approximation degree, and the local error indicator. We observe, as expected, the distinct polynomial bandwidths visible for the two wavefronts. At the beginning of the simulation, the less steep, but faster, wave on the right-hand side of the domain shows a bandwidth where high-order polynomials are employed. Close to the jump in the conductivity, the wave speed and slope change, evolving similarly to the wave on the left, which is steeper.

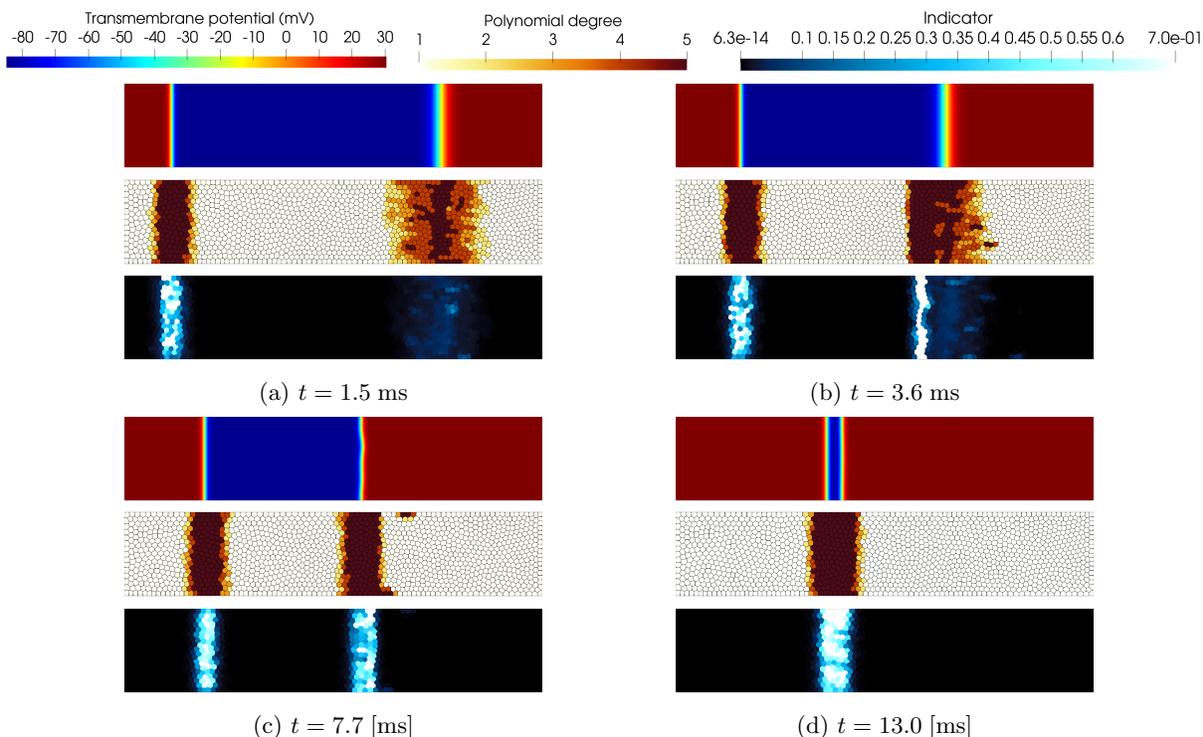


Figure 16: Test case 2b. Computed solution, element-wise polynomial approximation degree and local error indicator defined as in Equation (12) for different time snapshots  $t = 1.5, 3.6, 7.7, 13.0$  [ms].

## 6 Application to neuronal electrophysiology

In this section, we present a test case demonstrating the capabilities of the proposed  $p$ -adaptive method in approximating physio-pathological scenarios of brain electrophysiology.

### 6.1 Grey matter tissue

In this test case, we challenge the proposed  $p$ -adaptive algorithm in the context of neuronal electrophysiology, where we exploit the coupling of the monodomain equation with the Barreto-Cressman conductance-based ionic model (1), which is capable of representing the dynamics of neuronal electrophysiology in brain tissues. We focus on the evolution of pathological neuronal impulses, representing dynamics generated by a potassium ion imbalance in the transmembrane space. This spiking behavior, typical of epileptic seizure, is investigated in portions of grey matter tissue modeled isotropically. The initial condition for the potential is defined by taking  $u^0 = -50$  [mV]

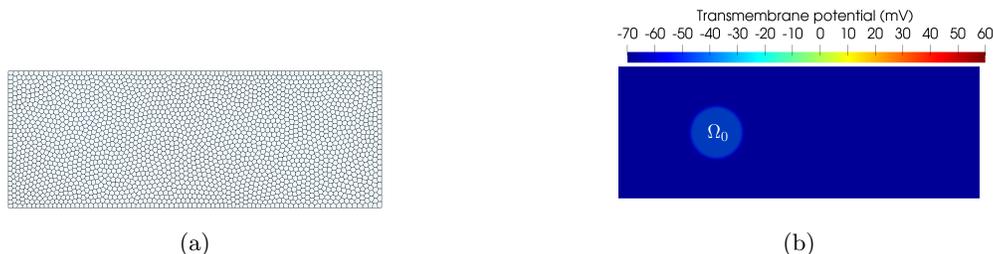


Figure 17: Test case 3. Computational domain and polytopal grid with 2500 elements ( $h = 0.1148$ ) (left) and initial condition for the transmembrane potential (right).

