Space-time multilevel quadrature methods and their application for cardiac electrophysiology

Seif Ben Bader, Helmut Harbrecht, Rolf Krause, Michael Multerer, Alessio Quaglino, Marc Schmidlin

Departement Mathematik und Informatik
Fachbereich Mathematik
Universität Basel
CH-4051 Basel

Preprint No. 2021-13
May 2021

dmi.unibas.ch

Space-time multilevel quadrature methods and their application for cardiac electrophysiology

S. Ben Bader[1], H. Harbrecht[2], R. Krause[1], M. Multerer[1], A. Quaglino[1,3], and M. Schmidlin[2]

[1]Center for Computational Medicine in Cardiology,
Institute of Computational Science,
Università della Svizzera italiana, Lugano, Switzerland
[2]Department of Mathematics and Computer Science,
University of Basel, Basel, Switzerland
[3]NNAISENSE SA, Lugano, Switzerland

Last update: May 4, 2021

Abstract

We present a novel approach which aims at high-performance uncertainty quantification for cardiac electrophysiology simulations. Employing the monodomain equation to model the transmembrane potential inside the cardiac cells, we evaluate the effect of spatially correlated perturbations of the heart fibers on the statistics of the resulting quantities of interest. Our methodology relies on a close integration of multilevel quadrature methods, parallel iterative solvers and space-time finite element discretizations, allowing for a fully parallelized framework in space, time and stochastics. Extensive numerical studies are presented to evaluate convergence rates and to compare the performance of classical Monte Carlo methods such as standard Monte Carlo (MC) and quasi-Monte Carlo (QMC), as well as multilevel strategies, i.e. multilevel Monte Carlo (MLMC) and multilevel quasi-Monte Carlo (MLQMC) on hierarchies of nested meshes. We especially also employ a recently suggested variant of the multilevel approach for non-nested meshes to deal with a realistic heart geometry.

1 Introduction

The heart is by all means one of the most complex and fascinating organs in the human body. It harmoniously orchestrates the body activity through the vital supply of blood to all of its components. This activity is achieved via its pumping function that is a result of a very complex contraction and relaxation cycle occurring in the cardiac cells. The latter is itself controlled by a non-trivial pattern of electrical activation.

A misregulation of the electrical activity of the heart can result in several diseases, having in worst cases lethal consequences. Therefore, it is of major importance to model and understand the heart activity, as this would allow for a better clinical diagnosis and treatment of patients.

The heart consists of fibers, compare Figure 1, that help propagate the electrical potential inside the cardiac muscle. This process is originally initiated by a stimulus coming from the sinoatrial node (SA) located on top of the left and right atria. From there, the signal spreads all over the heart muscle in the form of a traveling wave front. The propagation also takes place in the heart cells that have the ability to actively respond to the electrical stimulation through voltage-gated ion channels.

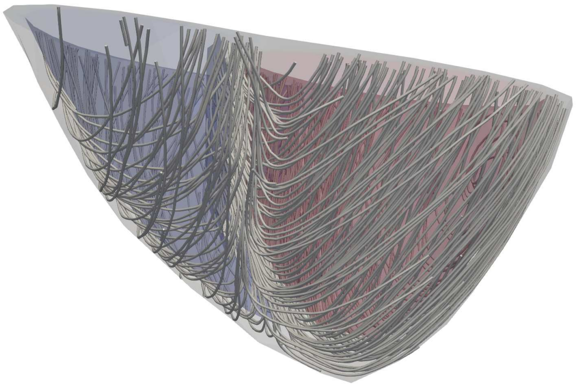


Figure 1: Mathematical reconstruction of the fiber field surrounding a synthetic heart geometry.

A mathematical model for describing the potential inside the cardiac muscle therefore has to provide a suitable ionic channel model in order to fully capture the phenomenological behaviour. Combined with time–dependency, spatial diffusion and a forcing function modelling the initial stimulus, one arrives at the monodomain equation, which has been derived in [24, 27]. It can be written in the following form:

$$
\begin{aligned}
\frac{\partial u(\boldsymbol{x},t)}{\partial t} - \nabla \cdot \big(\boldsymbol{G}(\boldsymbol{x})\nabla u(\boldsymbol{x},t)\big) + I_{\text{ion}}\big(u(\boldsymbol{x},t)\big) = I_{\text{app}}(\boldsymbol{x},t), &\quad \text{for } (\boldsymbol{x},t) \in D \times (0,T], \\
\boldsymbol{G}(\boldsymbol{x})\nabla u(\boldsymbol{x},t) \cdot n = 0, &\quad \text{for } (\boldsymbol{x},t) \in \partial D \times (0,T], \\
u(\boldsymbol{x},0) = 0, &\quad \text{for } \boldsymbol{x} \in D.
\end{aligned} \tag{1}
$$

Here, $D \subset \mathbb{R}^d$ is the domain representing the heart, $u = u(\boldsymbol{x},t)$ is the electrical potential, $\boldsymbol{G}\colon D \to \mathbb{R}^{d\times d}$ is an anisotropic conductivity tensor modeling the fiber direction, $T \in \mathbb{R}^+$ is the end time, $I_{\text{app}}\colon D \times [0,T] \to \mathbb{R}$ is the forcing function for the stimulus created by the SA node, and $I_{\text{ion}}\colon \mathbb{R} \to \mathbb{R}$ is an ion channel model. The latter can be modeled in several ways accounting for different levels of detail and complexity, see [23]. We rely here on the Fitz-Hugh Nagumo model, see [8], for which we have:

$$
I_{\text{ion}}(u) = \alpha(u - u_{\text{rest}})(u - u_{\text{th}})(u - u_{\text{peak}}), \quad \alpha > 0. \tag{2}
$$

The values u_{rest}, u_{th} and u_{peak} are characteristic potential values of the electrical activation process. They respectively represent the resting potential u_{rest} (cell is unactivated), the threshold potential u_{th} (cell is triggered) and the peak value u_{peak} (cell is activated).

As depicted in Figure 1, the fibers have a very complex but also well-organized structure, exhibiting key features that can be identified in all healthy subjects, such as a helical distribution with opposite orientations, from the endocardium to the epicardium. However, the exact fiber dislocations can vary not only from patient to patient, but can also change over time within the same patient due to pathologies, such as infarctions. Then, the fiber structure is perturbed with the introduction of high variability areas in the presence of scars. To faithfully model the conductivity tensors used in electrophysiology, accurate measurements of these fibers are required. However, such measurements cannot be made available on a routine basis. Given that a highly accurate model of fibers and thus the conductivities are generally unavailable except for a few test subjects, it is of paramount importance to gauge the influence that uncertainties in the conductivity tensor have on the simulated activation patterns. Modeling and simulating the uncertainty in the fiber directions will be one of the major aspects of this article.

We apply state of the art methods in uncertainty quantification (UQ). This means that we combine space-time GMRES with a block Jacobi preconditioner [5] for solving the monodomain equation with multilevel quadrature methods for the UQ. In our practical implementation, we use the multilevel (quasi-) Monte Carlo method, compare [1, 11, 14, 17, 20]. Therefore, additional smoothness of the solution is required as already pointed out in [17]. This smoothness has been verified in the stationary case for anistropic random diffusion problems in [15], for linear random advection-diffusion-reaction problems in [21], and for semilinear random diffusion problems in

[31]. Our numerical results in Section 5 show that the quasi-Monte Carlo method based on Halton points is superior over the Monte Carlo method and that the multilevel versions are superior over the single-level versions of these quadrature methods. Indeed, the highest efficiency is provided by the multilevel quasi-Monte Carlo method.

The rest of this article is organized as follows. In Section 2, we present the random model for the fibers of the heart muscle. Then, Section 3, is concerned with the space-time solver for the monodomain equation. Quadrature methods to treat the randomness are outlined in Section 4. Finally, in Section 5, numerical experiments are presented in order to validate the present approach also in case of simulations for realistic heart geometries.

2 Preliminaries

2.1 Random fiber directions

Within this article, we will consider an uncertainty on the random fiber directions. To this end, let $(\Omega, \mathcal{F}, \mathbb{P})$ denote a complete and separable probability space. Then, for a given Banach space $\mathcal{X}$ and $1 \leq p \leq \infty$, the space $L^p(\Omega;\mathcal{X})$ denotes the Lebesgue-Bochner space, see [22], which contains all equivalence classes of strongly measurable functions $v\colon \Omega \to \mathcal{X}$ with finite norm

$$\|v\|_{L^p} := \begin{cases} \left(\displaystyle\int_\Omega \|v(\omega)\|_{\mathcal{X}}^p \right)^{1/p} \mathrm{d}\mathbb{P}(\omega), & p < \infty, \\ \operatorname{ess\,sup}_{\omega\in\Omega} \|v(\omega)\|_{\mathcal{X}}, & p = \infty. \end{cases}$$

In this context, a function $v\colon \Omega \to \mathcal{X}$ is said to be strongly measurable if there exists a sequence of simple functions $v_n\colon \Omega \to \mathcal{X}$, such that for almost every $\omega \in \Omega$ we have $\lim_{n\to\infty} v_n(\omega) = v(\omega)$. Note that we also have the usual inclusion $L^p(\Omega;\mathcal{X}) \supset L^q(\Omega;\mathcal{X})$ provided that $1 \leq p \leq q \leq \infty$. Given that $\big(\mathcal{X}, (\cdot,\cdot)_{\mathcal{X}}\big)$ is a separable Hilbert space, the Bochner space $L^2(\Omega;\mathcal{X})$ is a separable Hilbert space as well, where the inner product is defined as

$$(u,v)_{L^2} := \int_\Omega \big(u(\omega), v(\omega)\big)_{\mathcal{X}} \,\mathrm{d}\mathbb{P}(\omega).$$

In particular, this space is isometrically isomorphic to the tensor product space $L^2(\Omega) \otimes \mathcal{X}$, we refer to [25] for the details.

