Received: 31 August 2017 | Revised: 13 December 2017 | Accepted: 27 December 2017
DOI: 10.1002/cnm.2959

CMBE17: MS19 - SELECTED PAPERS

WILEY

An adaptive hybridizable discontinuous Galerkin approach for cardiac electrophysiology

Julia M. Hoermann[1] | Cristóbal Bertoglio[2,3] | Martin Kronbichler[1] | Martin R. Pfaller[1] | Radomir Chabiniok[4,5,6] | Wolfgang A. Wall[1]

[1] Institute for Computational Mechanics, Technical University Munich, Boltzmannstr 15, Garching b. München, 85748, Germany

[2] Center for Mathematical Modeling, Universidad de Chile, Beaucheff 851, Santiago 8370456, Chile

[3] Johann Bernoulli Institute, University of Groningen, Nijenborgh 9 Groningen, 9747 HZ, Netherlands

[4] Inria, Paris-Saclay University, Palaiseau, France

[5] LMS, Ecole Polytechnique, CNRS, Paris-Saclay University, Palaiseau, France

[6] School of Biomedical Engineering and Imaging Sciences (BMEIS), St Thomas' Hospital, King's College, London, UK

Correspondence
Martin Kronbichler, Institute for Computational Mechanics, Technical University Munich, Boltzmannstr 15, 85748 Garching b. München, Germany.
Email: kronbichler@lnm.mw.tum.de

Abstract

Cardiac electrophysiology simulations are numerically challenging because of the propagation of a steep electrochemical wave front and thus require discretizations with small mesh sizes to obtain accurate results. In this work, we present an approach based on the hybridizable discontinuous Galerkin method (HDG), which allows an efficient implementation of high-order discretizations into a computational framework. In particular, using the advantage of the discontinuous function space, we present an efficient p-adaptive strategy for accurately tracking the wave front. The HDG allows to reduce the overall degrees of freedom in the final linear system to those only on the element interfaces. Additionally, we propose a rule for a suitable integration accuracy for the ionic current term depending on the polynomial order and the cell model to handle high-order polynomials. Our results show that for the same number of degrees of freedom, coarse high-order elements provide more accurate results than fine low-order elements. Introducing p-adaptivity further reduces computational costs while maintaining accuracy by restricting the use of high-order elements to resolve the wave front. For a patient-specific simulation of a cardiac cycle, p-adaptivity reduces the average number of degrees of freedom by 95% compared to the nonadaptive model. In addition to reducing computational costs, using coarse meshes with our p-adaptive high-order HDG method also simplifies practical aspects of mesh generation and postprocessing.

KEYWORDS

cardiac electrophysiology, finite element method, hybridizable discontinuous Galerkin

1 | INTRODUCTION

Biophysical and computational modeling of the heart has been proposed and actively pursued as a tool for accelerating cardiovascular research, and translation to clinic to personalize the care is very promising.[1] One of the challenges modelers have to face is the high cost of the computations, in particular when moving towards the whole heart modeling and coupling different physics and scales. To develop methods that reduce the computing time while keeping numerical accuracy is essential for speeding up fundamental research and most importantly the translation of modeling into clinical practice.

Cardiac electrophysiology simulations are classically based on monodomain or bidomain reaction-diffusion equations for the transmembrane electrical potential, which are coupled to a model for the gating dynamics of the ionic channels on the cell membrane. Because of the fast upstroke of the action potential, which is caused by voltage-dependent sodium channels, a numerically robust calculation of the propagation of the wave across the tissue is well known to be computationally challenging. This rapid increase of the transmembrane potential in one cell over a few milliseconds results also in a steep wave front in space,[2] requiring high resolution temporal and spatial discretizations.[3]

Galerkin methods are popular in cardiac electrophysiology because of their simple applicability to complex geometries and higher-order discretizations.[4-12] Attempts have been made to improve the performance of the numerical method by using h-adaptivity in space,[13-15] adaptivity in time,[6,16,17] and in both space and time.[18-22] Adaptivity in space based on low-order elements require remeshing during the calculation, which involves a considerable computational effort. Furthermore, this approach has to couple the numerical solver with a mesh generator. Therefore, studies about efficient simulations with high-order elements and high-order p-adaptive elements were performed for continuous Galerkin (CG) in Arthurs et al.[23,24] High-order discretizations have been shown to be more efficient than low-order ones for a variety of contexts, ie, they achieve better accuracy with fewer degrees of freedom and less computational cost.[24,25] However, the adaption of the polynomial degree in CG faces additional challenges, eg, for preserving continuity between the elements.[26]

To the authors' best knowledge, discontinuous Galerkin (DG) approaches have not received any attention yet for the electrophysiology problem. The discontinuous setting allows a straightforward implementation of global or local high-order discretizations and/or local mesh refinement.

The hybridizable discontinuous Galerkin (HDG) method[27] is a special case of the DG family of methods that allows a reduction of the degrees of freedom that appear in the final linear system by static condensation on the element to those defined on the faces between the elements. The goal of this work is to present a suitable HDG formulation for cardiac electrophysiology, study the numerical performance of HDG in this context, and to compare it against standard CG approaches.

The remainder of the paper is organized as follows. In Section 2, we summarize the electrophysiology model problem and recall the CG and HDG discretization approaches. We then discuss the strategies used for the calculation of the ionic current term with focus on high polynomial degrees of the function spaces. In Section 3, we detail the setups of the numerical experiments and the cell model we use, and exemplify the ionic current integration rule for a specific cell model. We also determine a practical choice for the stabilization parameter in HDG. The numerical results comprise an academic geometry for benchmarking purposes and a simulation on a patient-specific heart, where both CG and (p-adaptive) HDG methods are assessed.

2 | METHODS

2.1 | Electrophysiology equations

The classical monodomain model is given in the domain $\Omega \subset \mathbb{R}^3$ in the following way: Find the transmembrane potential $u : \Omega \times t \rightarrow \mathbb{R}$ and the gating variables $\mathbf{w} : \Omega \times t \rightarrow \mathbb{R}^m$ such that

$$\left\{ \begin{aligned} \chi(C_m \partial_t u - I_{ion}(u, \mathbf{w})) &= \nabla \cdot (\mathbf{D} \nabla u) && \text{in } \Omega \times (0, T], \\ \partial_t \mathbf{w} - \mathbf{g}(u, \mathbf{w}) &= 0 && \text{in } \Omega \times (0, T], \\ \mathbf{n} \cdot (\mathbf{D} \nabla u) &= 0 && \text{on } \partial\Omega \times (0, T], \\ u(x, 0) &= u_0(x) && \text{in } \Omega, \\ \mathbf{w}(x, 0) &= \mathbf{w}_0(x) && \text{in } \Omega. \end{aligned} \right. \tag{1}$$

The physical (given) constants are the ratio of membrane area per tissue volume χ, the local membrane capacitance C_m, the electrical conductivity tensor $\mathbf{D}$, the initial conditions u_0 and w_0, and the outward pointing unit normal vector $\mathbf{n}$ to $\partial\Omega$. The given nonlinear functions $I_{ion} : \mathbb{R}^{m+1} \rightarrow \mathbb{R}$ and $\mathbf{g} : \mathbb{R}^{m+1} \rightarrow \mathbb{R}^m$ describe the total ionic current flow through the membrane of the myocyte and the kinetics of the gating variables as defined by the cell model, respectively.

2.2 | Time semidiscretization

For the sake of clarity of the presentation, we first introduce the time marching scheme used. It is important to remark that the choice is independent on the spatial discretization. We adopt a semi-implicit backward Euler approach: The diffusion term is discretized implicitly in time, while the reaction term is evaluated explicitly.[28,29] Assuming uniform time intervals $t_0, t_1, \dots$ with $dt = t_n - t_{n-1} \forall n > 0$, the time semidiscrete problem reads, given $u^0, \mathbf{w}^0$, find the transmembrane potential $u^n : \Omega \to \mathbb{R}$ and the gating variables $\mathbf{w}^n : \Omega \to \mathbb{R}^m$ for $n > 0$ such that

$$\begin{cases} \chi(C_m \frac{u^n - u^{n-1}}{dt} - I_{ion}(u^{n-1}, \mathbf{w}^{n-1})) = \nabla \cdot (\mathbf{D}\nabla u^n) & \text{in } \Omega, \\ \frac{\mathbf{w}^n - \mathbf{w}^{n-1}}{dt} - \mathbf{g}(u^n, \mathbf{w}^n) = 0 & \text{in } \Omega, \\ \mathbf{n} \cdot (\mathbf{D}\nabla u^n) = 0 & \text{on } \partial\Omega. \end{cases} \tag{2}$$

Note that this scheme allows important gains in computational efficiency: The problem for u^n is linear, and hence, system matrices and preconditioners are assembled only once at the beginning of the computation. CFL-like restrictions on the time stepping, which arise from spatial derivatives and are thus mesh-dependent, are avoided because of the implicit treatment of the diffusion terms. The latter can be of particular interest when dealing with spatial adaptivity as it is done in this work. Finally, note that since the cell model used in this work is linear in the gating variables, and hence, only linear problems have to be solved in the implicit parts.

2.3 | Continuous Galerkin approximation

In this section, we describe the spatial discretization of Equation 2 by a continuous Galerkin approximation.

Let us assume a domain Ω and a triangulation $\mathcal{T}_h$ of this domain with elements of characteristic size h. We will denote each of these elements as K. Let us recall the standard function spaces V_h^p of piecewise continuous polynomials of degree p:

$$V_h^p = \{v \in H^1(\Omega) : v|_K \in \mathcal{P}^p(K) \text{ for element } K \in \mathcal{T}_h\},$$

where $H^1(\Omega)$ denotes the Sobolev space $W^{1,2}(\Omega)$ and $\mathcal{P}^p(K)$ the set of polynomials of maximal degree p on a domain K. The fully discretized problem read then as: Find $u_h^n \in V_h^p$ such that

$$C_m \int_{\mathcal{T}_h} \frac{u_h^n - u_h^{n-1}}{dt} \varphi_h \, d\mathbf{x} + \frac{1}{\chi} \int_{\mathcal{T}_h} \mathbf{D}\nabla u_h^n \cdot \nabla \varphi_h \, d\mathbf{x} = \sum_K \tilde{I}_K^{n-1}(u_h^{n-1}, \varphi_h) \tag{3}$$

for all $\varphi_h \in V_h^p$, and with I_K given, resulting from the approximation of the integral

$$\tilde{I}_K^{n-1}(v, \phi) \approx I_K^{n-1}(v, \phi) := \int_K I_{ion}(v, \mathbf{w}^{n-1}) \phi \, d\mathbf{x}, \quad v, \phi : \Omega \to \mathbb{R}. \tag{4}$$

In literature, there exist different methods to compute this approximation, which result in different spatial discretizations of the gating variables $\mathbf{w}^n$. Since these can indeed be applied to any type of Galerkin scheme, our specific choice is detailed in Section 2.5.