on a subregion  $\Omega_0$ , while the remaining part of the domain has  $u^0 = -67$  [mV], as done in [11]. In this case, a pathological condition is simulated on a slab of grey matter tissue on a mesh of 2500 elements ( $h = 0.1148$ ), cf. Figure 17, and throughout the section we set  $p_{\max} = 5$ . In Table 4 we report the initial conditions of all the variables of the ionic system, and in Table 5 we list the values of the parameters of the Barreto-Cressman

Table 4: Test case 3. Parameters for the initial condition of the coupled monodomain Barreto-Cressman ionic model. Values taken from [18].

Variable	Initial Value	Units	Variable	Initial Value	Units
$n^0$	15.5	mM	$c^0$	0	mM
$k^0$	7.8	mM	$g^{n,0}$	0.0936	-
$g^{c,0}$	0.08553	-	$g^{k,0}$	0.96859	-

Table 5: Test case 3. Parameters for Barreto-Cressman ionic model. Values taken from [18].

Parameter	Description	Values
$C_m$	Membrane capacitance	$10^{-2}$ mF/cm <sup>2</sup>
$G_{\text{AHP}}$	Conductance of afterhyperpolarization current	0.01 mS/cm <sup>2</sup>
$G_{\text{KL}}$	Conductance of potassium leak current	0.05 mS/cm <sup>2</sup>
$G_{\text{Na}}$	Conductance of persistent sodium current	100 mS/cm <sup>2</sup>
$G_{\text{CIL}}$	Conductance of chloride leak current	0.05 mS/cm <sup>2</sup>
$G_{\text{NaL}}$	Conductance of sodium leak current	0.0175 mS/cm <sup>2</sup>
$G_{\text{Ca}}$	Calcium conductance	0.1 mS/cm <sup>2</sup>
$G_{\text{K}}$	Conductance of potassium current	40.0 mS/cm <sup>2</sup>
$G_{\text{glia}}$	Strength of glial uptake	66.66 mM/s
$K_{\text{bath}}$	Conductance of potassium	8.0 mM

model used for this simulation, cf. [18]. The conductivity values are taken as in [7] for a portion of grey matter tissue, namely,  $\Sigma = 0.773411$  [mS · mm<sup>-1</sup>]. As extensively analyzed in [11, 61], we exploit a pathological value of the parameter  $K_{\text{bath}} = 8$  [mM] that allows the model to enter the Hopf-type bifurcation [62]. This configuration results in a very fast and auto-induced bursting behavior. In this work, the pathological behavior is further accentuated by introducing an external forcing term, which increase the frequency of the spikes, accelerating the bursting behavior. The forcing term is defined as follows:

$$I^{\text{ext}}(\mathbf{x}, t) = \frac{A}{1 + e^{\sin(t)}} \mathbb{1}_{\Omega_0}(\mathbf{x}). \quad (23)$$

Figure 18 depicts the acceleration of the pathological dynamics generated by this forcing: in the self-induced case, the action potential generates a second peak after 40 [ms] (Figure 18a); in contrast, the introduction of the external force anticipates the second peak at approximately 7 [ms] (Figure 18b). As before, we consider the error indicator for the adaptive choice of the polynomial degree defined in Equation (12). Figure 19 reports the computed solution,

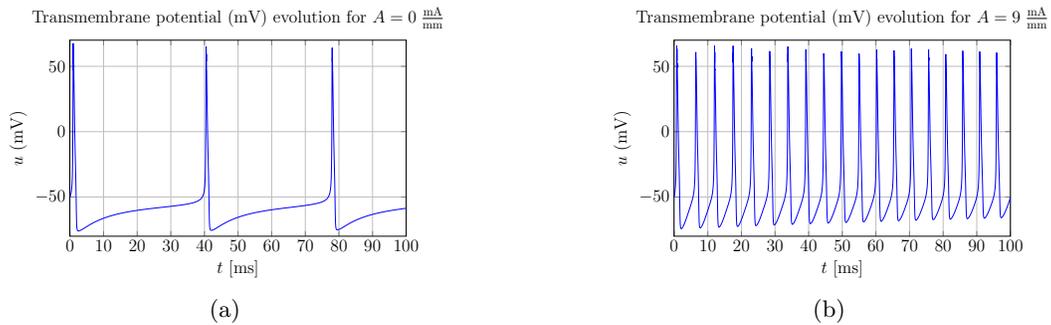


Figure 18: Test case 3. Figure (18a) shows the dynamics of the auto-induced and bursting pathological behavior of the 0D-ionic model. In Figure (18b) we shows the evolution of the transmembrane potential in the 0D-ionic model in presence of the forcing term defined in Equation (23) with amplitude  $A = 9 \frac{\text{mA}}{\text{mm}}$ .

the element-wise polynomial approximation degree, and the computed total local error indicator defined as in (12) at different time snapshots  $t = 3.4, 7.2, 8.4, 13.2$  [ms]. As expected, larger values of the local error indicator and, as a consequence, higher-order polynomials are observed in correspondence with the depolarization and polarization wavefronts. As it can also be seen from the evolution of the 0D-model in Figure 18a, an equally rapid decay (polarization) follows the potential peak, which also requires high-order polynomial degrees for the accurate local approximation. Regions of quiescence correspond instead to linear polynomial approximation degrees, where the



the numerical results, it can be concluded that our  $p$ -adaptive algorithm demonstrates the ability to automatically adjust the polynomial degree where neuronal waves are located.