Subsequently, we will always equip the space $\mathbb{R}^d$ with the Euclidean norm $\|\cdot\|_2$ induced by the canonical inner product $\langle\cdot,\cdot\rangle$ and $\mathbb{R}^{d\times d}$ with the norm $\|\cdot\|_F$ induced by the Frobenius inner product $\langle \boldsymbol{A}, \boldsymbol{B}\rangle_F := \operatorname{tr}(\boldsymbol{A}^\intercal \boldsymbol{B})$. To account for the anisotropies generated by the cardiac fibers, we consider a conductivity tensor as proposed and analyzed in [19], see also [15]. It is of the form

$$\boldsymbol{G}(\boldsymbol{x},\omega) := g\boldsymbol{I} + \big(\|\boldsymbol{V}(\boldsymbol{x},\omega)\|_2 - g\big) \frac{\boldsymbol{V}(\boldsymbol{x},\omega)\boldsymbol{V}^\intercal(\boldsymbol{x},\omega)}{\boldsymbol{V}^\intercal(\boldsymbol{x},\omega)\boldsymbol{V}(\boldsymbol{x},\omega)}, \tag{3}$$

where $g > 0$ is a given value and $\boldsymbol{V} \in L^\infty\big(\Omega; L^\infty(D;\mathbb{R}^d)\big)$ is a random vector field. Moreover, we require that there exist some constants $b_{\min} \leq 1$ and $b_{\max} \geq 1$ such that $b_{\min} \leq g \leq b_{\max}$ and

$$b_{\min} \leq \operatorname{ess\,inf}_{\boldsymbol{x}\in D} \|\boldsymbol{V}(\boldsymbol{x},\omega)\|_2 \leq \operatorname{ess\,sup}_{\boldsymbol{x}\in D} \|\boldsymbol{V}(\boldsymbol{x},\omega)\|_2 \leq b_{\max} \quad \mathbb{P}\text{-almost surely.} \tag{4}$$

The model (3) represents a medium that has homogeneous diffusion strength g perpendicular to $\boldsymbol{V}$ and diffusion strength $\|\boldsymbol{V}(\boldsymbol{x},\omega)\|_2$ in the direction of $\boldsymbol{V}$. The randomness of the specific direction and length of $\boldsymbol{V}$ therefore quantifies the uncertainty of this notable direction and its diffusion strength.

Lemma 2.1. *A conductivity tensor of the form* (3) *is well-defined and indeed also satisfies a uniform ellipticity condition, i.e.*

$$b_{\min} \leq \operatorname{ess\,inf}_{\boldsymbol{x}\in D} \|\boldsymbol{G}(\boldsymbol{x},\omega)\|_2 \leq \operatorname{ess\,sup}_{\boldsymbol{x}\in D} \|\boldsymbol{G}(\boldsymbol{x},\omega)\|_2 \leq b_{\max} \quad \mathbb{P}\text{-\textit{almost surely.}} \tag{5}$$

Proof. For almost every $\omega \in \Omega$ and almost every $\boldsymbol{x} \in D$, we have that $\boldsymbol{G}(\boldsymbol{x},\omega)$ is well-defined, because of

$$\boldsymbol{V}^\intercal(\boldsymbol{x},\omega)\boldsymbol{V}(\boldsymbol{x},\omega) = \|\boldsymbol{V}(\boldsymbol{x},\omega)\|_2^2 \geq b_{\min}^2 > 0,$$

and clearly symmetric. Furthermore, we can choose $\boldsymbol{u}_2, \ldots, \boldsymbol{u}_d \in \mathbb{R}^d$ that are perpendicular to $\boldsymbol{V}(\boldsymbol{x},\omega)$ and are linearly independent. Thus, for $i = 2, \ldots, d$, it holds that

$$\boldsymbol{G}(\boldsymbol{x},\omega)\boldsymbol{u}_i = g\boldsymbol{u}_i \quad \text{and} \quad \boldsymbol{G}(\boldsymbol{x},\omega)\boldsymbol{V}(\boldsymbol{x},\omega) = \|\boldsymbol{V}(\boldsymbol{x},\omega)\|_2\boldsymbol{V}(\boldsymbol{x},\omega).$$

Consequently, we obtain for almost every $\omega \in \Omega$ and almost every $\boldsymbol{x} \in D$ that

$$\begin{aligned}\lambda_{\min}\big(\boldsymbol{G}(\boldsymbol{x},\omega)\big) &= \min\{g, \|\boldsymbol{V}(\boldsymbol{x},\omega)\|_2\} \geq b_{\min},\\ \lambda_{\max}\big(\boldsymbol{G}(\boldsymbol{x},\omega)\big) &= \max\{g, \|\boldsymbol{V}(\boldsymbol{x},\omega)\|_2\} \leq b_{\max}.\end{aligned}$$

This shows (5). □

2.2 Karhunen-Loève expansion

To make random (vector) fields feasible for numerical computations, we separate the spatial variable $\boldsymbol{x}$ and the stochastic parameter ω by computing the Karhunen-Loève expansion. To this end, we require the expectation and the covariance of the underlying random field. For example, in case of $\boldsymbol{V}$, they are given by

$$\mathbb{E}[\boldsymbol{V}](\boldsymbol{x}) = \int_\Omega \boldsymbol{V}(\boldsymbol{x},\omega)\,\mathrm{d}\mathbb{P}(\omega)$$

and

$$\operatorname{Cov}[\boldsymbol{V}](\boldsymbol{x},\boldsymbol{x}') = \int_\Omega \boldsymbol{V}_0(\boldsymbol{x},\omega)\boldsymbol{V}_0^\intercal(\boldsymbol{x}',\omega)\,\mathrm{d}\mathbb{P}(\omega),$$

respectively, where

$$\boldsymbol{V}_0(\boldsymbol{x},\omega) := \boldsymbol{V}(\boldsymbol{x},\omega) - \mathbb{E}[\boldsymbol{V}](\boldsymbol{x})$$

denotes the centered vector field.

Given the eigenpairs $\{\lambda_k, \boldsymbol{\psi}_k\}_k$ of the Hilbert-Schmidt operator $\mathcal{C}$ defined by $\operatorname{Cov}[\boldsymbol{V}]$, that is

$$(\mathcal{C}\boldsymbol{u})(\boldsymbol{x}) := \int_D \operatorname{Cov}[\boldsymbol{V}](\boldsymbol{x},\boldsymbol{x}')\boldsymbol{u}(\boldsymbol{x}')\,\mathrm{d}\boldsymbol{x}',$$

the Karhunen-Loève expansion of $\boldsymbol{V}$ reads

$$\boldsymbol{V}(\boldsymbol{x},\omega) = \mathbb{E}[\boldsymbol{V}](\boldsymbol{x}) + \sum_{k=1}^{\infty} \sqrt{\lambda_k}\boldsymbol{\psi}_k(\boldsymbol{x})Y_k(\omega). \tag{6}$$

Herein, the uncorrelated, normalised and centered random variables $\{Y_k\}_k$ are obtained in accordance with

$$Y_k(\omega) := \frac{1}{\sqrt{\lambda_k}} \int_D \boldsymbol{V}_0^{\intercal}(\boldsymbol{x},\omega)\boldsymbol{\psi}_k(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}.$$

Note that as $\boldsymbol{V} \in L^\infty\big(\Omega; L^\infty(D;\mathbb{R}^d)\big)$ we particularly know that $\boldsymbol{\psi}_k \in L^\infty(D;\mathbb{R}^d)$ and $Y_k \in L^\infty\big(\Omega;\mathbb{R}\big)$, see [19].

Now, by introducing $\sigma_k := \sqrt{\lambda_k}\,\|Y_k\|_{L^\infty(\Omega;\mathbb{R})}$ we can assume, without loss of generality, that $Y_k \in [-1,1]$ and thus may instead consider the vector field $\boldsymbol{V}$ in the parametrised form

$$\boldsymbol{V}(\boldsymbol{x},\boldsymbol{\omega}) = \mathbb{E}[\boldsymbol{V}](\boldsymbol{x}) + \sum_{k=1}^{\infty} \sigma_k\boldsymbol{\psi}_k(\boldsymbol{x})\omega_k, \tag{7}$$

where $\boldsymbol{\omega} := [\omega_k]_{k\in\mathbb{N}} \in \square := [-1,1]^{\mathbb{N}}$ and ω_k is the canonical random variable on the probability space $\big([-1,1], \mathcal{B}([-1,1]), \mathbb{P}_{Y_k}\big)$. Consequently, we can also view $\boldsymbol{G}(\boldsymbol{x},\boldsymbol{\omega})$ as being parametrised by $\boldsymbol{\omega}$.

We now impose some common assumptions, which make the Karhunen-Loève expansion computationally feasible.

Assumption 2.2. *The random variables $\{Y_k\}_{k\in\mathbb{N}}$ are independent and uniformly distributed on $[-1,1]$, which indeed implies that $\sigma_k = \sqrt{3\lambda_k}$ and that $\mathbb{P}_{Y_k}$ coincides with the normalised Lebesgue measure on $[-1,1]$. Moreover, the sequence $\boldsymbol{\gamma} = \{\gamma_k\}_k$, given by*

$$\gamma_k := \|\sigma_k\boldsymbol{\psi}_k\|_{L^\infty(D;\mathbb{R}^d)},$$

is at least in $\ell^1(\mathbb{N})$, where we have set $\boldsymbol{\psi} := \mathbb{E}[\boldsymbol{V}]$ and $\sigma_0 := 1$.