2.4 | Hybridizable discontinuous Galerkin approximation

The derivation of the HDG approximation is based on the work of Nguyen et al[30] for the linear convection-diffusion equation. We will briefly introduce its derivation, starting from the strong form of the mixed formulation for the diffusion term of Equation 2, ie,

$$\begin{cases} C_m \frac{u^n - u^{n-1}}{dt} - \frac{1}{\chi} \nabla \cdot \mathbf{q}^n = I_{ion}(u^{n-1}, \mathbf{w}^{n-1}) \\ \mathbf{q}^n - \mathbf{D}\nabla u^n = 0 \end{cases} \tag{5}$$

and its weak form on the element K

$$\begin{cases} C_m \int_K \frac{u^n - u^{n-1}}{dt} \varphi \, d\mathbf{x} + \frac{1}{\chi} \int_K \mathbf{q}^n \cdot \nabla \varphi \, d\mathbf{x} - \frac{1}{\chi} \int_{\partial K} \mathbf{n} \cdot \hat{\mathbf{q}}^n \varphi \, d\mathbf{x} = I_K^{n-1}(u^{n-1}, \varphi), \\ \int_K \mathbf{D}^{-1} \mathbf{q}^n \cdot \boldsymbol{\psi} \, d\mathbf{x} + \int_K u \nabla \cdot \boldsymbol{\psi} \, d\mathbf{x} - \int_{\partial K} \lambda^n \mathbf{n} \cdot \boldsymbol{\psi} \, d\mathbf{x} = 0, \end{cases} \tag{6}$$

for all $\varphi \in L_2(K), \boldsymbol{\psi} \in [L_2(K)]^d$. Here, $\lambda^n \in L_2(e)$ is the single-valued trace of u^n over the face $e \in \mathcal{E}_h$. $\mathcal{E}_h$ is the set of all interior and boundary faces, where an interior face is defined as $e = \partial K_1 \cap \partial K_2$ between 2 elements K_1 and K_2 and a boundary face is defined as $e = \partial K \cap \partial \Omega$ for an element K. The vector $\hat{\mathbf{q}}^n$ is the trace on ∂K of $\mathbf{q}^n$, which is defined in the HDG method by

$$\hat{\mathbf{q}}^n = \mathbf{q}^n - \tau(u^n - \lambda^n)\mathbf{n} \quad \text{on } \partial K, \tag{7}$$

with the stabilization parameter $\tau > 0$.

Finally, we close the system by enforcing the jump across of the trace of the flux $\hat{\mathbf{q}}$ along each edge e to be zero, ie,

$$[[\hat{\mathbf{q}}^n]] = \mathbf{0} \text{ ,on all } e \in \mathcal{E}_h. \tag{8}$$

For the spatial discretization, we define the finite element spaces of piecewise discontinuous polynomials on the element's volume

$$W_h^p = \{v \in L^2(\Omega) \, : \, v|_K \in \mathcal{P}^p(K) \ \text{ for element } \ K \in \mathcal{T}_h\}$$

and on the elements edges

$$M_h^p = \{\mu \in L^2(\mathcal{E}_h) \, : \, \mu|_e \in \mathcal{P}^p(e) \text{ for face } e \in \mathcal{E}_h\}.$$

With these new definitions, we can now formulate the fully discretized system of equations corresponding to the HDG method: Find $(u_h^n, \mathbf{q}_h^n, \lambda_h^n) \in W_h^p \times [W_h^p]^d \times M_h^p$ such that

$$\begin{cases} \sum\limits_K \left(\int_K C_m \frac{u_h^n - u_h^{n-1}}{dt} \varphi_h \, \mathrm{d}\mathbf{x} + \frac{1}{\chi} \int_K \mathbf{q}_h^n \nabla \varphi_h \, \mathrm{d}\mathbf{x} - \frac{1}{\chi} \int_{\partial K} \mathbf{q}_h^n \cdot \mathbf{n} \varphi_h - \tau(u_h^n - \lambda_h^n) \varphi_h \, \mathrm{d}\mathbf{x} \right) = \sum\limits_K \tilde{I}_K^{n-1}(u_h^{n-1}, \varphi_h) \\ \sum\limits_K \left(\int_K \mathbf{D}^{-1} \mathbf{q}_h^n \cdot \boldsymbol{\psi}_h \, \mathrm{d}\mathbf{x} + \int_K u_h^n \nabla \cdot \boldsymbol{\psi}_h \, \mathrm{d}\mathbf{x} - \int_{\partial K} \lambda_h^n \mathbf{n} \cdot \boldsymbol{\psi}_h \, \mathrm{d}\mathbf{x} \right) = 0 \\ \sum\limits_K \left(\int_{\partial K} \mathbf{q}_h^n \cdot \mathbf{n} \mu_h \, \mathrm{d}\mathbf{x} - \int_{\partial K} \tau(u_h^n - \lambda_h^n) \mu_h \, \mathrm{d}\mathbf{x} \right) = 0 \end{cases} \tag{9}$$

for all $(v_h, \boldsymbol{\psi}_h, \mu_h) \in W_h^p \times [W_h^p]^d \times M_h^p$.

One of the advantages of HDG methods is that the size of the final linear system can be considerably reduced using its structure. First, let us rewrite the system of equations as follows

$$\begin{cases} m(u_h^n, \varphi_h)/dt - b(\varphi_h, \mathbf{q}_h^n) + a(u_h^n, \varphi_h) - c(\lambda_h^n, \varphi_h) = f^{n-1}(\varphi_h) \\ d(\mathbf{q}_h^n, \boldsymbol{\psi}_h) + b(u_h^n, \boldsymbol{\psi}_h) - e(\lambda_h^n, \boldsymbol{\psi}_h) = 0 \\ e(\mu_h, \mathbf{q}_h^n) - c(\mu_h, u_h^n) + h(\lambda_h^n, \mu_h) = 0 \end{cases} \tag{10}$$

with the integral forms given by

$$\begin{aligned} m(u, \varphi) &= \textstyle\sum_{\mathrm{K}} \int_K C_m \chi u \varphi \, \mathrm{d}\mathbf{x}, & d(\mathbf{q}, \boldsymbol{\psi}) &= \textstyle\sum_{\mathrm{K}} \int_K \mathbf{D}^{-1} \mathbf{q} \cdot \boldsymbol{\psi} \, \mathrm{d}\mathbf{x}, \\ a(u, \varphi) &= \textstyle\sum_{\mathrm{K}} \int_{\partial K} \tau u \varphi \, \mathrm{d}\mathbf{x}, & e(\lambda, \boldsymbol{\psi}) &= \textstyle\sum_{\mathrm{K}} \int_{\partial K} \lambda \boldsymbol{\psi} \cdot \mathbf{n} \, \mathrm{d}\mathbf{x}, \\ b(u, \boldsymbol{\psi}) &= \textstyle\sum_{\mathrm{K}} \int_K u \nabla \cdot \boldsymbol{\psi} \, \mathrm{d}\mathbf{x}, & f^{n-1}(\varphi) &= m(u_h^{n-1}, \varphi)/dt + \chi \sum\limits_K \tilde{I}_K^{n-1} \left(u_h^{n-1}, \varphi \right), \\ c(\lambda, \varphi) &= \textstyle\sum_{\mathrm{K}} \int_{\partial K} \varphi \tau \lambda \, \mathrm{d}\mathbf{x}, & h(\lambda, \mu) &= \textstyle\sum_{\mathrm{K}} \int_{\partial K} \tau \lambda \mu \, \mathrm{d}\mathbf{x}, \end{aligned} \tag{11}$$

for all $(\mathbf{q}, u, \lambda)$ and (φ, ψ, μ) in $(V_h^p \times W_h^p \times M_h^p)$.

Denoting the vectors of degrees of freedom U^n, Q^n, Λ^n associated to $u_h^n, \mathbf{q}_h^n, \lambda_h^n$, respectively, we can write the resulting system in matrix form as

$$\begin{bmatrix} M/dtU^n \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} A & B & -C \\ -B^T & D & -E \\ -C^T & E^T & H \end{bmatrix} \begin{bmatrix} U^n \\ Q^n \\ \Lambda^n \end{bmatrix} = \begin{bmatrix} F^{n-1} \\ 0 \\ 0 \end{bmatrix}. \tag{12}$$

Since the matrices A, B, D do no couple between different elements, we can use static condensation (Schur complements) on the element level to reduce the global system size.[27] For that purpose, we eliminate the interior variables U^n and Q^n in an element-by-element fashion, so that the remaining degrees of freedom are those of the trace variable Λ^n on the faces of $\mathcal{E}_h$.

2.5 | Ionic current approximation

As stated above, there exist several methods to approximate the element integral of the nonlinear ionic current term I_{ion} in Galerkin formulations for cardiac electrophysiology

$$\tilde{I}_K(u_h, \varphi_h) \approx \int_K I_{ion}(u_h, \mathbf{w})\varphi_h \, \mathrm{d}\mathbf{x}$$

with both u_h and φ_h given through spatial polynomial functions of degree p, see, eg, previous studies[7,9,10] and references therein.

In this work, we adopt a numerical approximation of I_K by defining one gating variable at each quadrature point $\mathbf{x}_\ell$ within the element K, namely,

$$\tilde{I}_K(u_h, \varphi_h) = \sum_{\ell=1}^{b_K} \alpha_\ell I_{ion}(u_h(\mathbf{x}_\ell), \mathbf{w}_\ell)\varphi_h(\mathbf{x}_\ell) \tag{13}$$

with α_ℓ the quadrature weights and b_K the number of integration points on the element K.