## 6.2 Brain Stem

In this test, we evaluate the performance of the proposed  $p$ -adaptive PolyDG method in a geometrically complex domain representing a sagittal section of a human brain stem reconstructed starting from structural Magnetic Resonance Images (MRI) from the OASIS-3 database [63]. The considered brain sagittal section has a maximum longitudinal length of 7.5 cm and a vertical length of 7 cm. Here, we simulate pathological seizures dynamics in a brain section characterized by heterogeneous tissue, such as grey and white matter. As in the previous sections, we consider an external forcing term defined as in Figure 18b. An initial fine triangular grid has been agglomerated using the approach based on neural networks proposed in [64], leading to a polytopal mesh of 3523 elements, cf. Figure (21a), where we also show the subdivision into grey and white matter. Out of the 3523 polygons constituting our agglomerated mesh, 2012 are in the white matter and the remaining in the grey matter. In Figure (21b), we

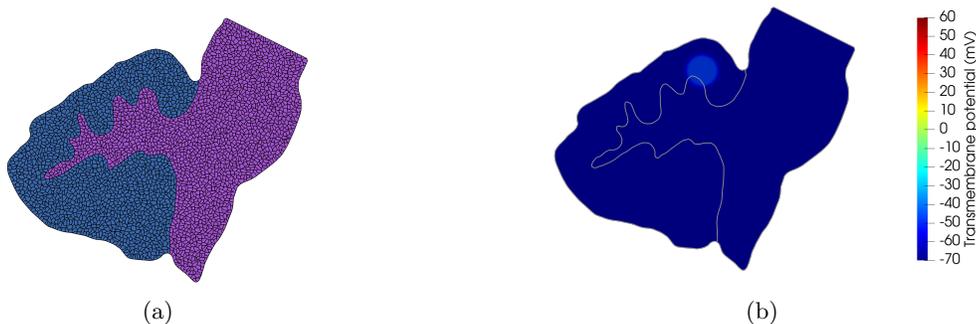


Figure 21: Test case 4. Agglomerated polytopal grid consisting of 3523 elements of the sagittal section of a human brain stem with differentiation of tissues as white matter (purple) and grey matter (blue) (left); pathological initial condition for the transmembrane potential ( $u_0 = -50$  [mV] in  $\Omega_0$ ).

report the unstable initial condition. The behavior of white and grey matter in response to transmembrane ionic imbalances or external stimuli differs because of the preponderance of cell bodies in grey matter and axons in white matter. This makes the white matter anisotropic; in this simulation, we consider the following values for the conductivity tensors for grey and white matter

$$\Sigma_{\text{GM}} = \begin{bmatrix} 0.7354 & 0 \\ 0 & 0.7354 \end{bmatrix}, \quad \Sigma_{\text{WM}} = \begin{bmatrix} 1.7354 & 0 \\ 0 & 1.2854 \end{bmatrix}, \quad (24)$$

respectively. For this simulation, we consider the same values of the parameters and initial condition as before, cf. Table 5 and Table 4. As before, the initial condition is discretized  $p_{\max} = 5$  in all the mesh elements, corresponding to  $7.3e + 4$  degrees of freedom. Figure 22 shows the computed transmembrane potential together with the element-wise polynomial approximation degree and the local error indicator at three different time snapshots  $t = 1.0, 9.4, 17.2$  [ms]. Also in this simulation, the proposed  $p$ -adaptive strategy is able to correctly capture the depolarization and polarization wavefronts and automatically and dynamically adapt the local polynomial degree to capture the steep variations of the transmembrane potential, while reducing the computational burden. This is also confirmed by the results reported in Figure 23, where we show the time-evolution of the NDoF. We observe that the  $p$ -adaptive methods significantly reduced the NDoF compared to the corresponding scheme employing a uniform polynomial approximation degree (for which  $\text{NDoF} = 7.3e + 4$ ). After 14ms, the NDoF drops to  $4e + 4$ , reducing the total number of degrees of freedom by approximately 40%. Similar to the previous test case, we observe a gradual increase in the NDoF over time, driven by the appearance of multiple wavefronts that need to be properly tracked. The same trend can be observed in the evolution of the number of elements checked by the  $p$ -adaptive scheme and the number of mesh elements where the polynomial degree is equal to 1, 2, 3, 4, and 5 (cf. Figure 23 center and right). Once the first wavefront has passed the whole domain, a stable distribution is reached both in terms of the total NDoF and the polynomial approximation degree distribution.