2.3 Discretization of the random vector field

The Karhunen-Loève expansion in the form of (7) cannot directly be used on a computer. In what follows, we therefore present a means how the Karhunen-Loève expansion can be numerically approximated with finite elements. As before, we only consider the vector valued case here. Let the random vector field be given by its expectation $\mathbb{E}[\boldsymbol{V}](\boldsymbol{x}) = [\mathbb{E}_i[\boldsymbol{V}](\boldsymbol{x})]_{i=1}^d$ and its covariance function $\operatorname{Cov}[\boldsymbol{V}](\boldsymbol{x},\boldsymbol{x}') = [\operatorname{Cov}_{i,j}[\boldsymbol{V}](\boldsymbol{x},\boldsymbol{x}')]_{i,j=1}^d$, which we assume to be at least continuous. Moreover, let $\{\boldsymbol{x}_i\}_{i=1}^n \subset D$ be the vertices of the nodal finite element basis $\{\phi_1,\ldots,\phi_n\}$, i.e. $\phi_i(\boldsymbol{x}_j) = \delta_{i,j}$, coming from the finite element space $\mathcal{S}_L$ where $n = n_L = \dim(\mathcal{S}_L)$, see Subsection 3.1. Then, we can approximate the expectation by its finite element interpolant

$$\mathbb{E}[\boldsymbol{V}](\boldsymbol{x}) \approx \sum_{i=1}^n \mathbb{E}[\boldsymbol{V}](\boldsymbol{x}_i)\phi_i(\boldsymbol{x})$$

and in complete analogy the covariance by

$$\operatorname{Cov}[\boldsymbol{V}](\boldsymbol{x},\boldsymbol{x}') \approx \sum_{i,j=1}^n \operatorname{Cov}[\boldsymbol{V}](\boldsymbol{x}_i,\boldsymbol{x}_j)\phi_i(\boldsymbol{x})\phi_j(\boldsymbol{x}').$$

In order to determine the Karhunen-Loève expansion of $\boldsymbol{V}$, we have to solve the operator eigenvalue problem

$$\int_D \operatorname{Cov}[\boldsymbol{V}](\boldsymbol{x},\boldsymbol{x}')\boldsymbol{\psi}(\boldsymbol{x}')\,\mathrm{d}\boldsymbol{x}' = \lambda\boldsymbol{\psi}(\boldsymbol{x}).$$

Thus, by replacing $\operatorname{Cov}[\boldsymbol{V}]$ with its finite element interpolant and testing with respect to the basis functions $\phi_i \otimes \boldsymbol{e}_j$, $i = 1,\ldots,n$, $j = 1,\ldots,d$, where $\{\boldsymbol{e}_j\}_j$ is the canonical basis of $\mathbb{R}^d$, we end up with the generalized algebraic eigenvalue problem

$$\begin{bmatrix}\boldsymbol{M} & & \\ & \ddots & \\ & & \boldsymbol{M}\end{bmatrix}\boldsymbol{C}\begin{bmatrix}\boldsymbol{M} & & \\ & \ddots & \\ & & \boldsymbol{M}\end{bmatrix}\boldsymbol{v} = \lambda\begin{bmatrix}\boldsymbol{M} & & \\ & \ddots & \\ & & \boldsymbol{M}\end{bmatrix}\boldsymbol{v}, \quad \boldsymbol{v} \in \mathbb{R}^{dn}. \tag{8}$$

Herein, the matrix

$$\boldsymbol{C} := \begin{bmatrix}\big[\operatorname{Cov}_{1,1}[\boldsymbol{V}](\boldsymbol{x}_i,\boldsymbol{x}_j)\big]_{i,j=1}^n & \cdots & \big[\operatorname{Cov}_{1,d}[\boldsymbol{V}](\boldsymbol{x}_i,\boldsymbol{x}_j)\big]_{i,j=1}^n \\ \vdots & \ddots & \vdots \\ \big[\operatorname{Cov}_{d,1}[\boldsymbol{V}](\boldsymbol{x}_i,\boldsymbol{x}_j)\big]_{i,j=1}^n & \cdots & \big[\operatorname{Cov}_{d,d}[\boldsymbol{V}](\boldsymbol{x}_i,\boldsymbol{x}_j)\big]_{i,j=1}^n\end{bmatrix} \in \mathbb{R}^{dn\times dn}$$

is the covariance function evaluated in all combinations of grid points, while

$$\boldsymbol{M} := [m_{i,j}]_{i,j=1}^n \in \mathbb{R}^{n\times n} \quad \text{with } m_{i,j} := \int_D \phi_j\phi_i\,\mathrm{d}\boldsymbol{x}$$

denotes the finite element mass matrix.

The algebraic eigenvalue problem (8) can now efficiently be solved by means of the pivoted Cholesky decomposition as follows: Let $\boldsymbol{C} \approx \boldsymbol{L}\boldsymbol{L}^\intercal$ with $\boldsymbol{L} \in \mathbb{R}^{dn\times M}$ and $M \ll n$ be the low-rank approximation generated by the pivoted Cholesky decomposition of $\boldsymbol{C}$ as described in, e.g. [16, 18]. Then, we approximate the eigenvalue problem (8) by

$$\begin{bmatrix}\boldsymbol{M} & & \\ & \ddots & \\ & & \boldsymbol{M}\end{bmatrix}\boldsymbol{L}\boldsymbol{L}^\intercal\begin{bmatrix}\boldsymbol{M} & & \\ & \ddots & \\ & & \boldsymbol{M}\end{bmatrix}\boldsymbol{v} = \lambda\begin{bmatrix}\boldsymbol{M} & & \\ & \ddots & \\ & & \boldsymbol{M}\end{bmatrix}\boldsymbol{v}, \quad \boldsymbol{v} \in \mathbb{R}^{dn}. \tag{9}$$

This eigenvalue problem is equivalent to the much smaller eigenvalue problem

$$\boldsymbol{L}^\intercal \begin{bmatrix} \boldsymbol{M} & & \\ & \ddots & \\ & & \boldsymbol{M} \end{bmatrix} \boldsymbol{L}\tilde{\boldsymbol{v}} = \lambda\tilde{\boldsymbol{v}}, \quad \tilde{\boldsymbol{v}} \in \mathbb{R}^M. \tag{10}$$

In particular, if $\tilde{\boldsymbol{v}}_i$ is an eigenvector of (10) with eigenvalue λ_i, then $\boldsymbol{v}_i := \boldsymbol{L}\tilde{\boldsymbol{v}}_i$ is an eigenvector of (9) with eigenvalue λ_i. Moreover, there holds

$$\boldsymbol{v}_i^\intercal \begin{bmatrix} \boldsymbol{M} & & \\ & \ddots & \\ & & \boldsymbol{M} \end{bmatrix} \boldsymbol{v}_j = \lambda_i \delta_{i,j}.$$

Remark 2.3. *The cost for computing the pivoted Cholesky decomposition is $\mathcal{O}(dnM^2)$ and, since all entries of $\boldsymbol{C}$ can be computed on the fly without the need of storing the entire matrix $\boldsymbol{C}$, the storage cost is $\mathcal{O}(dnM)$. Moreover, the small eigenvalue problem* (10) *can be solved with cost $\mathcal{O}(M^3)$. Thus, since usually $M \ll n$, the overall cost for computing the Karhunen-Loève expansion of $\boldsymbol{V}$ by the suggested approach is also $\mathcal{O}(dnM^2)$ in total.*

Based on the suggested low-rank approach, we end up with a discretized random field of the form

$$\tilde{\boldsymbol{V}}(\boldsymbol{x},\boldsymbol{\omega}) = \sum_{i=1}^n \mathbb{E}[\boldsymbol{V}](\boldsymbol{x}_i)\phi_i(\boldsymbol{x}) + \theta \sum_{k=1}^M \sigma_k\omega_k \sum_{i=1}^n \boldsymbol{c}_{k,i}\phi_i(\boldsymbol{x}), \quad \boldsymbol{\omega} \in [-1,1]^M, \tag{11}$$

where the coefficients $\boldsymbol{c}_{k,i} \in \mathbb{R}^d$ are obtained from combining all coefficients from the eigenvector $\boldsymbol{v}_k$ that interact with the basis function ϕ_i. Moreover, we introduce the scaling parameter $\theta > 0$ to guarantee (4) in our numerical studies.

2.4 Quantities of interest

Due to the randomness of the heart fibers' orientations as described above, the monodomain equation (1) now translates into the following parametric version provided for all $\boldsymbol{\omega} \in [-1,1]^M$:

$$\frac{\partial u(\boldsymbol{z},\boldsymbol{\omega})}{\partial t} - \nabla \cdot \big(\boldsymbol{G}(\boldsymbol{x})\nabla u(\boldsymbol{z},\boldsymbol{\omega})\big) + I_{\text{ion}}\big(u(\boldsymbol{z},\boldsymbol{\omega})\big) = I_{\text{app}}(\boldsymbol{z}), \quad \boldsymbol{z} := (\boldsymbol{x},t) \in D \times (0,T]. \tag{12}$$

Our aim is to determine statistics of the random solution $u(\boldsymbol{z},\boldsymbol{\omega})$, which amounts to the evaluation of the high-dimensional integral given by

$$\text{QoI}[u] = \int_{[-1,1]^M} \mathcal{F}\big(u(\cdot,\boldsymbol{\omega})\big)\rho(\boldsymbol{\omega})\,\mathrm{d}\boldsymbol{\omega}. \tag{13}$$

Here, $\rho(\boldsymbol{\omega}) = \prod_{i=1}^M \rho_k(\omega_k)$ is the joint density function of $\boldsymbol{\omega}$ from (11) and $\mathcal{F}$ denotes a functional that encodes a particular quantity of interest. We shall focus here on three different quantities of interest.

Transmembrane potential. The transmembrane potential over the totality of the heart geometry and its evolution in time is the quantity obtained by solving the monodomain equation. This is demonstrated as an electrical potential wave travelling through the heart, cf. Figure 2. The functional $\mathcal{F}$ in this case is simply the identity function, i.e.

$$\mathcal{F}\big(u(\cdot,\boldsymbol{\omega})\big) = u(\cdot,\boldsymbol{\omega}).$$

We remark that considering a fine discretization in space and time, the full information on the transmembrane potential represents a high-dimensional output that might easily become a burden at the memory level in a context of a UQ study.

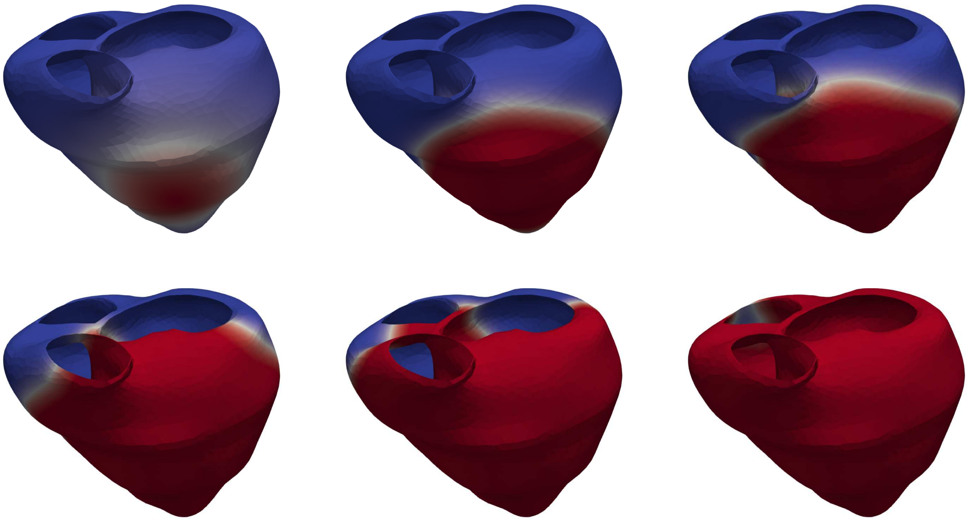


Figure 2: Wavefront propagation of transmembrane potential.