This strategy allows to represent the ionic term at a higher degree compared with the ionic current interpolation (see, eg, Krishnamoorthi et al[10]), where the ionic current is discretized using the same space as for the potential. Note that in both the HDG and CG cases, the number of quadrature points to build the term $\tilde{I}_K$ is the same, for a given polynomial order p.

In this work, the number of quadrature points b_K is chosen depending on the polynomial degree p of the finite element spaces on element K. Concretely, assuming that I_{ion} has the potential u involving polynomial expressions in u of degree up to k_{ion}, we propose to choose the quadrature order to integrate exactly polynomials of degree $p(k_{ion}+1)$, since the integral in the weak form comprises not only $I_{ion}(u)$ but also the test function. For the integration rule, we do not take into account the additional nonpolynomial terms in the model, eg, the Heaviside function. Note that for high-order curved geometry descriptions, there could still be a small integration error, but it is expected to be of higher order than the discretization error. For the numerical examples, we use hexahedral and tetrahedral elements with appropriate intergration rules. An example will be shown in Section 3.2. Note that keeping track of $\mathbf{w}_\ell$ and I_{ion} in quadrature points can be interpreted as a polynomial approximation of at least degree $\lceil((k_{ion}+1)p+1)/2\rceil - 1$ in the discontinuous setting by Lagrange polynomials through the integration points.

We want to remark that other ionic current approximations could also be used, eg, where the gating variables are stored at the degrees of freedom of the finite element approximation of the potential. However, additionally, investigating the effect of the type of ionic current approximation is out of the scope of this article.

2.6 | p-Adaptivity for HDG

The HDG method allows for a straightforward adaptation of the degree of the polynomial basis functions since different polynomial degrees between neighboring elements can be independently chosen. A key ingredient is to apply a proper error indicator for selecting the local polynomial order. In this work, we choose a simplified version of the error indicator presented in Arthurs et al[23] based on the jump of the gradient of the potential across the face γ. Therefore, on each element K, the error indicator is defined via the numerical gradient on the faces $\hat{\mathbf{q}}$ as

$$e_K^n = \sum_{\gamma \in \partial K} \|\hat{\mathbf{q}}_\gamma^n \cdot \mathbf{n}\|_{L_2(\gamma)}^2 \frac{1}{A_\gamma} \tag{14}$$

with A_γ the surface area of the face. Then the new degree on the element is calculated via

$$p_K^n = p_K^{n-1} + \left\lceil \frac{1}{\omega} \ln\left(\frac{e_K^{n-1}}{e_{tol}}\right) \right\rceil \tag{15}$$

for a constant $\omega = 1.66$ according to Arthurs et al,[23] which is estimated from convergence rates of high-order CG elements and a given error tolerance e_{tol}. This value did lead to satisfactory results that also were quite insensitive to slight variations.

The adaption of the polynomial degree is calculated for each element independently, while the polynomial degree of the face is defined by the higher of the 2 polynomial degrees of the elements sharing this face.

3 | NUMERICAL EXPERIMENTS

3.1 | Academic problem setup

In our test, we use a modified version of the problem setup from Niederer et al.[3] The geometry corresponds to a myocardial tissue cuboid of size 12 × 4 × 2mm, see Figure 1A. To start the propagation of the electrochemical wave, we use the initial condition

$$u(\mathbf{x},0) = 0.5(1 - \tanh(1000(x-2))), \tag{16}$$

shown in Figure 1B.

Given that our study focuses on the investigation of different spatial approximations of the problem, we restrict the cardiac electrophysiology problem to the model of Bueno-Orovio et al,[31] which is able to reproduce important physiological properties, eg, action potential curves and upstroke velocities, with only 3 gating variables and 3 ionic currents. The model is given as

$$\begin{aligned} \partial_t u &= \nabla \cdot (\sigma \nabla v) - I_{ion}(u,\mathbf{w}), \\ \partial_t w_1 &= (1 - H(u-\theta_{w_1}))(w_{1,\infty} - w_1)/\tau_{w_1}^- - H(u-\theta_{w_1})w_1/\tau_{w_1}^+, \\ \partial_t w_2 &= (1 - H(u-\theta_{w_2}))(w_{2,\infty} - w_2)/\tau_{w_2}^- - H(u-\theta_{w_2})w_2/\tau_{w_2}^+, \\ \partial_t w_3 &= ((1 + \tanh(k_{w_3}(u-u_{w_3})))/2 - w_3)/\tau_{w_3}, \end{aligned} \tag{17}$$

with $H(\cdot)$ the Heaviside function and with the computation of the ionic current I_{ion} as

$$\begin{aligned} I_{ion}(u,\mathbf{w}) &= I_{fi} + I_{so} + I_{si}, \\ I_{fi} &= -w_1 H(u-\theta_{w_1})(u-\theta_{w_1})(u_u - u)/\tau_{fi}, \\ I_{so} &= (u-u_0)(1 - H(u-\theta_{w_2}))/\tau_0 + H(u-\theta_{w_2})\tau_{so}, \\ I_{si} &= -H(u-\theta_{w_2})w_2 w_3/\tau_{si}, \end{aligned} \tag{18}$$

and with the definitions of the functions

$$\begin{aligned} \tau_v^- &= (1 - H(u-\theta_v^-))\tau_{v_1}^- + H(u-\theta_v^-)\tau_{v_2}^-, \\ \tau_w^- &= \tau_{w_1}^- + (\tau_{w_2}^- - \tau_{w_1}^-)(1 - \tanh(k_w^-(u-u_w^-)))/2, \\ \tau_{so} &= \tau_{so1} + (\tau_{so2} - \tau_{so1})(1 - \tanh(k_{so}(u-u_{so})))/2, \\ \tau_s &= (1 - H(u-\theta_w))\tau_{s_1} + H(u-\theta_w)\tau_{s2}, \\ \tau_s &= (1 - H(u-\theta_o))\tau_{o_1} + H(u-\theta_o)\tau_{o2}, \\ w_\infty &= (1 - H(u-\theta_o))(1 - u/\tau_{w_\infty} + H(u-\theta_o)w_\infty^*, \\ v_\infty &= \begin{cases} 1 & u < \theta_v^- \\ 0 & u \geqslant \theta_v^- \end{cases} . \end{aligned}$$

The parameter values, which can be adapted to model different cell types, can be found in the original work.[31] In the present study, we choose the epicardial parameter values (see Table 1). If not indicated otherwise, the diffusion coefficient is chosen as $\sigma = 0.1\ \text{mm}^2/\text{ms}$. The time step is chosen as $\Delta t = 0.1$ ms, and different mesh sizes are used as indicated for each experiment.

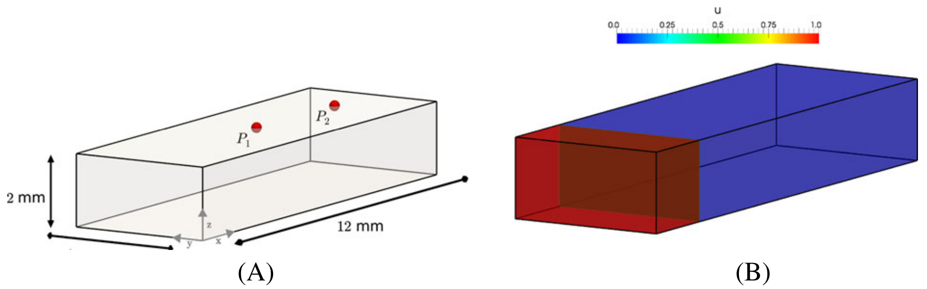
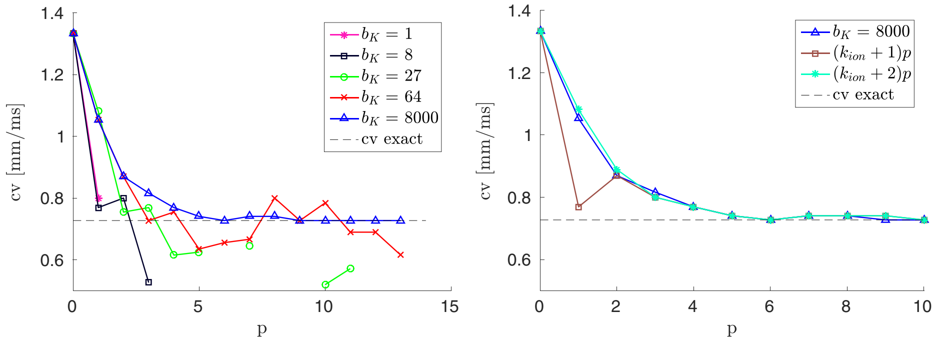


FIGURE 1 Myocardial tissue slab. A, Rectangular domain (12 × 4 × 2 mm). The time difference in activation is measured between the 2 red points P_1 and P_2 ($x_1 = 6$ mm, $y_1 = 2$ mm, $z_1 = 2$ mm, and $x_2 = 10$ mm, $y_2 = 2$ mm, $z_2 = 2$ mm). B, Initial condition

TABLE 1 Epicardial model parameter values

u_o	u_u	θ_v	θ_w	θ_v^-	θ_o	τ_{v1}^-
0	1.55	0.3	0.13	0.006	0.006	60
τ_{v2}^-	τ_v^+	τ_{w1}^-	τ_{w1}^-	K_w^-	u_w-	τ_w^+
1150	1.4506	60	15	65	0.03	200
τ_{fi}	τ_{o1}	τ_{o2}	τ_{so1}	τ_{so2}	k_{so}	u_{so}
0.11	400	6	30.0181	0.9957	2.0458	0.65
τ_{s1}	τ_{s2}	k_s	u_s	τ_{si}	τ_{w_∞}	w_∞^*
2.7342	16	2.0994	0.9087	1.8875	0.07	0.94



(A) (B)

FIGURE 2 Conduction velocity plotted over the polynomial degree p for varying number of integration points b_K. A, Comparison between different constant numbers of integration points. B, Comparison between adapted number of integration points and a large constant number

To compare the results, we calculate the conduction velocity cv in all simulations by measuring the activation times (acttime(·)) of point $P_1(6, 2, 2)$ and point $P_2(10, 2, 2)$ (see Figure 1A) and with the distance between the 2 points to obtain the conduction velocity via $cv = \|P_2 - P_1\|/(\text{acttime}(P_2) - \text{acttime}(P_1))$. We define the activation time as the time when the voltage is larger than 1.7 mV, which is equivalent to the value of 1 for the dimensionless variable u in the chosen cellular model. The converged conduction velocity amounts to around 0.73 mm/ms.