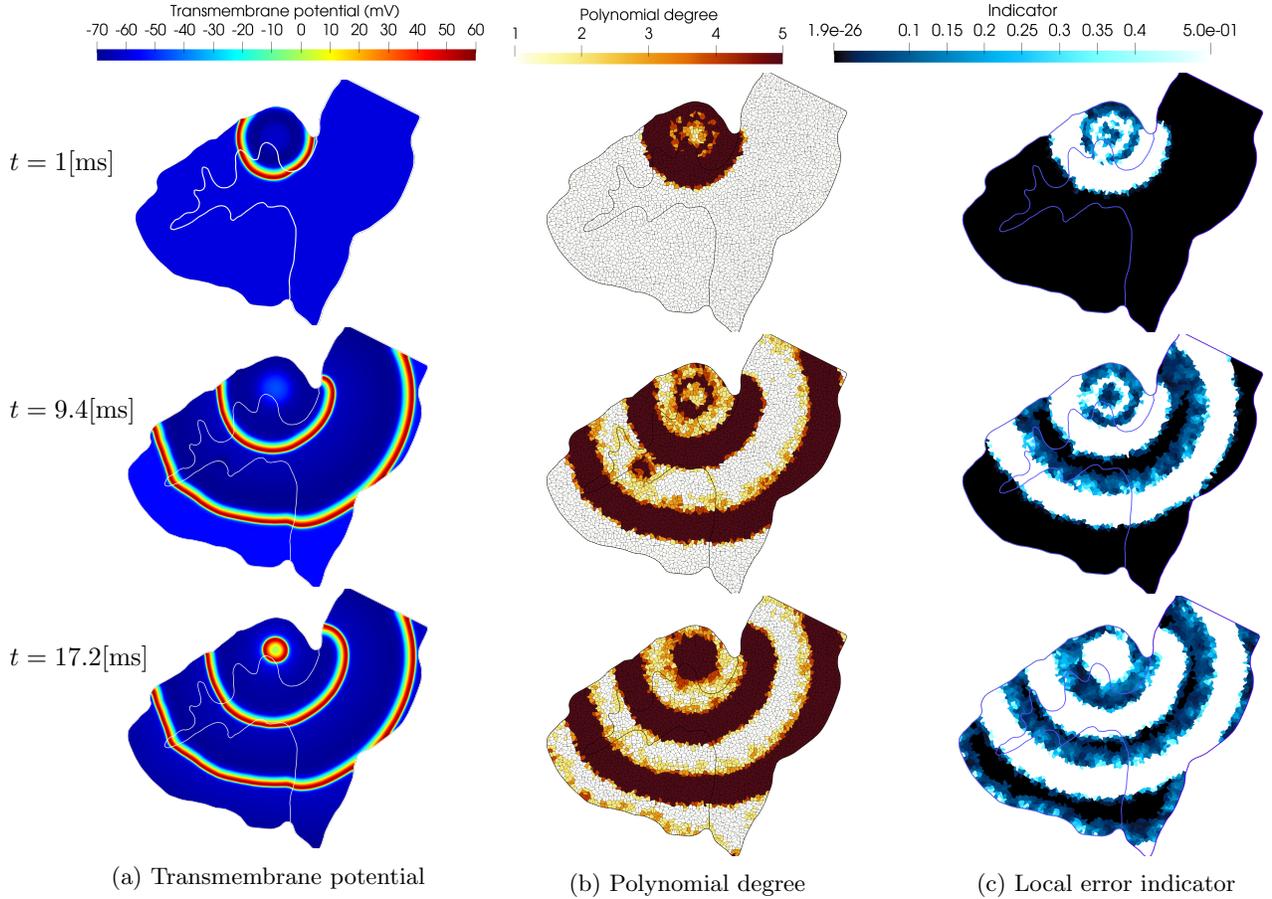


Figure 22: Test case 4. Computed transmembrane potential, element-wise polynomial approximation degree and local error indicator at three different time snapshots  $t = 1.0, 9.4, 17.2$  [ms].