Activation map. The transmembrane potential can be used in order to extract the activation map of the heart. This reduces the size of the output to that of the dimension indicating the times at which cells are activated. The activation time $a(\boldsymbol{x}_0,\boldsymbol{\omega})$ at a given location $\boldsymbol{x}_0$ is defined as the right inverse

$$a(\boldsymbol{x}_0,\boldsymbol{\omega}) = \min\{t \in [0,T] : u(\boldsymbol{x}_0,t,\boldsymbol{\omega}) \geq u_{\text{th}}\}.$$

The functional $\mathcal{F}$ for this case can therefore be written as $\mathcal{F}\big(u(\cdot,\boldsymbol{\omega})\big) = a(\boldsymbol{x}_0,\boldsymbol{\omega})$.

Action potential. Another relevant quantity of interest is the evaluation of the transmembrane time evolution at a given location $\boldsymbol{x}_0$. Mathematically speaking, this corresponds to the

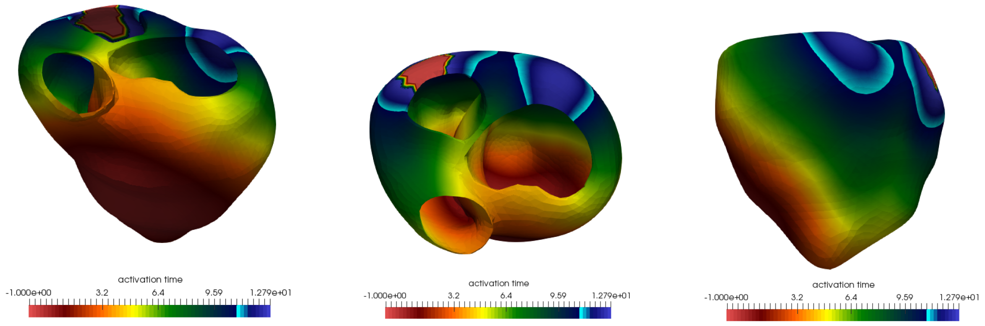


Figure 3: Activation map of a heart. Negative value signifies non activated region.

functional

$$\mathcal{F}\big(u(\cdot,\boldsymbol{\omega})\big) = u(\boldsymbol{x}_0, t, \boldsymbol{\omega}).$$

Regarding the approximation of the integral (13) for a given functional $\mathcal{F}$, we rely on high-dimensional quadrature methods that require solving the monodomain equation in quadrature points represented by different realizations of $\boldsymbol{\omega} \in [-1,1]^M$. The resulting procedure is a sampling method and requires a finite element solve for every sample.

3 Discretization of the monodomain equation

The parametric monodomain equation (12) can be solved numerically for all parameters $\boldsymbol{\omega} \in [-1,1]^M$ by means of finite elements in space and finite differences in time, i.e. by using a sequential time-stepping method. In view of employing multilevel quadrature methods for approximating the integral (13), we set the stage for a similar refinement rate of the space and time grid resolutions. In particular, we employ an all-at-once approach in space and time, where we assemble a large space-time system that is solved in parallel [26].

This approach allows for a similar error decay with respect to the space and time discretization steps. In addition, it enhances the parallel scalability of the numerical method, by allowing parallelization also in the time dimension. For a comprehensive review of parallel-in-time methods, see [9].

3.1 Space-time assembly of the heat equation

Let us consider a nested sequence of shape regular tetrahedralizations $\{\mathcal{T}_l\}_{l\geq 0}$ of the spatial domain D, where each $\mathcal{T}_l$ is of mesh size $h_l \sim 2^{-l}$. For all levels $l \geq 0$, we define the continuous, piecewise linear finite element spaces

$$\mathcal{S}_l = \{v_l \in C^0(D) : v_{l|T} \in \mathbb{P}_1(T),\ T \in \mathcal{T}_l\}.$$

We denote with $\{\phi_{l,i}\}_{i=1}^{n_l} \subset \mathbb{P}_1$ the sets of linear nodal basis functions for $\mathcal{S}_l$. For each index l we also partition the time interval $[0, T]$ into $m_l - 1$ equisized subintervals of length $\Delta t_l = T/(m_l - 1)$, such that $\Delta t_l \sim h_l$. This uniform partition is thus given by the nodes $t_{l,k} = (k-1)\Delta t_l$ with $k = 1, \ldots, m_l$.

We start by neglecting the non-linear term from (12), that is I_{ion}, to derive instead the space-time linear system arising from the closely related heat equation. For the sake of readability, we also assume that a particular realization of parameter $\boldsymbol{\omega} \in [-1, 1]^M$ is given, and therefore disregard it for the analysis that follows. Assuming that the solution $u(\boldsymbol{x}, t)$ is sufficiently regular in D, we derive the weak formulation:

$$\begin{aligned}&\text{for all } t \in (0, T], \text{ find } u(\cdot, t) \in H^1(D) \text{ such that}\\ &\int_D \frac{\partial u(\boldsymbol{x}, t)}{\partial t} v(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} + \int_D \boldsymbol{G}(\boldsymbol{x}) \nabla u(\boldsymbol{x}, t) \nabla v(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \int_D I_{\text{app}}(\boldsymbol{x}, t) v(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}\\ &\qquad\qquad\qquad\qquad\qquad\qquad\text{for all } v \in H^1(D).\end{aligned}$$

Assosiated with this weak formulation, we have the Galerkin approximation on level l given by:

$$\begin{aligned}&\text{for all } t \in (0, T], \text{ find } u_l(\cdot, t) \in \mathcal{S}_l \text{ such that}\\ &\int_D \frac{\partial u_l(\boldsymbol{x}, t)}{\partial t} v_l(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} + \int_D \boldsymbol{G}(\boldsymbol{x}) \nabla u_l(\boldsymbol{x}, t) \nabla v_l(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \int_D I_{\text{app}}(\boldsymbol{x}, t) v_l(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}\\ &\qquad\qquad\qquad\qquad\qquad\qquad\text{for all } v_l \in S_l.\end{aligned}$$

As each function $u_l(\cdot, t) \in \mathcal{S}_l$ can be expressed as a linear combination of the corresponding basis elements, i.e.

$$u_l(\boldsymbol{x}, t) = \sum_{i=1}^{n_l} u_{l,i}(t) \phi_{l,i}(\boldsymbol{x}), \tag{14}$$

we can recover the semi-discrete formulation of the problem:

$$\boldsymbol{M}_l \frac{\partial \boldsymbol{u}_l(t)}{\partial t} + \boldsymbol{K}_l \boldsymbol{u}_l(t) = \boldsymbol{I}_{\text{app},l}(t), \quad \boldsymbol{u}_l(t) = [u_{l,1}(t), \ldots, u_{l,n_l}(t)]^\intercal. \tag{15}$$

Here, $\boldsymbol{M}_l \in \mathbb{R}^{n_l \times n_l}$ and $\boldsymbol{K}_l \in \mathbb{R}^{n_l \times n_l}$ are the mass and stiffness matrices on level l defined as

$$\boldsymbol{M}_l := \left[\int_D \phi_{l,j}(\boldsymbol{x}) \phi_{l,i}(\boldsymbol{x}) d\boldsymbol{x}\right]_{i,j=1}^{n_l}, \quad \boldsymbol{K}_l := \left[\int_D \boldsymbol{G}(\boldsymbol{x}) \nabla \phi_{l,j}(\boldsymbol{x}) \nabla \phi_{l,i}(\boldsymbol{x}) d\boldsymbol{x}\right]_{i,j=1}^{n_l},$$

and the right-hand side $\boldsymbol{I}_{\text{app},l}(t) \in \mathbb{R}^{n_l}$ is

$$\boldsymbol{I}_{\text{app},l}(t) := \left[\int_D I_{\text{app},l}(\boldsymbol{x}, t) \phi_{l,i}(\boldsymbol{x})\right]_{i=1}^{n_l}.$$

We next apply the second order Crank–Nicolson method for the time discretization of (15) and obtain for $k = 1, \dots, m_l - 1$ the system of equations

$$\left(\boldsymbol{M}_l + \frac{\Delta t_l}{2}\boldsymbol{K}_l\right)\boldsymbol{u}_{l,k+1} + \left(-\boldsymbol{M}_l + \frac{\Delta t_l}{2}\boldsymbol{K}_l\right)\boldsymbol{u}_{l,k} = \boldsymbol{I}_{\text{app},l,k} \quad \text{and} \quad \boldsymbol{u}_{l,k} := \boldsymbol{u}_l(t_{l,k}),$$
$$\text{with} \quad \boldsymbol{I}_{\text{app},l,k} := \frac{\Delta t_l}{2}\big(\boldsymbol{I}_{\text{app},l}(t_{l,k+1}) + \boldsymbol{I}_{\text{app},l}(t_{l,k})\big). \tag{16}$$