3.2 | Choice of the ionic current integration formula

Here, we briefly describe the application of the rule defined in Section 2.5 to the ionic model.[31] We identify the maximal polynomial degree of u in the model as $k_{ion} = 2$ through I_{fi}. Therefore, we choose the number of quadrature points b_K such that polynomials of degree d

$$d = (k_{ion} + 1)p = 3p \tag{19}$$

are exactly integrated.

We need a definition of an integration rule for the ionic current term that varies with the polynomial order. This necessity is visualized in Figure 2 for $\sigma = 0.1\ \text{mm/ms}^{-2}$ and $\tau = 1\ \text{mm/ms}^{-1}$ and $h = 1$ mm. We observe, in general, 2 types of issues arising when a fixed number of integration points is chosen. The first issue is a propagation block, which means that no propagation takes place. This could happen if the ionic flow from the neighboring element is not sufficient to reach the threshold for the self depolarization in an element. It occurs in our examples when the polynomial degree is increased above a certain value, see, eg, the case $b_K = 1, 8, 27$ in Figure 2A. For one integration point ($b_K = 1$), only at polynomial degrees zero and one the propagation takes place. For $b_K = 8$, propagation takes place until degree 3, and for $b_K = 27$, no propagation occurs for degrees 6, 8, 9, 12, and 13. The second issue is observed with $b_K = 27$ where the interaction between different computational errors, for example, the insufficient integration of the ionic current term and coarse meshes, could lead to a propagation for some polynomial degrees (6, 10, and 11) but with wrong results. This is the second issue. In the above mentioned cases for $b_K = 27$ and every time for $b_K = 64$ for high polynomial degrees, a propagation takes place but the computed conduction velocity is wrong, since the error in the computation of the ionic current dominates. This can be seen in the oscillating behavior of the computed conduction velocity for increasing polynomial degree (see Figure 2A).

Note that considerably increasing the number of integration points (eg, $b_K = 8000$) leads to a smooth convergence curve with respect to an increase of the polynomial order. However, using too many integration points results in an increased computational cost without a significant gain in the accuracy of the results. Finally, the adaptation of b_K in dependence of the polynomial order as defined in Equation 19 is able to give convergence at a much lower cost (see Figure 2B). However, for polynomial degree 1 for the adapted integration points, the conduction velocity is lower than expected. This is due to the additional nonpolynomial terms in the model in the calculation of the ionic current term. For low order $p = 1$, we experimentally determined that using integration exact to degree 3 is not accurate enough. Thus, we tested a stricter rule for the integration points so that polynomials of degree d

$$d = (k_{ion} + 2)p = 4p \tag{20}$$

are exactly integrated. This stricter rule (Equation (20)) solves the problem with low polynomial degrees, but it has almost no influence for the computation with higher polynomial degrees (see Figure 2B). Hence, to reduce the computational cost for high polynomial degrees, we decided to use the rule from Equation 19 for high order polynomials, but for $p = 1$, we use the rule defined in Equation 20 so that in any case, at least polynomials of degree 4 are exactly integrated, ie,

$$d = \begin{cases} 3p\,, & p > 1 \\ 4p\,, & p \leq 1 \end{cases}. \tag{21}$$

3.3 | Choice of the stabilization parameter

In this section, we study the influence of the HDG stabilization parameter τ on the solution quality. First, from Equation 7, we can observe that a large stabilization parameter $\tau \to \infty$ forces the solution u_h to be continuous, therefore reproducing the CG solution. This is verified in Figure 3, where the solutions are almost identical and thus the curves are the same.

Figure 4 presents results for different electrical conductivities σ in a wide range of values of physiological interest.[2,32] Firstly, we can see that with increasing polynomial order p, the influence of the stabilization parameter decreases. In Figure 4A only $h = 0.5$ mm is plotted since propagation for $h = 2$ mm takes place only for larger polynomial degrees. In all plots in Figure 4, it is visible that for larger τ, the conduction velocity is overestimated, and for smaller τ, it is underestimated. A small τ could also lead to the case that no propagation takes place. This propagation block can be seen in the plots, when no value for the conduction velocity is plotted. Decreasing the element size also leads to a decrease of the influence of the stabilization parameter. Nevertheless, overestimation and underestimation are still visible for high and low stabilization parameters, respectively.

In Figure 5A,B, the results for a more detailed range of $\tau = 0.11$ mm/ms and $\tau = 110$ mm/ms are plotted, respectively. We conclude that a reasonable value for the stabilization parameter τ for a wide range of physiological diffusivities and

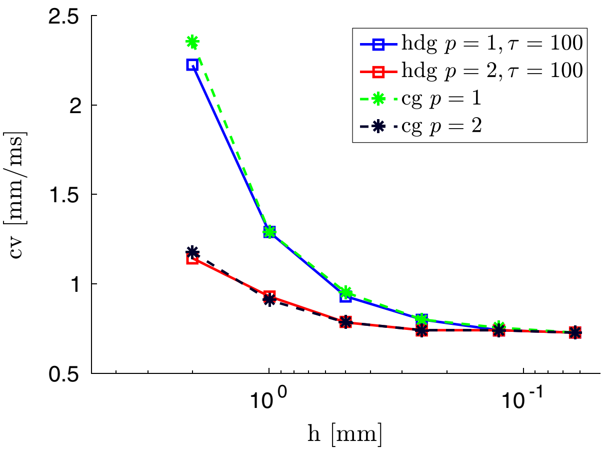
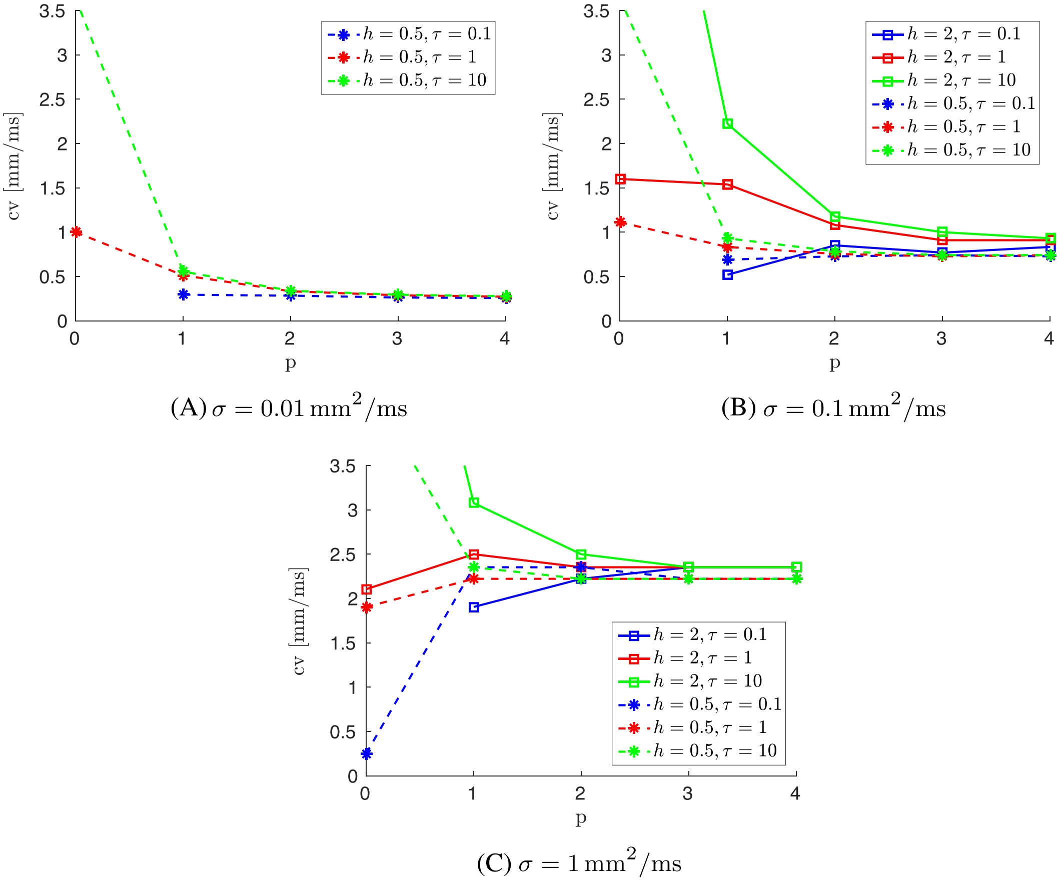


FIGURE 3 Conduction velocity (cv) plotted over the element size h for $\sigma = 0.1$ mm^2/ms. The HDG solution for large τ is the same as the CG solution

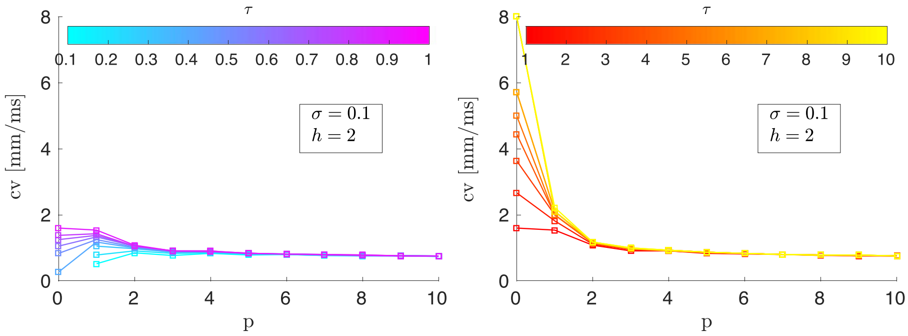


(A) $\sigma = 0.01\,\mathrm{mm}^2/\mathrm{ms}$

(B) $\sigma = 0.1\,\mathrm{mm}^2/\mathrm{ms}$

(C) $\sigma = 1\,\mathrm{mm}^2/\mathrm{ms}$

FIGURE 4 Conduction velocities for varying HDG stabilization parameters τ and diffusion coefficients σ



(A) τ =0.1 to 1 mm/ms

(B) τ =1 to 10 mm/ms

FIGURE 5 Detailed analysis of the influence of the stabilization parameter τ for $\sigma = 0.1$

element sizes is $\tau = 1$ mm/ms using the minimal cell model.[31] For other cell models, the behavior of the stabilization parameter can be easily tested in a similar fashion. Note that one should not use too low polynomial degrees to ensure only a small dependence on τ.