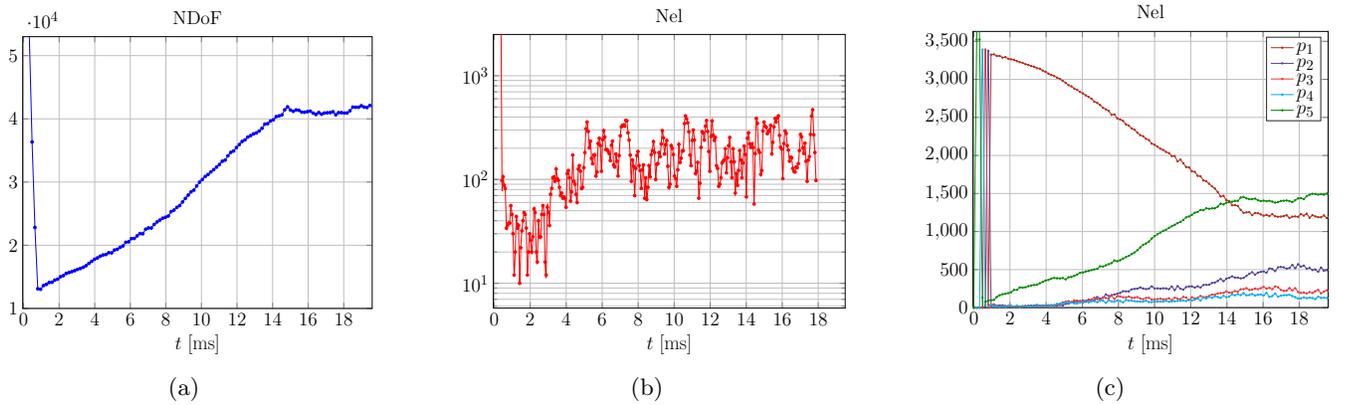


Figure 23: Test case 4. Left: Evolution of the number of degrees of freedom (NDoF) as a function of time, driven by our  $p$ -adaptive algorithm. Center: Evolution of the number of elements (Nel) where the local polynomial degree is updated over time. Right: Evolution of the number of elements (Nel) with local polynomial degree equal to 1, 2, 3, 4, 5 over time. At the initial time, all the elements are discretized with  $p_{\max} = 5$ .

## 7 Conclusion

This work addressed some numerical challenges in the numerical approximation of neuronal electrophysiology phenomena characterized by complex geometries, multiscale phenomena, and steep wavefronts. The considered mathematical model consists of the monodomain equation coupled with a specific neuronal ionic model to simulate the

pathological behavior of the transmembrane potential. To overcome the associated computational burden, we proposed a novel  $p$ -adaptive algorithm based on the discontinuous Galerkin method on polytopal meshes coupled with Crank-Nicolson time marching scheme. Our automatic  $p$ -adaptive strategy is driven by a suitable a-posteriori error estimator to dynamically adjust the local polynomial approximation degree, enabling the dynamic and efficient capture of wavefronts while simultaneously reducing the total number of degrees of freedom. Thanks to a novel clustering algorithm, we automatically select a threshold value to efficiently identify elements for adaptive updates, further improving efficiency. We tested the proposed  $p$ -adaptive algorithm on a wide set of benchmark test cases, demonstrating that the number of degrees of freedom can be significantly reduced while preserving the accuracy of the numerical solution. We also demonstrated the capabilities of the proposed  $p$ -adaptive method in approximating physio-pathological scenarios of brain electrophysiology. The results highlighted the  $p$ -adaptive method's capability to capture the complex interplay of physiological and pathological phenomena at a reasonable computational cost.

## Declaration of competing interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this article.

## Acknowledgments

The brain MRI images were provided by OASIS-3: Longitudinal Multimodal Neuroimaging: Principal Investigators: T. Benzinger, D. Marcus, J. Morris; NIH P30 AG066444, P50 AG00561, P30 NS09857781, P01 AG026276, P01 AG003991, R01 AG043434, UL1 TR000448, R01 EB009352. AV-45 doses were provided by Avid Radiopharmaceuticals, a wholly-owned subsidiary of Eli Lilly.

## References

- [1] M. Viceconti, M. De Vos, S. Mellone, and L. Geris, “Position paper from the digital twins in healthcare to the virtual human twin: a moon-shot project for digital health research”, *IEEE Journal of Biomedical and Health Informatics*, 2023.
- [2] A. Goriely, M. G. Geers, G. A. Holzapfel, J. Jayamohan, A. Jérusalem, S. Sivaloganathan, W. Squier, J. A. van Dommelen, S. Waters, and E. Kuhl, “Mechanics of the brain: perspectives, challenges, and opportunities”, *Biomechanics and modeling in mechanobiology*, vol. 14, pp. 931–965, 2015.
- [3] K.-A. Mardal, M. E. Rognes, T. B. Thompson, and L. M. Valnes, *Mathematical modeling of the human brain: from magnetic resonance images to finite element simulation*. Springer Nature, 2022.
- [4] H. Kager, W. Wadman, and G. Somjen, “Seizure-like afterdischarges simulated in a model neuron”, *Journal of Computational Neuroscience*, vol. 22, pp. 105–128, 2007.
- [5] G. Somjen, H. Kager, and W. Wadman, “Computer simulations of neuron-glia interactions mediated by ion flux”, *Journal of Computational Neuroscience*, vol. 25, pp. 349–365, 2008.
- [6] R. A. Stefanescu, R. Shivakeshavan, and S. S. Talathi, “Computational models of epilepsy”, *Seizure*, vol. 21, no. 10, pp. 748–759, 2012.
- [7] J. Schreiner and K.-A. Mardal, “Simulating epileptic seizures using the bidomain model”, *Scientific Reports*, vol. 12, no. 1, p. 10065, 2022.
- [8] M. J. Sætra, A. J. Ellingsrud, and M. E. Rognes, “Neural activity induces strongly coupled electro-chemo-mechanical interactions and fluid flow in astrocyte networks and extracellular space—a computational study”, *PLoS Computational Biology*, vol. 19, no. 7, p. e1010996, 2023.
- [9] P. Benedusi, A. J. Ellingsrud, H. Herlyng, and M. E. Rognes, “Scalable approximation and solvers for ionic electrodiffusion in cellular geometries”, *SIAM Journal on Scientific Computing*, vol. 46, no. 5, pp. B725–B751, 2024.