If we define $\boldsymbol{A}_l := \boldsymbol{M}_l + \frac{\Delta t_l}{2}\boldsymbol{K}_l$ and $\boldsymbol{B}_l := -\boldsymbol{M}_l + \frac{\Delta t_l}{2}\boldsymbol{K}_l$, the system of equations (16) can be summarized in compact form according to

$$\begin{bmatrix} \boldsymbol{A}_l & & & \\ \boldsymbol{B}_l & \boldsymbol{A}_l & & \\ & \ddots & \ddots & \\ & & \boldsymbol{B}_l & \boldsymbol{A}_l \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_{l,1} \\ \boldsymbol{u}_{l,2} \\ \vdots \\ \boldsymbol{u}_{l,m_l} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_{\text{app},l,1} \\ \boldsymbol{I}_{\text{app},l,2} \\ \vdots \\ \boldsymbol{I}_{\text{app},l,m_l} \end{bmatrix} \iff \boldsymbol{C}_l\boldsymbol{u}_l = \boldsymbol{I}_{\text{app},l}, \tag{17}$$

where $\boldsymbol{C}_l \in \mathbb{R}^{n_l m_l \times n_l m_l}$ is a large space-time system that can be distributed and solved in parallel and

$$\boldsymbol{u}_l := [\boldsymbol{u}_{l,1}, \boldsymbol{u}_{l,2}, \dots, \boldsymbol{u}_{l,m_l}]^\intercal \quad \text{and} \quad \boldsymbol{I}_{\text{app},l} := [\boldsymbol{I}_{\text{app},l,1}, \boldsymbol{I}_{\text{app},l,2}, \dots, \boldsymbol{I}_{\text{app},l,m_l}]^\intercal.$$

3.2 Space–time assembly of the monodomain equation

The discretization of (12) is an extension of the assembly procedure described in Section 3.1. In particular, the linear system (17) is modified to contain the discretization of the non-linear reaction term I_{ion}

$$\boldsymbol{C}_l\boldsymbol{u}_l + \boldsymbol{r}(\boldsymbol{u}_l) = \boldsymbol{I}_{\text{app},l}, \tag{18}$$

where $\boldsymbol{r}(\boldsymbol{u}_l) \in \mathbb{R}^{n_l m_l}$ is given by

$$\boldsymbol{r}(\boldsymbol{u}_l) := (\Delta t_l I_{m_l} \otimes M_l)\boldsymbol{I}_{\text{ion}}(\boldsymbol{u}_l) \quad \text{with } \boldsymbol{I}_{\text{ion}}(\boldsymbol{u}_l) := [I_{\text{ion}}(u_1), \dots, I_{\text{ion}}(u_{n_l m_l})]^\intercal.$$

Here, n and m are respectively the space and time degrees of freedom.

The non–linear equation (18) is solved by using Newton's method. The Jacobian $\boldsymbol{J}(\mathbf{u}_l) \in \mathbb{R}^{n_l m_l \times n_l m_l}$ of the non-linear operator on the left-hand side of (18) is given by

$$\boldsymbol{J}(\mathbf{u}_l) = \boldsymbol{C}_l + (\Delta t_l \boldsymbol{I}_{m_l} \otimes \boldsymbol{M}_l) \cdot \boldsymbol{J}\boldsymbol{I}_{\text{ion}}(\boldsymbol{u}_l)$$

with $\boldsymbol{J}\boldsymbol{I}_{\text{ion}}(\boldsymbol{u}_l) \in \mathbb{R}^{n_l m_l \times n_l m_l}$ being the block diagonal matrix

$$\boldsymbol{J}\boldsymbol{I}_{\text{ion}}(\mathbf{u}_l) := \begin{bmatrix} I'_{\text{ion}}(u_1) & & & \\ & I'_{\text{ion}}(u_2) & & \\ & & \ddots & \\ & & & I'_{\text{ion}}(u_{n_l m_l}) \end{bmatrix}.$$

3.3 Solution strategy

We rely on a solution strategy that retakes the main features of the one used in [3]. Specifically, we approach the parallelization of the solver, the preconditioning and the Newton initial guess strategy as follows.

Multiple time blocks strategy. Combination of the Newton's method with the space–time all–at–once approach may face a problem of convergence when a large time interval is required. We resort to a multiple time blocks strategy, see also [4], consisting of decomposing the original time interval into smaller chunks (the so–called time blocks). The problem is solved sequentially on each of the time blocks, provided that the initial condition of the current time block is set to the final state of the previous one. Considering a time interval $[0, T]$, uniformly partitioned in m time steps, the latter is further divided in K multiple blocks such that there exists $T' \in \mathbb{R}$ and $m' \in \mathbb{N}$ for which we have $T = KT'$ and $m = Km'$. We denote the solution on the $k-$th time block with

$$\mathbf{u}_{[k]} = [\mathbf{u}_{[k],1}, \dots, \mathbf{u}_{[k],m'}]^\intercal \in \mathbb{R}^{nm'},$$

where n is the number of spatial degrees of freedom. The initial condition of the current time block to solve translates to,

$$\mathbf{u}_{[k+1],0} = \mathbf{u}_{[k],m'} \quad \text{for all } k = 1, \dots, K-1,$$

with respect to that computed from the previous time block. The Newton initial guess is further set to be the final time step solution of the previous block, generalized to all the time steps, that is

$$\mathbf{u}^{(0)}_{[k+1]} = [\mathbf{u}_{[k],m'}, \dots, \mathbf{u}_{[k],m'}]^\intercal \in \mathbb{R}^{nm'}.$$

Preconditioning. Every linear problem in the form of (17) (arising at every Newton iteration) is solved by means of a space-time parallel GMRES with block Jacobi preconditioner. The spectral analysis of the space-time system in (17) motivates this choice, see [5] for all the details.

Newton initial guess. We use a sample based Newton initial guess strategy. For each sample, the first Newton iterate is provided by an unperturbed reference solution. This is done in two ways, locally, according to the multiple time blocks strategy, and globally with respect to the original time interval. We compare their performances in Figure 4 with the direct method and the multiple time blocks strategy described above, by solving the monodomain equation on six time blocks. The acronyms DM, MTB, LNIG and GNIG designate respectively the Direct Method, the Multiple Time Block, the Local Newton Initial Guess and the Global Newton Initial Guess.

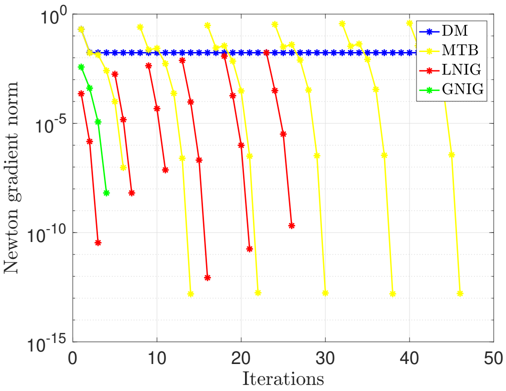


Figure 4: Comparison of the different Newton initial guess strategies for a sample on cube.

4 Multilevel quadrature methods

In order to compute the quantities of interest under consideration, i.e.

$$\mathrm{QoI}[u] = \int_{[-1,1]^M} \mathcal{F}\big(u(\cdot,\boldsymbol{\omega})\big)\rho(\boldsymbol{\omega})\,\mathrm{d}\boldsymbol{\omega},$$

we employ multilevel quadrature methods. To this end, we introduce the sequence $\mathcal{F}_l[u](\boldsymbol{\omega}) := \mathcal{F}\big(u_l(\cdot,\boldsymbol{\omega})\big)$ that approximates $\mathcal{F}[u](\boldsymbol{\omega}) := \mathcal{F}\big(u(\cdot,\boldsymbol{\omega})\big)$ and, instead of the single level estimator

$$\mathrm{QoI}_L^{\mathrm{SL}}[u] := \mathcal{Q}_L\big(\mathcal{F}_L[u](\cdot)\big), \tag{19}$$

consider the multilevel estimator

$$\mathrm{QoI}_L^{\mathrm{ML}}[u] := \sum_{l=0}^{L} \mathcal{Q}_{L-l}\big(\mathcal{F}_l[u](\cdot) - \mathcal{F}_{l-1}[u](\cdot)\big), \tag{20}$$

where $\{\mathcal{Q}_l\}_{l\geq 0}$ is a sequence of quadrature rules, and $\mathcal{Q}_{-1} \equiv 0$. This is the standard and widely used multilevel estimator, which has been introduced in [1, 11, 20]. It consists in defining the multilevel estimator as the sum of quadratures applied to the difference of finite element solutions. The construction of this multilevel estimator has been shown to be equivalent to the sparse grid combination technique of the finite element space and the stochastic space, compare [10, 17]. In particular, the roles of these spaces can be exchanged to present the multilevel estimator (20) in a different way. Namely, it can equivalently be written as

$$\mathrm{QoI}_L^{\mathrm{ML}}[u] := \sum_{l=0}^{L} (\mathcal{Q}_l - \mathcal{Q}_{l-1})\big(\mathcal{F}_{L-l}[u](\cdot)\big). \tag{21}$$

This is especially favourable in case of non-nested meshes. In addition, the computational complexity is reduced when nested quadrature points are applied, see [14] for the details.

For the approximation error of the multilevel quadrature, there holds a sparse tensor product-like error estimate. If $\varepsilon_l \to 0$ is a monotonically decreasing sequence with $\varepsilon_l \cdot \varepsilon_{L-l} = \varepsilon_L$ for every $L \in \mathbb{N}$ and

$$\|\mathcal{Q}_{L-l}\mathcal{F}[u] - \mathrm{QoI}[u]\| \leq c_1\varepsilon_{L-l} \quad \text{and} \quad \|\mathcal{F}[u] - \mathcal{F}_l[u]\| \leq c_2\varepsilon_l$$

for some suitable norms and constants $c_1, c_2 > 0$, then

$$\big\| \mathrm{QoI}_L^{\mathrm{ML}}[u] - \mathrm{QoI}[u]\big\| \leq CL\varepsilon_L$$

for a constant $C > 0$, provided that u and u_l are sufficiently regular. We refer to [16] for details on the multilevel quadrature. Moreover, we remark that the presented error estimate is based on error equilibration. It is, however, also possible to equilibrate the computational work or the degrees of freedom, see [12, 13].

An important component for the multilevel quadrature is the intergrid transfer of the data. The transfer of data from the coarse level to the fine level is not needed if only real-valued quantities of interest, based on point evaluations, are considered. Otherwise, the estimator (21) can efficiently be computed by transferring the data only after accumulating it for each level. The transfer of data from the fine level to the coarse level however is mandatory for each sample, as the Karhunen-Loève expansion has to be computed on the fine grid.