3.4 | Time discretization

For the sake of completeness, we briefly present the results of a time step refinement for a polynomial order of $p = 2$. In Figure 6, one can see the conduction velocity for different combinations of spatial and temporal refinement. For temporal refinement, we consider time steps of $\Delta t = 10^{-1}, 10^{-2}, 10^{-3}$, and 10^{-4} ms and for spatial refinement element sizes of $h = 2$,1, 0.5, 0.25, and 0.125 mm. With a time step of $\Delta t = 0.1$ ms, the solution is not yet fully converged in terms of

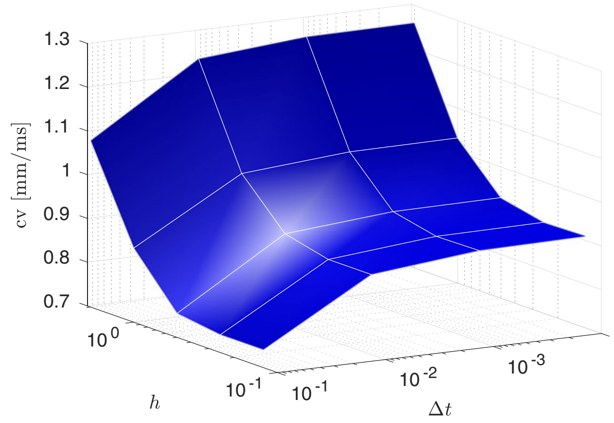


FIGURE 6 Conduction velocity plotted over the element size h and the time step Δt for a polynomial order of $p = 2$. The range from the element size is $h = 2, 1, 0.5, 0.25, 0.125$ mm and the time steps are $\Delta t = 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}$ ms

temporal errors. However, decreasing the time step does not change the convergence behavior of the general solution. Thus, analyzing different spatial discretization methods can be done on a coarser time step, without missing important characteristics of the solution. Figure 6 shows that with the semi-implicit time discretization, the conduction velocity is overestimated for a coarse mesh size, while for a large time step, it is underestimated.

3.5 | Convergence and efficiency analysis with *h* and *p* refinement

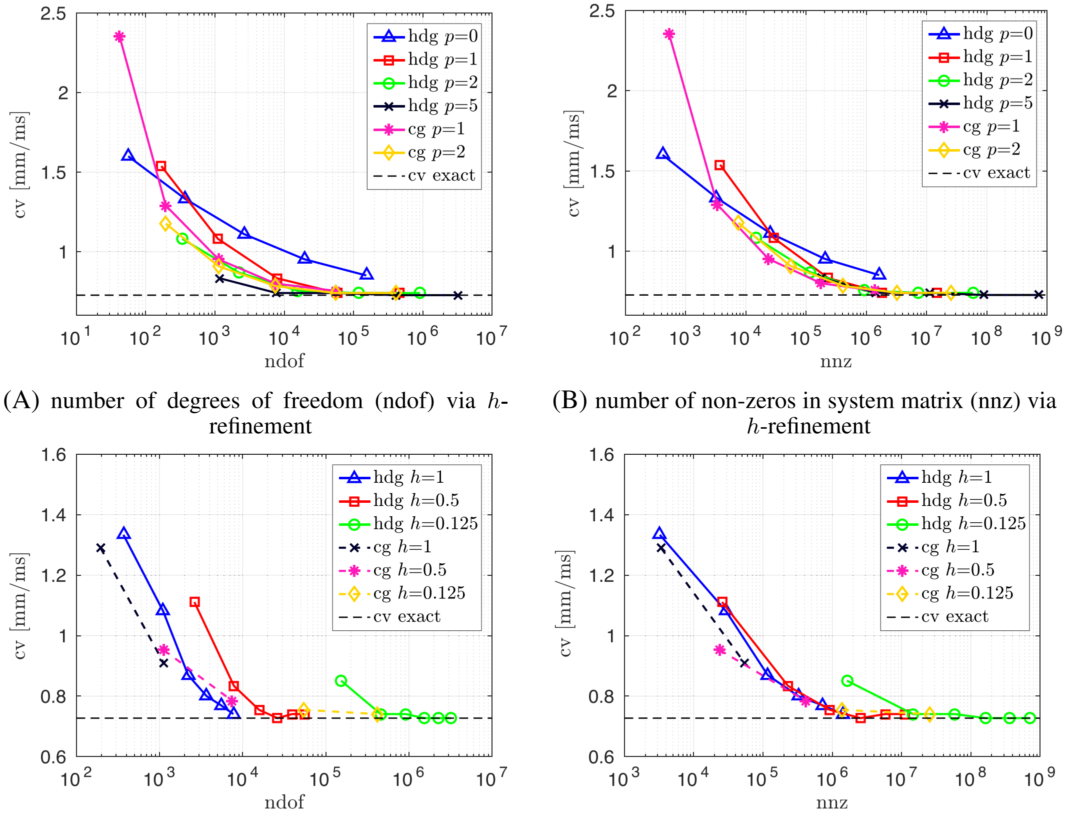
Next, we analyze the change in conduction velocity when decreasing the element size and increasing polynomial order for the HDG method for both tetrahedral and hexahedral meshes. Additionally, we compare the HDG discretization method with CG discretizations in more detail. For the HDG method, we use a stabilization parameter $\tau = 1$ mm/ms (see Section 3.3). For both methods, HDG and CG, the rule for the integration points for the ionic current is adapted to the polynomial degree as defined in Sections 2.5 and 3.2.

To compare the results, we record the number of degrees of freedom (ndof) and the number of nonzeros in the system matrix (nnz), since they can be seen as the key factors for the calculation time.[26] For the CG method, we consider these quantities directly from the discrete system for the potential u_h. In HDG, we do it for the condensed system involving only the trace variables Λ_h. We want to point out that for low order, the linear systems for HDG have more unknowns than for CG, while for high order, HDG has less unknowns than CG, ie, the unknowns for HDG increases more slowly with increasing order.[26]

First, we compare the HDG and CG methods for hexahedral meshes. Both methods appear to deliver similar results in terms of precision versus ndof and nnz (see Figure 7). It is also evident that both ways, ie, to increase the degree or to reduce the mesh size, are valid to better approximate the conduction velocity.

Additionally, our results suggest that to achieve a converged conduction velocity for HDG and CG and independent of the mesh type, approximately the same number of degrees of freedom and nonzero entries in the system matrix are needed, except for a polynomial order $p = 0$ in HDG. For the ndof, this can be seen in Figure 7A where the polynomial order is kept fixed and the element size is decreasing, and more clearly in Figure 7C, where the element size is kept fixed and polynomial order is increasing, and similarly for the nnz in Figure 7B,D.

For the same number of degrees of freedom, if the result is approximated with high-order elements instead of small sized lower-order elements, the obtained conduction velocity is closer to the exact solution (see Figure 8A). In terms of number of nnz, increasing the polynomial order or decreasing the element size leads to the same result (see Figure 8B). Thus, the HDG method becomes increasingly competitive as the polynomial degree is increased. For $h = 0.5$, the conduction velocity increases slightly at one point, although the conduction velocity should decrease because of the general convergence behavior. The slight increase could happen since we measure only a scalar value to look for the convergence of the solution. A coarse time step, a coarse mesh, and the averaging of the values around the measuring point could lead to small rounding errors, which cause a slight change in the computed conduction velocity. However, the convergence behavior of the overall problem does not change.



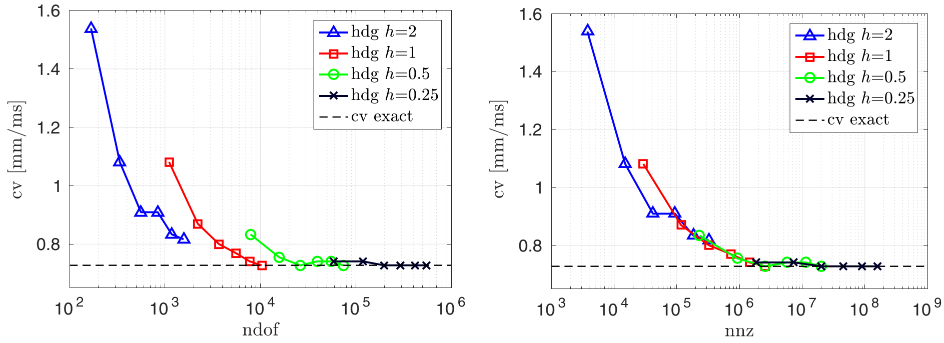
(A) number of degrees of freedom (ndof) via h-refinement

(B) number of non-zeros in system matrix (nnz) via h-refinement

(C) number of degrees of freedom (ndof) via p-refinement

(D) number of non-zeros in system matrix (nnz) via p-refinement

FIGURE 7 Conduction velocity plotted over ndof and nnz. (A) and (B) show the conduction velocity for the element sizes $h = 2, 1, 0.5, 0.25, 0.125$ mm. (C) and (D) show the conduction velocity for polynomial order $p = 0, 1, 2, 3, 4, 5$ for HDG and $p = 1$ and 2 for CG

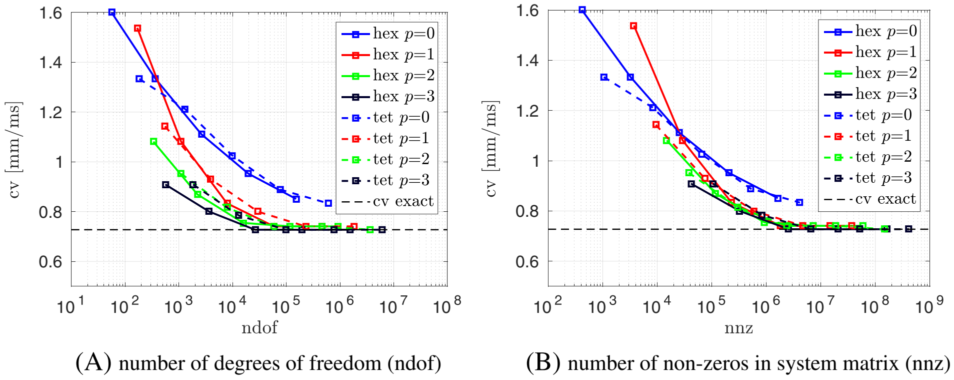


(A) number of degrees of freedom (ndof)