- [10] N. Berre, M. E. Rognes, and A. Massing, “Cut finite element discretizations of cell-by-cell emi electrophysiology models”, *SIAM Journal on Scientific Computing*, vol. 46, no. 4, pp. B527–B553, 2024.
- [11] C. B. Leimer Saglio, S. Pagani, M. Corti, and P. F. Antonietti, “A high-order discontinuous Galerkin method for the numerical modeling of epileptic seizures”, *arXiv preprint arXiv:2401.14310*, 2024.
- [12] E. M. Izhikevich, “Neural excitability, spiking and bursting”, *International Journal of Bifurcation and Chaos*, vol. 10, no. 06, pp. 1171–1266, 2000.
- [13] Y. Wei, G. Ullah, and S. J. Schiff, “Unification of neuronal spikes, seizures, and spreading depression”, *Journal of Neuroscience*, vol. 34, no. 35, pp. 11733–11743, 2014.
- [14] L. Øyehaug, I. Østby, C. M. Lloyd, S. W. Omholt, and G. T. Einevoll, “Dependence of spontaneous neuronal firing and depolarisation block on astroglial membrane transport mechanisms”, *Journal of Computational Neuroscience*, vol. 32, pp. 147–165, 2012.
- [15] R. S. Fisher, W. V. E. Boas, W. Blume, C. Elger, P. Genton, P. Lee, and J. Engel Jr, “Epileptic seizures and epilepsy: definitions proposed by the international league against epilepsy (ilae) and the international bureau for epilepsy (ibe)”, *Epilepsia*, vol. 46, no. 4, pp. 470–472, 2005.
- [16] A. L. Hodgkin and A. F. Huxley, “Propagation of electrical signals along giant nerve fibres”, *Proceedings of the Royal Society of London. Series B-Biological Sciences*, vol. 140, no. 899, pp. 177–183, 1952.
- [17] J. R. Cressman, G. Ullah, J. Ziburkus, S. J. Schiff, and E. Barreto, “The influence of sodium and potassium dynamics on excitability, seizures, and the stability of persistent states: I. Single neuron dynamics”, *Journal of Computational Neuroscience*, vol. 26, pp. 159–170, 2009.
- [18] E. Barreto and J. R. Cressman, “Ion concentration dynamics as a mechanism for neuronal bursting”, *Journal of Biological Physics*, vol. 37, pp. 361–373, 2011.
- [19] K. H. Jæger and A. Tveito, “Deriving the bidomain model of cardiac electrophysiology from a cell-based model; properties and comparisons”, *Frontiers in Physiology*, vol. 12, p. 811029, 2022.
- [20] P. Colli Franzone, L. Guerri, and S. Rovida, “Wavefront propagation in an activation model of the anisotropic cardiac tissue: asymptotic analysis and numerical simulations”, *Journal of mathematical biology*, vol. 28, pp. 121–176, 1990.
- [21] E. J. Vigmond, F. Aguel, and N. A. Trayanova, “Computational techniques for solving the bidomain equations in three dimensions”, *IEEE Transactions on Biomedical Engineering*, vol. 49, no. 11, pp. 1260–1269, 2002.
- [22] P. C. Franzone, L. Pavarino, and B. Taccardi, “Simulating patterns of excitation, repolarization and action potential duration with cardiac bidomain and monodomain models”, *Mathematical biosciences*, vol. 197, no. 1, pp. 35–66, 2005.
- [23] S. Göktepe and E. Kuhl, “Computational modeling of cardiac electrophysiology: a novel finite element approach”, *International journal for numerical methods in engineering*, vol. 79, no. 2, pp. 156–178, 2009.
- [24] G. Seemann, F. Sachse, M. Karl, D. Weiss, V. Heuveline, and O. Dössel, “Framework for modular, flexible and efficient solving the cardiac bidomain equations using PETSc”, in *Progress in industrial mathematics at ECMI 2008*, pp. 363–369, Springer, 2010.
- [25] C. Nagaiah, K. Kunisch, and G. Plank, “Numerical solution for optimal control of the reaction-diffusion equations in cardiac electrophysiology”, *Computational Optimization and Applications*, vol. 49, pp. 149–178, 2011.
- [26] C. J. Arthurs, M. J. Bishop, and D. Kay, “Efficient simulation of cardiac electrical propagation using high order finite elements”, *Journal of Computational Physics*, vol. 231, no. 10, pp. 3946–3962, 2012.
- [27] S. Pezzuto, J. Hake, and J. Sundnes, “Space-discretization error analysis and stabilization schemes for conduction velocity in cardiac electrophysiology”, *International Journal for Numerical Methods in Biomedical Engineering*, vol. 32, no. 10, p. e02762, 2016.