We remark that has been shown in [32] that the computation of the stiffness matrix with respect to the random diffusion field is consistent with a piecewise linear finite element discretization if the midpoint rule with respect to the current grid is applied. Therefore, for the transfer of the random fields from the finest level to the coarser levels, we perform an element–wise transfer based on the midpoint rule; that is, for every element of the coarse level, we assign an constant diffusion value corresponding to the fine element containing its center. Therefore, the assembly of the stiffness matrix on the coarser levels can be performed with linear cost relative to the particular level of discretization.

5 Numerical experiments

5.1 Setup

The numerical experiments have been conducted for three test-case geometries: a cube, an idealized ventricle and a heart geometry (atria excluded) that was acquired through real patient CCT data. We will refer to the latter geometry with the term “realistic heart”. The simulations have been realized using `SLOTH`, see [30], a UQ Python library developed at the Institute of Computational Science (ICS) in Lugano. For this work, we extended it to the monodomain equation (and in general to all types of $3+1$ dimensional PDEs) by employing `Utopia`, see [33], for the finite element formulation.

Parameters for the monodomain equation. Regarding the models (1) and (2), we will always rely on the following parameters:

- The values for the ionic channel model $I_{\text{ion}}(u)$ in (2) are set as $\alpha = 1.4 \cdot 10^{-3}$ mV^{-2}ms^{-1}, $u_{\text{rest}} = 0$ mV, $u_{\text{th}} = 28$ mV, and $u_{\text{peak}} = 115$ mV.

- We choose
$$I_{\text{app}}(\boldsymbol{x}, t) = \left(u_{\text{rest}} + u_{\text{peak}} \exp\left(-\frac{(\boldsymbol{x} - \boldsymbol{x}_0)^2}{\sigma^2} \right) \right) \chi_{[0,t_1)}(t),$$
where $t_1 = \Delta t = 0.005$ ms is the function we rely on for the applied stimulus. Parameters σ and $\boldsymbol{x}_0$ represent respectively the power and the location of the stimulus. They are geometry dependent.

Parameters for Karhunen-Loève expansion. For the numerical experiments, we introduce a scaling factor $\theta \in \mathbb{R}$ into the covariance kernel $\text{Cov}[\boldsymbol{V}]$ to be able to easily scale the applied perturbation size. The stochastic dimension M in the computed, parametric Karhunen-Loève expansion (11) arises from prescribing the truncation error $\epsilon = 10^{-2}$ in the pivoted Cholesky decomposition. The other parameters used in the Karhunen-Loève expansion differ from one experiment to another and are listed below:

- **Cube.** We consider isotropic diffusion and a scaling factor $\theta = 0.3$ on $D = [-0.5, 0.5]^3$. The covariance matrix is induced by the scalar covariance kernel $\text{Cov}[\boldsymbol{V}](\boldsymbol{x}, \boldsymbol{x}') = \theta^2 e^{\frac{-\|\boldsymbol{x}-\boldsymbol{x}'\|_2^2}{\sigma_{\text{KL}}}}$ with $\sigma_{\text{KL}} = 0.25$. The low-rank Cholesky approximation of the covariance matrix yielded the stochastic dimension $M = 66$. The mean diffusion is set to $\mathbb{E}[\boldsymbol{V}](\boldsymbol{x}) = 3.325 \cdot 10^{-3}$mm^2 ms^{-1} for all $\boldsymbol{x} \in D$.

- **Idealized ventricle.** We consider isotropic diffusion and the scaling factor $\theta = 0.3$. The covariance kernel is given by $\text{Cov}[\boldsymbol{V}](\boldsymbol{x}, \boldsymbol{x}') = \theta^2 e^{\frac{-\|\boldsymbol{x}-\boldsymbol{x}'\|_2^2}{\sigma_{\text{KL}}}}$ with $\sigma_{\text{KL}} = 0.5$. The stochastic dimension is given by $M = 87$. The mean diffusion is set to $\mathbb{E}[\boldsymbol{V}](\boldsymbol{x}) = 3.325 \cdot 10^{-3}$mm^2 ms^{-1} for all $\boldsymbol{x} \in D$. The domain's bounding box is given by $[0.80, 2.8] \times [1.01, 3.5] \times [0.60, 2.6]$.

- **Heart geometry.** We consider anisotropic diffusion with a block-diagonal covariance matrix given by $\text{Cov}_{i,j}[\boldsymbol{V}](\boldsymbol{x}, \boldsymbol{x}') = \delta_{i,j}\theta^2 e^{\frac{-\|\boldsymbol{x}-\boldsymbol{x}'\|_2^2}{\sigma_{\text{KL}}}}$ for $1 \leq i, j \leq 3$, where $\delta_{i,j}$ is the Kronecker delta. We set $\sigma_{\text{KL}} = 0.16$. The scaling factor is set to $\theta = 0.3$. The stochastic dimension is $M = 135$. Furthermore, the perpendicular diffusion in (3) is chosen as $g =$

$1.625 \cdot 10^{-3}\mathrm{mm}^2\ \mathrm{ms}^{-1}$ while $\mathbb{E}[\boldsymbol{V}](\boldsymbol{x})$ is specified later. The domain's bounding box is given by $[-0.41, 0.66] \times [0.21, 1.03] \times [0.56, 1.61]$.

Reference solution and error metrics. In all the convergence and work comparison graphs that follow, the referenced root mean square error in the H^q norm ($L^2 = H^0$ and H^1 for respectively $q = 0$ and $q = 1$) for $\mathrm{QoI}_l = \mathrm{QoI}_l^{\mathrm{SL}}$ or $\mathrm{QoI}_l = \mathrm{QoI}_l^{\mathrm{ML}}$ is given by

$$e_l = \Big(\mathbb{E}\Big[\|\,\mathrm{QoI}_l[u] - \mathrm{QoI}_{\mathrm{ref}}[u]\|^2_{L^2((0,T);H^q(D))}\Big]\Big)^{1/2} \tag{22}$$

in case of a space–time quantity of interest, such as the transmembrane potential,

$$e_l = \Big(\mathbb{E}\Big[\|\,\mathrm{QoI}_l[u] - \mathrm{QoI}_{\mathrm{ref}}[u]\|^2_{H^q(0,T)}\Big]\Big)^{1/2} \tag{23}$$

in case of a time quantity of interest, such as the action potential at a given location $\boldsymbol{x}_0$ in space, and

$$e_l = \Big(\mathbb{E}\Big[|\,\mathrm{QoI}_l[u] - \mathrm{QoI}_{\mathrm{ref}}[u]|^2\Big]\Big)^{1/2} \tag{24}$$

in the case of a scalar quantity of interest, such as the activation time at a given point $\boldsymbol{x}_0$ in space. Note that the mean in the above expressions is taken over the realisations of the possibly non-deterministic quadrature formulas. Specifically, for the Monte Carlo quadrature, the expectation for both, the single-level and multilevel runs, are approximated by averaging over 10 simulations at each level of precision for the nested case study, and 5 simulations for the non–nested example. The reference quantity of interest $\mathrm{QoI}_{\mathrm{ref}}[u]$ is computed by using $N =$ 10'000 samples drawn from the Halton sequence.

The intergrid transfer of a space–time quantity of interest from a given coarse level l to the fine level L, required to evaluate the error (22), is performed by means of the tensor product of the space and time interpolation matrices. Obviously, the intergrid transfer of a time quantity of interest from a given coarse level l to the fine level L, required to evaluate the error (23), is performed analogously by means of the time interpolation matrices.

Quadrature methods. In our experiments, we will consider the Monte Carlo (MC) and quasi-Monte Carlo (QMC) quadrature method and their multilevel pendants MLMC and MLQMC. Let us recall that the error is of order 2^{-2l} in the L^2 norm (2^{-l} in the H^1 norm) when using linear finite elements of mesh size $h_l = 2^{-l}$. Therefore, in view of the convergence rates for MC and QMC, the number of samples to be executed by these methods on a level l to get the same order of error 2^{-2l} is respectively given by

$$N_{\mathrm{MC},l} = 2^{4l} \quad \text{and} \quad N_{\mathrm{QMC},l} = 2^{2l}. \tag{25}$$

Regarding the H^1–error, the number of samples to be executed on a level l to get the same order of error 2^{-l} is respectively given by

$$N_{\mathrm{MC},l} = 2^{2l} \quad \text{and} \quad N_{\mathrm{QMC},l} = 2^l. \tag{26}$$

5.2 Scalar random diffusion for simple geometries and nested meshes

In these first experiments, we consider a scalar, thus isotropic, random diffusion for the sake of simplicity. The experiments are conducted on the cube and the idealized ventricle geometry. Note that the intergrid mesh transfer in case of the space–time dependent quantity of interest is straightforward as the meshes are nested.

5.2.1 Cube geometry

We use a hierarchy of $L = 6$ nested mesh levels. Starting from the finest level $l = L-1$, the coarser levels $l = 0, 1, \ldots, L-2$ are successively obtained from the prior finer levels $l+1 = 1, 2, \ldots, L-1$ by uniformly coarsening in space and time. The number of space-time degrees of freedom (DOF), the space and time discretization steps of all the different levels are reported in Table 1.

l	0	1	2	3	4	5
DOF	16	256	4'096	65'536	1'048'576	16'777'216
h	0.5	0.25	0.125	0.0625	0.03125	0.015625
Δt	0.16	0.08	0.04	0.02	0.01	0.005

Table 1: Details about the considered mesh hierarchy for the cube geometry.

Controlled convergence of the over-all error. We intend to estimate and verify the convergence rate for the quadrature methods under consideration. The number of samples on each level is determined by the sampling strategy for controlling the error, cf. Section 4, by using the sample numbers (25) and (26). We report in Figure 5 the convergence of the error in L^2 and H^1 norms.

The plots show that we recover the expected convergence rates of the general error for all quadrature methods tested. Note that this does not imply that these quadrature methods are all equally efficient, but rather that they yield the same precision with a vastly different balancing of samples on every level. This is demonstrated quite clearly in the corresponding work comparison plot found in Figure 6.