(B) number of non-zeros in system matrix (nnz)

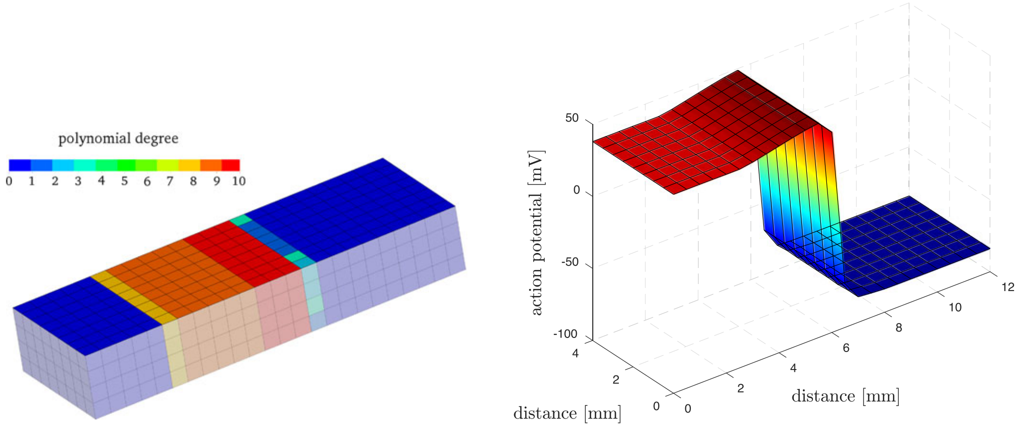
FIGURE 8 This is a closeup of the results for HDG from Figure 7C,D. The conduction velocity for polynomial order $p = 1, 2, 3, 4, 5, 6$ is plotted

In Figure 9, we compare tetrahedral with hexahedral meshes. Note that for a fixed h, tetrahedral meshes involve a larger number of elements than hexahedral meshes. Although for a fixed h in CG, both mesh discretizations, hexahedral and tetrahedral, deliver the same ndof in the structured mesh; in HDG (and particularly high order), the tetrahedral mesh involves a considerably larger number of degrees of freedom. However, we can clearly appreciate that we roughly require the same ndof and nnz with tetrahedral and hexahedral meshes for achieving similar precision, ie, the precision is only depending on the ndof and is independent of the element type. Note that for high-order tetrahedral elements, we use optimized nodal sets, the so-called warp and blend points, to avoid ill-conditioned interpolation and negative impact on linear solvers.[33]



(A) number of degrees of freedom (ndof) (B) number of non-zeros in system matrix (nnz)

FIGURE 9 Comparison between hexahedral (hex) and tetrahedral (tet) elements. The conduction velocity is plotted for element sizes of h = 2,1, 0.5, 0.25, 0.125 mm. The difference between the element types is relatively small



(A) polynomial degree distribution (B) action potential

FIGURE 10 Polynomial degree and action potential at time 6 ms. The element size is 0.5 mm and the tolerance was chosen as $e_{tol} = 10^{-4}$. A, Distribution of the adaptively chosen polynomial degree. B, Action potential of the upper surface. At the steep electrochemical wave front, the polynomial degree is high

3.6 | Results for p-adaptivity

The previous results show that increasing the polynomial order in HDG is a very efficient way to improve accuracy. To take advantage of the latter fact and move towards more efficient codes for computing wave propagation, as it has been postulated in previous works, it is natural to think about increasing the resolution at the wave front only. The option of doing this in a very convenient way is our main motivation for using HDG. Therefore, we now show the results of HDG using a p-adaptive approach.

We start the simulation with element order of $p = 0$ everywhere, define a maximal polynomial order $p_{\max}$, and apply the adaptivity strategy specified in Section 2.6. Figure 10 shows the result of a p-adaptive simulation for element size $h = 0.5$ mm and $p_{\max} = 10$. As expected, the method selects a high polynomial degree at the position of the steep electrochemical wave front, and after some time, when the wave has passed, the polynomial degree decreases again.

We present the results for different values of h, $p_{\max}$, and e_{tol} in more detail in Table 2, by comparing the error between computed conduction velocity and the exact one. The table shows the error of the constant polynomial degree, $p = p_{\max}$ everywhere, and the adaptive polynomial degree until degree $p_{\max}$. Note that the adaptive simulations always utilize the highest polynomial degree. To see the influence of the error tolerance, we calculated the error indicator without normalizing for 3 different element sizes $h = 2, 1, 0.5$ mm, ie, we use $A_\gamma = 1$ in Equation 14. The tested error tolerances are $e_{tol} = 10^{-2}, 10^{-4}, 10^{-8}$ (see Table 2). From Table 2, we can see that the error decreases with increasing polynomial degree for the p-constant method a large tolerance. This is not the case for the p-adaptive method with an error tolerance of $e_{tol} = 10^{-2}$, ie, p-adaptivity with a large error tolerance is not able to reproduce the p-constant solution. Only for small element sizes and small polynomial degrees, the method is able to approximate the result of the p-constant method.

TABLE 2 Comparison of the relative error to the exact conduction velocity of standard HDG method with the p-adaptive method for different polynomial degrees and different error tolerances e_{tol}

		p-Const	**p-Adaptive**			
	e_{tol}		10^{-2}	10^{-4}	10^{-8}	10^{-4}
	$p_{\max}$		error= $\lvert$cv-cv$^{\text{exact}}\rvert$/cv$^{\text{exact}}$			
$h = 2$ mm	2	48.5 %	52.62 %	48.5 %	44.37 %	44.37 %
	4	25.12 %	62.25 %	22.37 %	25.12 %	34.75 %
	5	14.12 %	89.75 %	16.87 %	14.12 %	27.87 %
	7	10 %	103.5 %	22.37 %	10 %	10 %
	10	3.12 %	44.37 %	37.5 %	3.12 %	3.12 %
$h = 1$ mm	2	19.62 %	19.62 %	19.62 %	16.87 %	19.62 %
	4	5.87 %	1.75 %	5.87 %	5.87 %	5.87 %
	5	1.75 %	2.38 %	1.75 %	3.12 %	1.75 %
	7	1.75 %	7.88 %	0.37 %	1.75 %	0.37 %
	10	0.37 %	7.88 %	0.37 %	0.37 %	0.37 %
$h = 0.5$ mm	2	3.12 %	2.38 %	3.12 %	3.12 %	3.12 %
	4	1.75 %	1.75 %	0.37 %	0.37 %	0.37 %
	5	1.75 %	14.12 %	0.37 %	0.37 %	0.37 %
	7	0.37 %	41.62 %	0.37 %	0.37 %	0.37 %
	10	0.37 %	62.25 %	2.38 %	0.37 %	0.37 %

Note: $p_{\max}$ is the maximal polynomial order allowed, h is the element size and cv is the conduction velocity

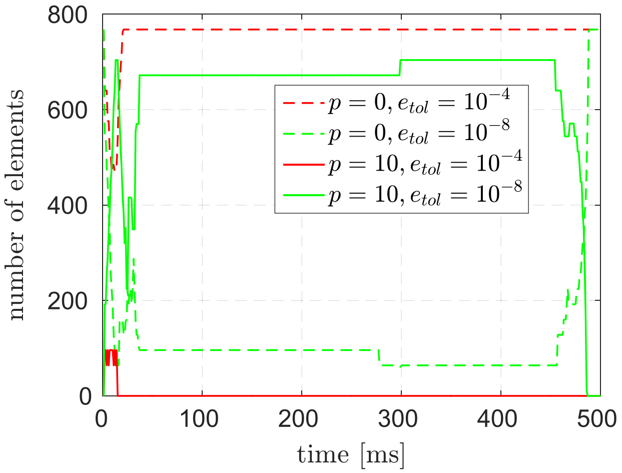
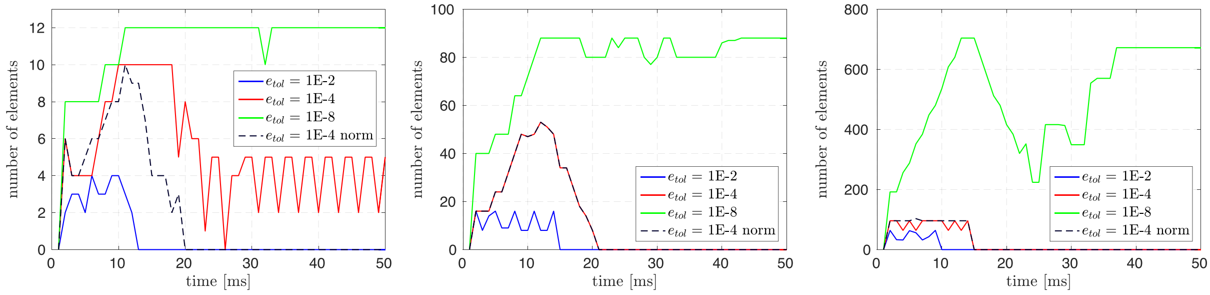


FIGURE 11 Number of elements with polynomial order of $p = 0$ and $p = 10$ during a p-adaptive simulation plotted over time. The element size is $h = 0.5$ mm

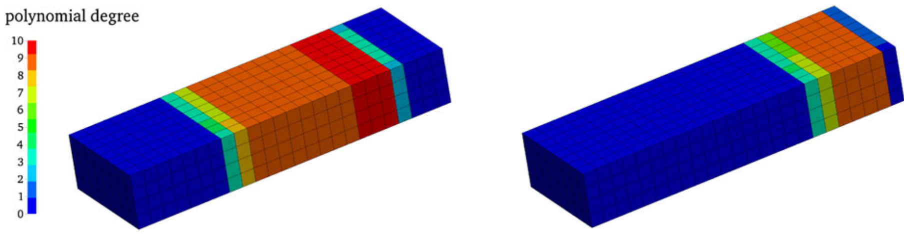
Decreasing the error tolerance leads to a better approximation of the p-constant result as it can be seen in Table 2 for $e_{tol} = 10^{-4}$ and $e_{tol} = 10^{-8}$. However, a smaller error tolerance increases the number of high-order elements (Figure 11). For a coarse tolerance (eg, $e_{tol} = 10^{-2}$), the number of elements with high order over time is small and it also decreases rapidly again. Decreasing the tolerance increases the number of high-order elements, and over time, the decrease of the number of high-order element is slower (Figure 11). A further decrease of the tolerance $e_{tol} = 10^{-8}$ results in a large number of high-order elements without returning to low order once the wave front has passed. The elements remain at high order until the action potential is returned to resting potential (Figure 11). This results almost in a constant polynomial degree over time. Thus, the smaller the error tolerance, the more the constant method is approached.