- [28] A. Quarteroni, T. Lassila, S. Rossi, and R. Ruiz-Baier, “Integrated heart—coupling multiscale and multiphysics models for the simulation of the cardiac function”, *Computer Methods in Applied Mechanics and Engineering*, vol. 314, pp. 345–407, 2017.
- [29] P. C. Franzone, L. F. Pavarino, and S. Scacchi, *Mathematical cardiac electrophysiology*, vol. 13. Springer, 2014.
- [30] A. Quarteroni, A. Manzoni, C. Vergara, *et al.*, *Mathematical modelling of the human cardiovascular system: data, numerical approximation, clinical applications*, vol. 33. Cambridge University Press, 2019.
- [31] R. Bordas, B. Carpentieri, G. Fotia, F. Maggio, R. Nobes, J. Pitt-Francis, and J. Southern, “Simulation of cardiac electrophysiology on next-generation high-performance computers”, *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 367, no. 1895, pp. 1951–1969, 2009.
- [32] C. D. Cantwell, S. Yakovlev, R. M. Kirby, N. S. Peters, and S. J. Sherwin, “High-order spectral/hp element discretisation for reaction–diffusion problems on surfaces: Application to cardiac electrophysiology”, *Journal of Computational Physics*, vol. 257, pp. 813–829, 2014.
- [33] P. C. Africa, M. Salvador, P. Gervasio, L. Dede, and A. Quarteroni, “A matrix-free high-order solver for the numerical solution of cardiac electrophysiology”, *Journal of Computational Physics*, vol. 478, p. 111984, 2023.
- [34] A. S. Patelli, L. Dedè, T. Lassila, A. Bartzzaghi, and A. Quarteroni, “Isogeometric approximation of cardiac electrophysiology models on surfaces: an accuracy study with application to the human left atrium”, *Computer Methods in Applied Mechanics and Engineering*, vol. 317, pp. 248–273, 2017.
- [35] M. Bucelli, M. Salvador, A. Quarteroni, *et al.*, “Multipatch isogeometric analysis for electrophysiology: simulation in a human heart”, *Computer Methods in Applied Mechanics and Engineering*, vol. 376, p. 113666, 2021.
- [36] M. Torre, S. Morganti, A. Nitti, M. D. de Tullio, F. S. Pasqualini, and A. Reali, “An efficient isogeometric collocation approach to cardiac electrophysiology”, *Computer Methods in Applied Mechanics and Engineering*, vol. 393, p. 114782, 2022.
- [37] P. F. Antonietti, L. Dedè, G. Loli, M. Montardini, G. Sangalli, and P. Tesini, “Space–time isogeometric analysis of cardiac electrophysiology”, *Computer Methods in Applied Mechanics and Engineering*, vol. 441, p. 117957, 2025.
- [38] Y. Coudière and R. Turpault, “A domain decomposition strategy for a very high-order finite volumes scheme applied to cardiac electrophysiology”, *Journal of Computational Science*, vol. 37, p. 101025, 2019.
- [39] G. R. de Souza, S. Pezzuto, and R. Krause, “High-order parallel-in-time method for the monodomain equation in cardiac electrophysiology”, *arXiv preprint arXiv:2405.19994*, 2024.
- [40] J. M. Hoermann, C. Bertoglio, M. Kronbichler, M. R. Pfaller, R. Chabiniok, and W. A. Wall, “An adaptive hybridizable discontinuous Galerkin approach for cardiac electrophysiology”, *International Journal for Numerical Methods in Biomedical Engineering*, vol. 34, no. 5, p. e2959, 2018.
- [41] F. Botta, M. Calafà, P. C. Africa, C. Vergara, and P. F. Antonietti, “High-order discontinuous galerkin methods for the monodomain and bidomain models”, *Mathematics in Engineering*, vol. 6, no. 6, p. 726 – 741, 2024.
- [42] P. F. Antonietti, S. Giani, and P. Houston, “Hp-version composite discontinuous galerkin methods for elliptic problems on complicated domains”, *SIAM Journal on Scientific Computing*, vol. 35, no. 3, pp. A1417–A1439, 2013.
- [43] F. Bassi, L. Botti, A. Colombo, D. Di Pietro, and P. Tesini, “On the flexibility of agglomeration based physical space discontinuous Galerkin discretizations”, *Journal of Computational Physics*, vol. 231, no. 1, p. 45 – 65, 2012.
- [44] A. Cangiani, E. H. Georgoulis, and P. Houston, “hp-version discontinuous Galerkin methods on polygonal and polyhedral meshes”, *Mathematical Models and Methods in Applied Sciences*, vol. 24, no. 10, pp. 2009–2041, 2014.
- [45] A. Cangiani, Z. Dong, and E. Georgoulis, “hp-version discontinuous Galerkin methods on essentially arbitrarily-shaped elements”, *Mathematics of Computation*, vol. 91, no. 333, pp. 1–35, 2022.