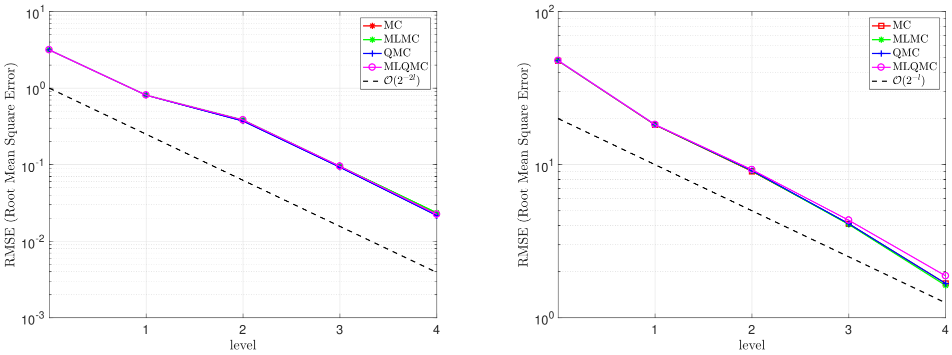


Figure 5: Convergence rate for the cube in L^2 (left) and H^1 (right) norms.

Asymptotical work behaviour. Equivalently to the controlled convergence concept with the previously introduced sampling strategy, we would like now to study the work in the context of the controlled error. To this end, we assume that the cost per solve on level l is given by $C_{\mathrm{FE},l} = 2^{\gamma dl}$, where γ is the complexity of the finite element solver used and d is the dimension of the physical problem considered (here $d = 4$). In view of (25), we can hence recover the total amount of work required by MC and QMC given a discretization level L with (19) by

$$W_{\mathrm{MC},L} = C_{\mathrm{FE},L} N_{\mathrm{MC},L} = 2^{\gamma dL} 2^{4L} = 2^{(\gamma d+4)L}$$

and

$$W_{\mathrm{QMC},L} = C_{\mathrm{FE},L} N_{\mathrm{QMC},L} = 2^{\gamma dL} 2^{2L} = 2^{(\gamma d+2)L}.$$

The total work for the multilevel pendants with L discretization levels can also be deduced from (25) with (20), and we can write

$$W_{\mathrm{MLMC},L} = \sum_{l=1}^{L} C_{\mathrm{FE},L} N_{\mathrm{MC},L-l} = \sum_{l=1}^{L} 2^{\gamma dl} 2^{4(L-l)} = 2^{4L} \sum_{l=1}^{L} 2^{(\gamma d-4)l}$$

and

$$W_{\mathrm{MLQMC},L} = \sum_{l=1}^{L} C_{\mathrm{FE},L} N_{\mathrm{QMC},L-l} = \sum_{l=1}^{L} 2^{\gamma dl} 2^{2(L-l)} = 2^{2L} \sum_{l=1}^{L} 2^{(\gamma d-2)l}.$$

These can further be reformulated as

$$W_{\mathrm{MLMC},L} = \begin{cases} L2^{4L} \text{ if } \gamma d = 4, \\ \dfrac{2^{\gamma dL} - 2^{4L}}{2^{\gamma d-4} - 1} \text{if } \gamma d \neq 4, \end{cases}$$

and

$$W_{\mathrm{MLQMC},L} = \begin{cases} L2^{2L} \text{ if } \gamma d = 2, \\ \dfrac{2^{\gamma dL} - 2^{2L}}{2^{\gamma d-2} - 1} \text{if } \gamma d \neq 2. \end{cases}$$

Therefore, the asymptotical work behaviour for MLMC is bounded by $\mathcal{O}(2^{4L})$ when $\gamma d < 4$ (with an additional log-factor if $\gamma d = 4$), and by $\mathcal{O}(2^{\gamma d L})$ if $\gamma d > 4$. Likewise, for MLQMC, the asymptotical work is bounded by $\mathcal{O}(2^{2L})$ when $\gamma d < 2$ (with an additional log-factor if $\gamma d = 2$), and by $\mathcal{O}(2^{\gamma d L})$ if $\gamma d > 2$.

The complexity parameter γ is therefore of major importance in the asymptotical work behaviour of the considered quadrature methods. In our case, given the Newton initial guess strategy, the solver preconditioning and the difference in parallel resources used from one level to another do not allow to give this parameter a concise value over all levels (compare Section 3). We suggest however to evaluate the work in terms of total execution time, in which the cost $C_{\mathrm{FE},l}$ for solving a sample at level l is given by the time to solution (averaged over 100 samples). The resulting plot of work comparison between the different methods is reported in Figure 6. On the on hand it is clearly visible that both MLMC and MLQMC show a significantly improved asymptotic efficiency compared to their single level counterpart. On the other hand comparing MLMC with MLQMC shows that MLQMC seems to be able to use the higher convergence order of QMC versus MC to achieve an improved asymptotic efficiency over MLMC.

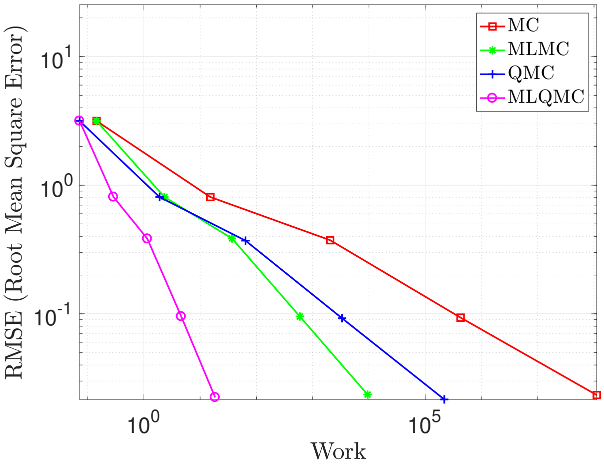


Figure 6: Work comparison for MC/MLMC/QMC/MLQMC. Work is computed on the basis of execution time on a single thread for all levels.

5.2.2 Idealized ventricle

In the second test case, we rely on a mesh hierarchy of $L = 3$ levels. The main reason for the limitation of levels number for this geometry is essentially due to the nestedness condition. Indeed, for this geometry as opposed to the simple cube one, we proceed in an inverted way, i.e. refining a given initial mesh. This procedure becomes demanding at the memory level very quickly since the refinement step increases the degrees of freedom by the factor $2^4 = 16$ due to the space–time discretization. In general, this limitation can very often be encountered when dealing with nested meshes for realistic geometries. This is the main motivation for relying on non-nested meshes for the last test case, see Subsection 5.3.

The number of space-time degrees of freedom (DOF), the space and time discretization steps of the three different levels are reported in Table 2. The meshes are visualized in Figure 7. The rates of convergence and the work of the different quadrature methods are found in Figure 8. As we have only three levels, the meaningfulness of the results is limited as it is impossible to conclude the asymptotic behaviour. Nevertheless, it is clearly seen that MLQMC is superior over the other methods.

l	0	1	2
DOF	154’546	2’120’420	31’184’747
h	0.1	0.05	0.025
Δt	0.02	0.01	0.005

Table 2: Details about the mesh hierarchy for the idealized ventricle geometry.

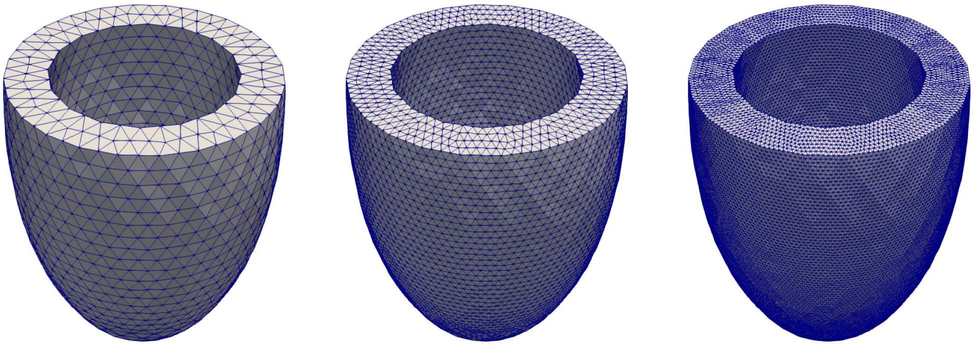


Figure 7: Nested mesh hierarchy for the idealized ventricle.

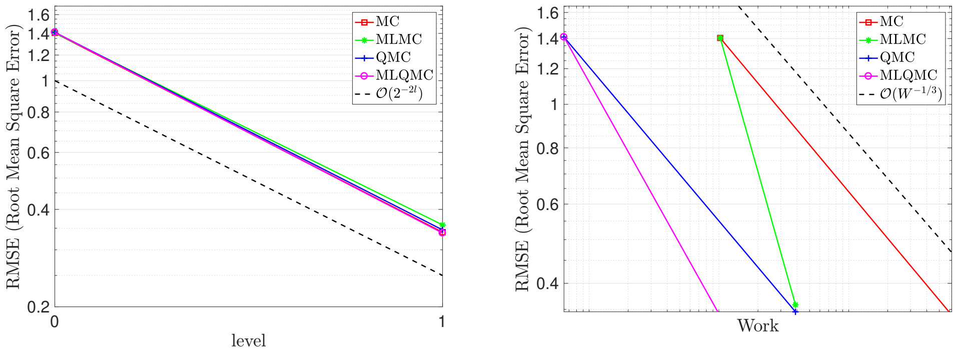


Figure 8: Convergence rate for the idealized ventricle in the L^2 norm (left) and work comparison (right).

5.3 Random fibers in a complex geometry with non-nested meshes

The last test case concerns a realistic heart geometry with data acquired from clinical measurements. As this is meant to be the synthesis of this work, we also account for anisotropic diffusion defined in (3). The associated expected fiber field $\mathbb{E}[\boldsymbol{V}](\boldsymbol{x})$ is shown in Figure 9. It is obtained from a mathematical reconstruction using transmural coordinates [29]. The transmural coordinates are derived by initially solving a diffusion problem with adapted boundary conditions at the contour of the left and right ventricles [2].