Comparing different error tolerances for different element sizes shows large differences if the error indicator is not normalized (Figure 12). For an error tolerance of $e_{tol} = 10^{-4}$, the element order does not decrease in the simulation with



(A) $h = 2\,\mathrm{mm}$ (B) $h = 1\,\mathrm{mm}$ (C) $h = 0.5\,\mathrm{mm}$

FIGURE 12 Number of elements with polynomial order of $p = 10$ during a p-adaptive simulation plotted over time for different element sizes $h = 2, 1, 0.5$ mm until a time of 50 ms



(A) time 10 ms (B) time 17 ms

FIGURE 13 Polynomial degree distribution for 2 different time steps for a mesh with element size of 0.5 mm and a normalized error indicator

element size $h = 2\,\mathrm{mm}$. Using a normalized error indicator calculation as stated in Equation 14 and an error tolerance of $e_{tol} = 10^{-4}$, we can see that in all cases, the polynomial degrees return to low order (Figure 12) and also the difference between constant and adaptive method is small (Table 2). Figure 13 shows the distribution of the polynomial degrees at different times for a calculation with a normalized error indicator for a mesh with $h = 0.5\,\mathrm{mm}$ element size. Slight differences in the polynomial order along the width of the geometry are due rounding errors. This asymmetry in the polynomial order distribution arises only after many time steps due to accumulated effects of an integer decision (which degree) from a continuous field that is almost the same (subject to roundoff). The polynomial degree increases when the electrochemical wave arrives and after around 10 ms the polynomial degree decreases again. Thus, our results verify that the error indicator calculation according to Equation 14 is a good option for localizing high degrees to the elements close to the wave front.

3.7 | Computations on a real biventricular geometry

To show the applicability of our method to real cardiac electrophysiology simulations, we use our p-adaptive HDG implementation to solve the electrophysiological propagation problem for a real human ventricle geometry. For this purpose, we segmented the left and right ventricle from magnetic resonance images acquired in a healthy 33-year-old female volunteer, with a dual-phase whole-heart 3D b-SSFP sequence[34] on a 1.5T Philips Achieva MRI scanner, acquisition matrix $212 \times 209 \times 200$, acquired voxel size $2 \times 2 \times 2$ mm, repetition time 4.5 ms, echo time 2.2 ms, echo train length 26, and flip angle 90^o. The diastasis was used to generate the computational mesh.

To investigate different levels of mesh refinement, we use tetrahedral elements with a maximal element size of 2, 1, and 0.5 mm, which results in 3 meshes with 59 801, 393 302, and 2 904 351 number of elements, respectively. On real heart geometries, mesh refinements inevitably lead to slight changes of the geometry. Hence, we examine whether the difference in the geometry strongly influences the results, or coarser meshes with high-order elements are able to reproduce the activation propagation accurately.

We initiate the propagation of the electrical wave with a stimulus current on the apex. The maximal polynomial degree is set to 5 and at the beginning all polynomial degrees are set to zero. For computational reasons, we use the

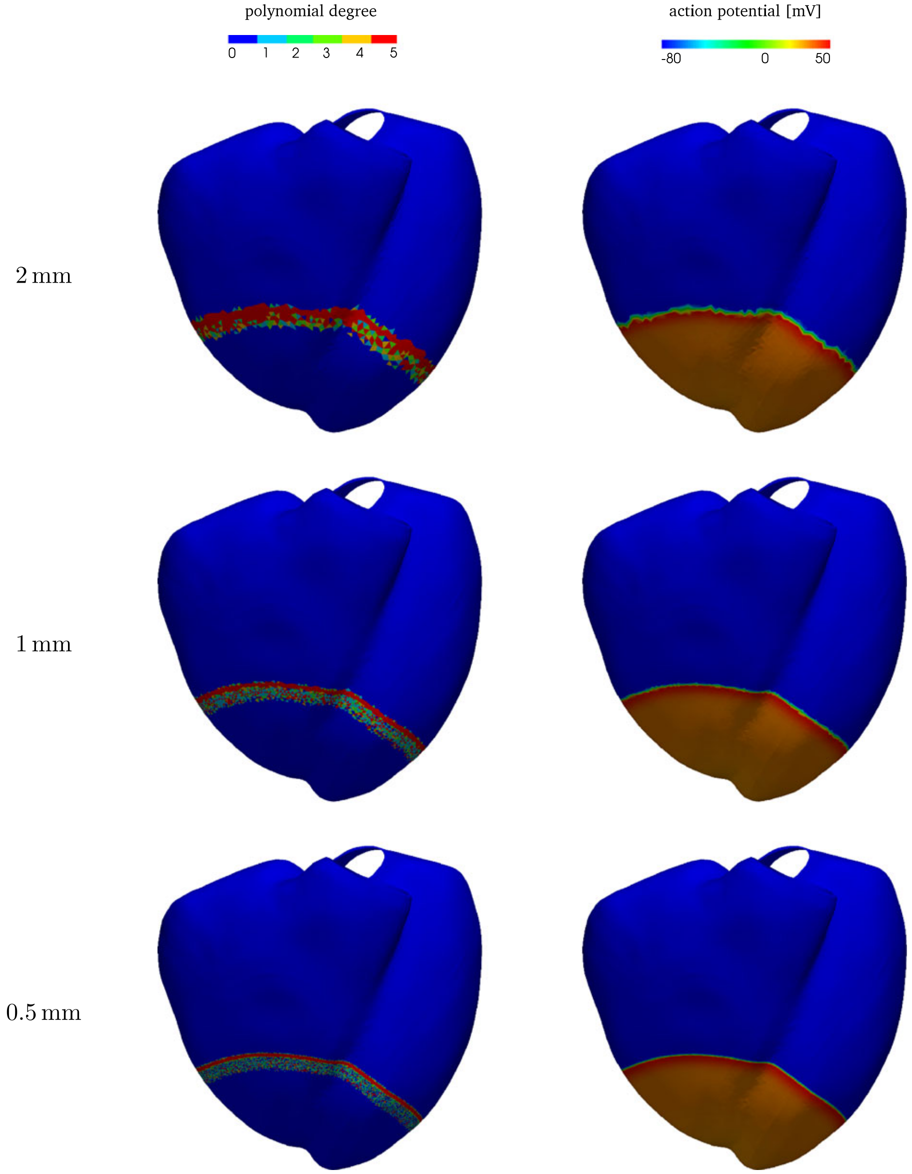


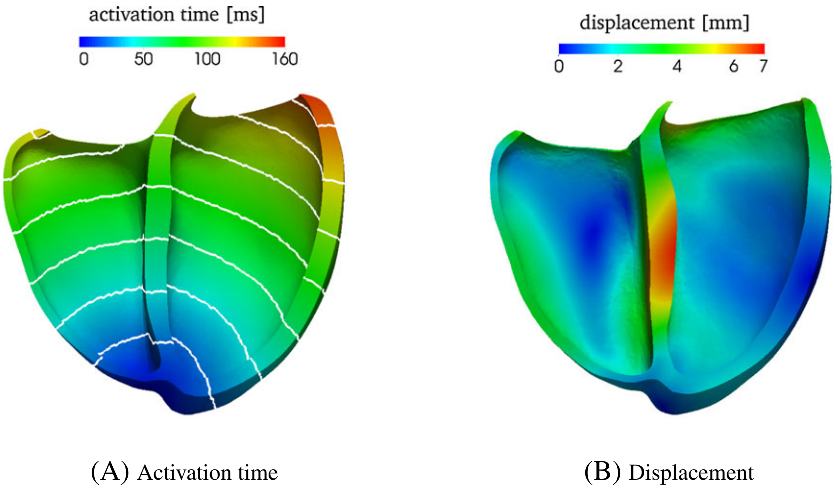
FIGURE 14 Polynomial degree and action potential at time 50 ms

integration rule defined in Equation 19 in this example for all polynomial degrees. The coarsest mesh shows a slightly faster activation than the finer ones, while only very small differences in the activation are visible between the 2 finer meshes (see Figure 14). The activation time of a sample point in the middle of the posterior wall at the junction between left and right ventricle has been captured for the different levels of refinement. The activation time of the sample point for the coarsest mesh is measured as about 40.7 ms, for the finer one about 45.4 ms, and for the finest mesh about 46.3 ms.

When using nonadaptive HDG elements for the example with a maximal element size of 1 mm one would have to use 17 739 225 ndof during all time steps. In comparison, with the p-adaptive HDG method, we have between 844 725 minimal and 2 275 660 maximal ndof. The ndof averaged over the whole simulation depends on a particular case, of course, ie, the geometry, the simulation time, and the activation sequence influence the ndof over time. In our case, for an element size of 1 mm, we show the average ndof until, eg, the end of the activation or the end of the heart cycle. The averaged ndof can be seen in Table 3. It decreases when simulating a larger part of the cardiac cycle, since the high polynomial degrees are used only at around the wave front. The decrease of the ndof is very useful, in particular when the calculated activation is coupled to mechanical simulation. In this case the activation takes place only in a short period of time, while the overall simulation is much longer.