- [46] J. M. Hoermann, C. Bertoglio, M. Kronbichler, M. R. Pfaller, R. Chabiniok, and W. A. Wall, “An adaptive hybridizable discontinuous galerkin approach for cardiac electrophysiology”, *International journal for numerical methods in biomedical engineering*, vol. 34, no. 5, p. e2959, 2018.
- [47] E. M. Cherry, H. S. Greenside, and C. S. Henriquez, “A space-time adaptive method for simulating complex cardiac dynamics”, *Physical Review Letters*, vol. 84, no. 6, p. 1343, 2000.
- [48] D. Schötzau and T. P. Wihler, “A posteriori error estimation for hp-version time-stepping methods for parabolic partial differential equations”, *Numerische Mathematik*, vol. 115, no. 3, pp. 475–509, 2010.
- [49] A. Cangiani, Z. Dong, E. H. Georgoulis, and P. Houston, “hp-version discontinuous Galerkin methods for advection-diffusion-reaction problems on polytopic meshes”, *ESAIM: Mathematical Modelling and Numerical Analysis*, vol. 50, no. 3, pp. 699–725, 2016.
- [50] M. Potse, B. Dubé, J. Richer, A. Vinet, and R. M. Gulrajani, “A comparison of monodomain and bidomain reaction-diffusion models for action potential propagation in the human heart”, *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 12, pp. 2425–2435, 2006.
- [51] J. Sundnes, B. F. Nielsen, K. A. Mardal, X. Cai, G. T. Lines, and A. Tveito, “On the computational complexity of the bidomain and the monodomain models of electrophysiology”, *Annals of Biomedical Engineering*, vol. 34, pp. 1088–1097, 2006.
- [52] B. S. Gutkin and G. B. Ermentrout, “Dynamics of membrane excitability determine interspike interval variability: a link between spike generation mechanisms and cortical spike train statistics”, *Neural Computation*, vol. 10, no. 5, pp. 1047–1065, 1998.
- [53] S. Salsa and G. Verzini, *Partial differential equations in action: from modelling to theory*, vol. 147. Springer Nature, 4th ed., 2022.
- [54] D. D. A. Pietro and J. Droniou, *The Hybrid High-Order Method for Polytopal Meshes: Design, Analysis, and Applications*. Springer, 1st ed., 2020.
- [55] D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini, “Unified analysis of discontinuous Galerkin methods for elliptic problems”, *SIAM Journal on Numerical Analysis*, vol. 39, no. 5, pp. 1749–1779, 2002.
- [56] J. Sundnes, G. T. Lines, X. Cai, B. F. Nielsen, K.-A. Mardal, and A. Tveito, *Computing the electrical activity in the heart*, vol. 1. Springer Science & Business Media, 2007.
- [57] A. Cangiani, Z. Dong, and E. H. Georgoulis, “A posteriori error estimates for discontinuous Galerkin methods on polygonal and polyhedral meshes”, *SIAM Journal on Numerical Analysis*, vol. 61, no. 5, pp. 2352–2380, 2023.
- [58] A. Cangiani, E. H. Georgoulis, and M. Sabawi, “A posteriori error analysis for implicit–explicit hp-discontinuous Galerkin timestepping methods for semilinear parabolic problems”, *Journal of Scientific Computing*, vol. 82, no. 2, p. 26, 2020.
- [59] P. F. Antonietti, S. Bonetti, M. Botti, M. Corti, I. Fumagalli, and I. Mazzieri, “lymph: discontinuous polytopal methods for multi-physics differential problems”, *ACM Transactions on Mathematical Software*, vol. 51, no. 1, pp. 1–22, 2025.
- [60] C. Talischi, G. H. Paulino, A. Pereira, and I. F. Menezes, “Polymesher: a general-purpose mesh generator for polygonal elements written in Matlab”, *Structural and Multidisciplinary Optimization*, vol. 45, pp. 309–328, 2012.
- [61] A. H. Erhardt, K.-A. Mardal, and J. E. Schreiner, “Dynamics of a neuron–glia system: the occurrence of seizures and the influence of electroconvulsive stimuli: A mathematical and numerical study”, *Journal of Computational Neuroscience*, vol. 48, pp. 229–251, 2020.
- [62] G. Ullah, J. R. Cressman Jr, E. Barreto, and S. J. Schiff, “The influence of sodium and potassium dynamics on excitability, seizures, and the stability of persistent states: Ii. Network and glial dynamics”, *Journal of Computational Neuroscience*, vol. 26, pp. 171–183, 2009.

- [63] P. J. LaMontagne, T. L. Benzinger, J. C. Morris, S. Keefe, R. Hornbeck, C. Xiong, E. Grant, J. Hassenstab, K. Moulder, A. G. Vlassenko, *et al.*, “Oasis-3: longitudinal neuroimaging, clinical, and cognitive dataset for normal aging and alzheimer disease”, *MedRxiv*, pp. 2019–12, 2019.
- [64] P. F. Antonietti, N. Farenga, E. Manuzzi, G. Martinelli, and L. Saverio, “Agglomeration of polygonal grids using graph neural networks with applications to multigrid solvers”, *Computers and Mathematics with Applications*, vol. 154, p. 45–57, 2024.