TABLE 3 Ndof averaged from start until given time, for the example with 1 mm maximal element size

Specified simulation end	Time	Averaged ndof over time
End of activation	150 ms	1 709 905
End of systole	300 ms	1 278 179
Diastasis	500 ms	1 107 227
End of heart cycle	1000 ms	988 151



(A) Activation time

(B) Displacement

FIGURE 15 Activation time with isochrones at each 20 ms and the displacement at peak systole at about 300 ms from the activation onset for a real heart geometry with an element size of approximately 1 mm

The electrophysiological model can also be coupled to a mechanical model to calculate electromechanics in the heart. The mechanical simulation is solved using continuous finite elements, while for the electrophysiological simulation, we use p-adaptive HDG discretization, but both meshes are the same. We couple the mechanical simulation through the action potential, which is stored for each element at the integration points. For more details about the coupling and the mechanical simulation, we refer to our work described in Hörmann et al.[35] In Figure 15, the activation time and the displacement at peak systole are shown for the ventricular geometry with a maximal element size of 1 mm.

4 | CONCLUSION

In this paper, we proposed and analyzed the use of an adaptive high-order HDG method for efficiently calculating the electric propagation in human hearts. As compared to classical DG methods, an HDG discretization reduces the degrees of freedom through static condensation on the element level, so that only the degrees of freedom defined on the faces between the elements show up in the global system of equations. An advantage of the HDG method (as well as for other DG methods) is the simple usage of spatially varying high-order elements, due to the discontinuity between the elements that is captured through numerical fluxes. An approach with high-order elements in turn additionally needs a good approximation of the integral of the ionic current term. In this paper, we have defined a rule for a suitable integration accuracy depending on the polynomial order and the cell model selected by the requirement to exactly integrate the leading current term. Furthermore, we have defined and explained a practical choice of the stabilization parameter for the HDG discretization for the electrophysiological problem. Comparing CG and HDG methods, we have seen a similar performance for low-order elements, but an increase in efficiency for high-order elements.

The electrophysiology calculation is determined by a steep electrochemical wave front that travels through cardiac tissue. We proposed a simplification of an error indicator previously reported to localize it and to define an error indicator, which can be used to suggest the appropriate order of the element and in this way realize a p-adaptive HDG approach for cardiac electrophysiology. In summary, HDG shows great potential to efficiently solve large-scale electrophysiological problems for complex geometries using coarse meshes and p-adaptive high-order elements, which can be additionally coupled with cardiac mechanical simulations.

ACKNOWLEDGMENTS

The authors would like to thank Eva Sammut, King's College London, for her help with the image data used in this paper. Thanks also goes to Martina Weigl for her help in segmenting the biventricular geometry.

ORCID

Martin Kronbichler http://orcid.org/0000-0001-8406-835X

REFERENCES

1. Chabiniok R, Wang VY, Hadjicharalambous M, et al.. Multiphysics and multiscale modelling, data–model fusion and integration of organ physiology in the clinic: ventricular cardiac mechanics. *Interface Focus*. 2016;6(2):2015008:1-2015008:24. https://doi.org/10.1098/rsfs.2015.0083
2. Qu Z, Hu G, Garfinkel A, Weiss JN. Nonlinear and stochastic dynamics in the heart. *Phys Rep*. 2014;543(2):61-162. https://doi.org/10.1016/j.physrep.2014.05.002
3. Niederer SA, Kerfoot E, Benson AP, et al.. Verification of cardiac tissue electrophysiology simulators using an N-version benchmark. *Philos Trans Royal Soc A Math Phys Eng Sci*. 2011;369(1954):4331-4351.
4. Fischer G, Tilg B, Modre R, et al.. A bidomain model based BEM-FEM coupling formulation for anisotropic cardiac tissue. *Ann Biomed Eng*. 2000;28(10):1229-1243.
5. Vigmond EJ, Aguel F, Trayanova NA. Computational techniques for solving the bidomain equations in three dimensions. *IEEE Trans Biomed Eng*. 2002;49(11):1260-1269.
6. Colli Franzone P, Pavarino LF. A parallel solver for reaction–diffusion systems in computational electrocardiology. *Math Models Methods Appl Sci*. 2004;14(06):883-911.
7. Göktepe S, Kuhl E. Computational modeling of cardiac electrophysiology: a novel finite element approach. *Int J Numer Meth Eng*. 2009;79(2):156-178.
8. Linge S, Sundnes J, Hanslien M, Lines GT, Tveito A. Numerical solution of the bidomain equations. *Philos Trans Royal Soc A Math Phys Eng Sci*. 2009;367(1895):1931-1950.
9. Pathmanathan P, Mirams GR, Southern J, Whiteley JP. The significant effect of the choice of ionic current integration method in cardiac electro-physiological simulations. *Int J Numer Method Biomed Eng*. 2011;27(11):1751-1770.
10. Krishnamoorthi S, Sarkar M, Klug WS. Numerical quadrature and operator splitting in finite element methods for cardiac electrophysiology. *Int J Numer Method Biomed Eng*. 2013;29(11):1243-1266.
11. Pezzuto S, Hake J, Sundnes J. Space-discretization error analysis and stabilization schemes for conduction velocity in cardiac electrophysiology. *Int J Numer Method Biomed Eng*. 2016;32(10):e02762, https://doi.org/10.1002/cnm.2762.
12. Patelli AS, Dedè L, Lassila T, Bartezzaghi A, Quarteroni A. Isogeometric approximation of cardiac electrophysiology models on surfaces: an accuracy study with application to the human left atrium. *Comput Methods in Appl Mech Eng*. 2017;317:248-273.
13. Lines GT, Grottum P, Tveito A. Modeling the electrical activity of the heart: a bidomain model of the ventricles embedded in a torso. *Comput Visual Sci*. 2002;5(4):195-213.
14. Trangenstein JA, Kim C. Operator splitting and adaptive mesh refinement for the Luo–Rudy I model. *J Comput Phys*. 2004;196(2):645-679.
15. Belhamadia Y. A time-dependent adaptive remeshing for electrical waves of the heart. *IEEE Trans Biomed Eng*. 2008;55(2):443-452.
16. Quan W, Evans SJ, Hastings HM. Efficient integration of a realistic two-dimensional cardiac tissue model by domain decomposition. *IEEE Trans Biomed Eng*. 1998;45(3):372-385.
17. Heidenreich EA, Rodríguez JF, Gaspar FJ, Doblaré M. Fourth-order compact schemes with adaptive time step for monodomain reaction–diffusion equations. *J Comput Appl Math*. 2008;216(1):39-55.
18. Cherry EM, Greenside HS, Henriquez CS. A space-time adaptive method for simulating complex cardiac dynamics. *Phys Rev Lett*. 2000;84(6):1343-1346. https://doi.org/10.1103/PhysRevLett.84.1343
19. Cherry EM, Greenside HS, Henriquez CS. Efficient simulation of three-dimensional anisotropic cardiac tissue using an adaptive mesh refinement method. *Chaos An Interdisciplin J Nonlinear Sci*. 2003;13(3):853-865.
20. Franzone PC, Deuflhard P, Erdmann B, Lang J, Pavarino LF. Adaptivity in space and time for reaction-diffusion systems in electrocardiology. *SIAM J Sci Comput*. 2006;28(3):942-962.
21. Whiteley JP. Physiology driven adaptivity for the numerical solution of the bidomain equations. *Ann Biomed Eng*. 2007;35(9):1510-1520.
22. Deuflhard P, Erdmann B, Roitzsch R, Lines GT. Adaptive finite element simulation of ventricular fibrillation dynamics. *Comput Visual Sci*. 2009;12(5):201-205.
23. Arthurs CJ, Bishop MJ, Kay D. Efficient simulation of cardiac electrical propagation using high-order finite elements II: adaptive p-version. *J Comput Phys*. 2013;253:443-470.

24. Arthurs CJ, Bishop MJ, Kay D. Efficient simulation of cardiac electrical propagation using high order finite elements. *J Comput Phys*. 2012;231(10):3946-3962.

25. Sevilla R, Hassan O, Morgan K. An analysis of the performance of a high-order stabilised finite element method for simulating compressible flows. *Comput Meth Appl Mech Eng*. 2013;253:15-27.

26. Giorgiani G, Fernández-Méndez S, Huerta A. Hybridizable discontinuous Galerkin p-adaptivity for wave propagation problems. *Int J Numer Methods Fluids*. 2013;72(12):1244-1262.

27. Cockburn B, Gopalakrishnan J, Lazarov R. Unified hybridization of discontinuous Galerkin, mixed, and continuous Galerkin methods for second order elliptic problems. *SIAM J Numer Anal*. 2009;47(2):1319-1365.

28. Whiteley JP. An efficient numerical technique for the solution of the monodomain and bidomain equations. *IEEE Trans Biomed Eng*. 2006;53(11):2139-2147.

29. Fernández MA, Zemzemi N. Decoupled time-marching schemes in computational cardiac electrophysiology and ECG numerical simulation. *Math Biosci*. 2010;226(1):58-75.

30. Nguyen NC, Peraire J, Cockburn B. An implicit high-order hybridizable discontinuous Galerkin method for linear convection–diffusion equations. *J Comput Phys*. 2009;228(9):3232-3254.

31. Bueno-Orovio A, Cherry EM, Fenton FH. Minimal model for human ventricular action potentials in tissue. *J Theor Biol*. 2008;253(3):544-560.

32. Kléber AG, Rudy Y. Basic mechanisms of cardiac impulse propagation and associated arrhythmias. *Physiol Rev*. 2004;84(2):431-488.

33. Hesthaven JS, Warburton T. *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*, Texts in applied mathematics. New York: Springer; 2008.

34. Uribe S, Tangchaoren T, Parish V, et al. Volumetric cardiac quantification by using 3d dual-phase whole-heart MR imaging. *Radiology*. 2008;248(2):606-614.

35. Hörmann JM, Bertoglio C, Nagler A, et al. Multiphysics modeling of the atrial systole under standard ablation strategies. *Cardiovasc Eng Technol*. 2017;8(2):205-218.

How to cite this article: Hoermann JM, Bertoglio C, Kronbichler M, Pfaller MR, Chabiniok R, Wall WA. An adaptive hybridizable discontinuous Galerkin approach for cardiac electrophysiology. *Int J Numer Meth Biomed Engng*. 2018;34:e2959. https://doi.org/10.1002/cnm.2959