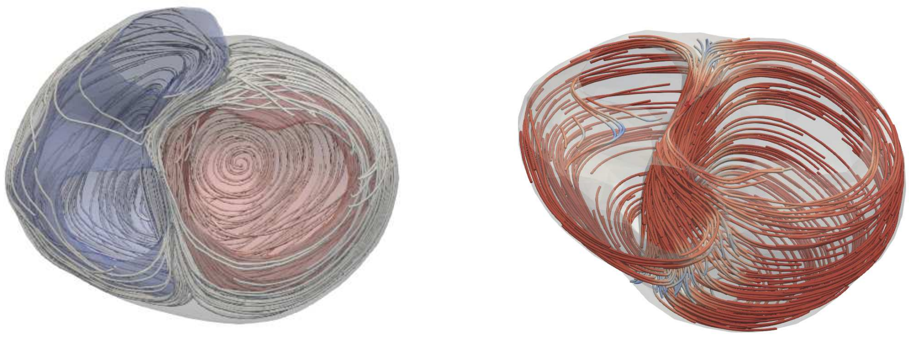


Figure 9: Initial state for fibers $\mathbb{E}[\boldsymbol{V}](\boldsymbol{x})$.

We also relax the nestedness condition by considering a hierarchy of non-nested meshes. As we have previously argued, the nestedness condition very quickly becomes a burden in considering a large number of levels. We rely on a mesh hierarchy with 6 levels in this example. They are shown in Figure 10. The details on the space-time DOF and discretization steps are reported in Table 3.

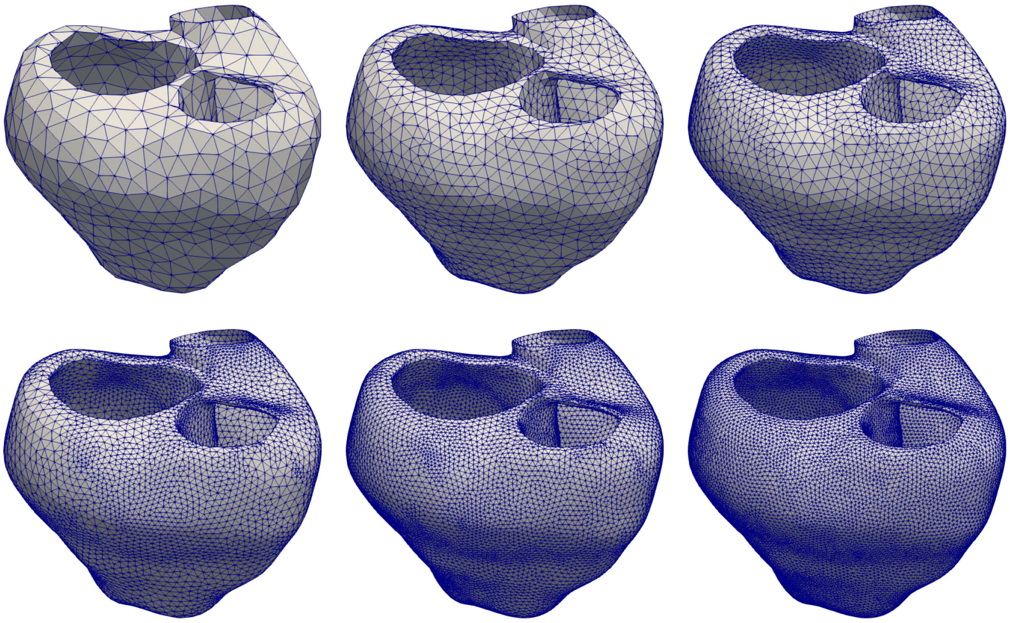


Figure 10: Non-nested mesh hierarchy for the realistic heart geometry.

l	0	1	2	3	4	5
DOF	18'480	113'312	583'104	1'740'800	8'777'728	34'894'848
h	0.16	0.08	0.04	0.03	0.02	0.01
Δt	0.16	0.08	0.04	0.02	0.01	0.005

Table 3: Details about the mesh hierarchy for the realistic heart geometry.

As we do not have nested finite element spaces, we rely here on the multilevel estimator (21). Moreover, we evaluate the convergence for the two quantities of interest, namely the action potential and the activation times for given locations in the domain.

5.3.1 Action potential

We evaluate the evolution of the action potential in several locations of the heart domain. The first example considers a set of points that are placed along the wall separating the left and right ventricles. These points are shown in Figure 11.

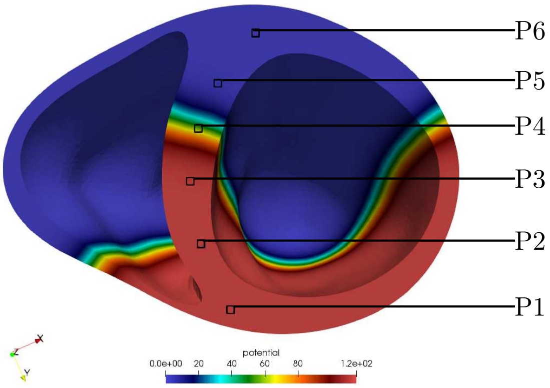
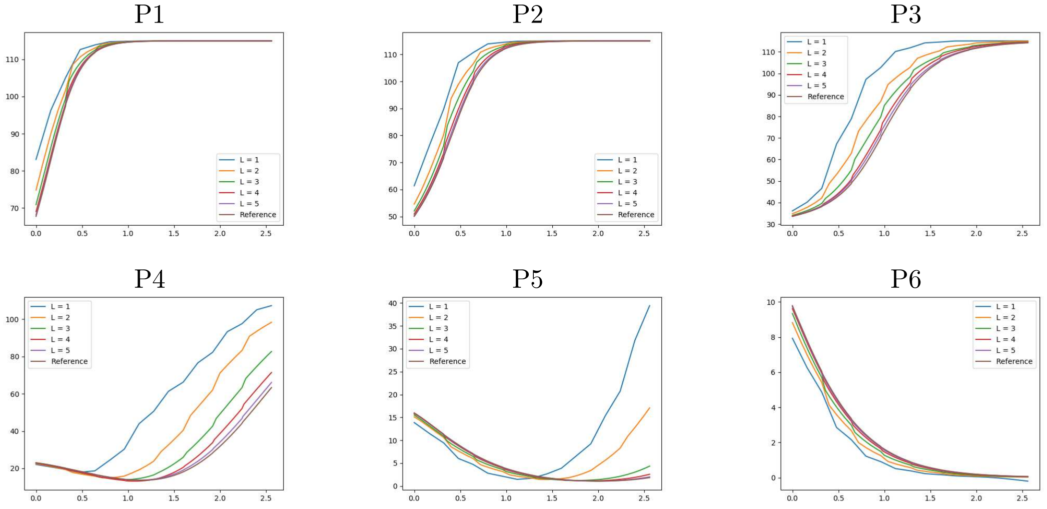


Figure 11: Locations selected along the wall separating the left and right ventricles.

As one can see, these points have been selected such that they trace the behaviour of locations at different distance from the stimulus center, starting from very close (the very below point) to relatively far (the very top point). We report in Figure 12 the action potential obtained by the MC quadrature method for different discretization levels.


P1

P2

P3

P4

P5

P6

Figure 12: Action potential behaviour given different mesh level discretizations for the points specified in Figure 11 following the order going from the bottom to the top.

In Figure 13, we report the convergence graphs of the (pointwise) error (23) for the action potential at the locations introduced in Figure 11 and $q = 1$. Notice that the graphs report the root mean square errors. The expected convergence rate is achieved for all quadrature methods tested.

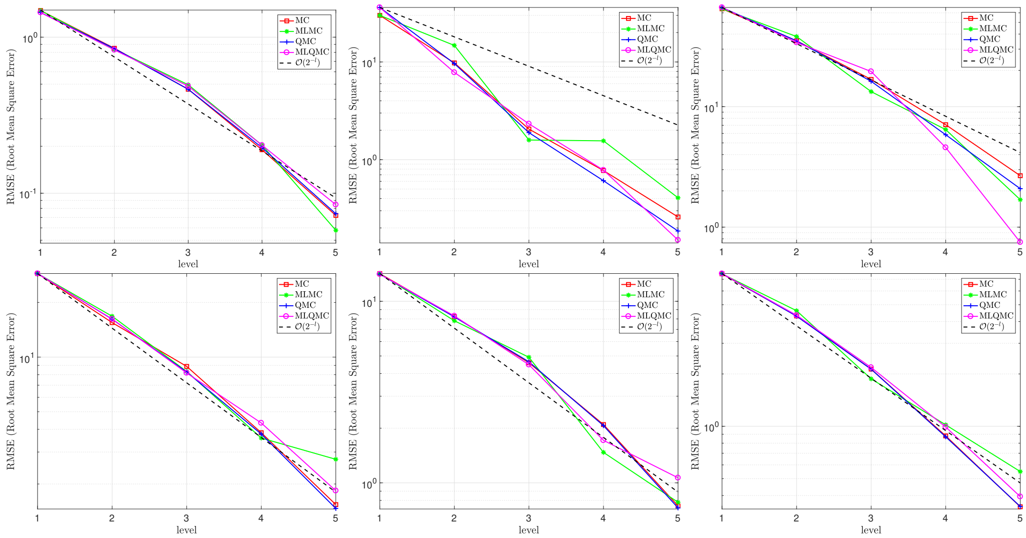


Figure 13: Convergence in the H^1 norm of the action potential at the locations specified in Figure 11 following the order going from the bottom to the top.

The second test is concerned with points located at the circumference of a horizontal cut of the heart surface. These are shown in Figure 14.

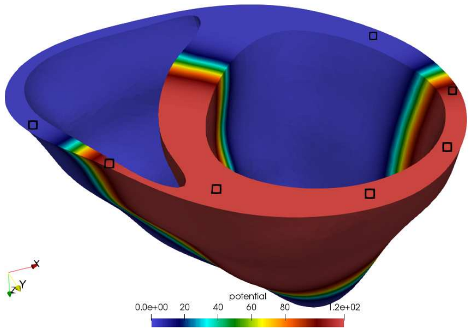


Figure 14: Locations selected at the circumference of a horizontal cut of the heart surface.

Since the behaviour of the action potential at these points follows a similar pattern to that of the previously shown ones, cf. Figure 12, we directly show the graph regarding the convergence of the action potential at these points in Figure 15. Again, we see the expected convergence rate for all quadrature methods tested.

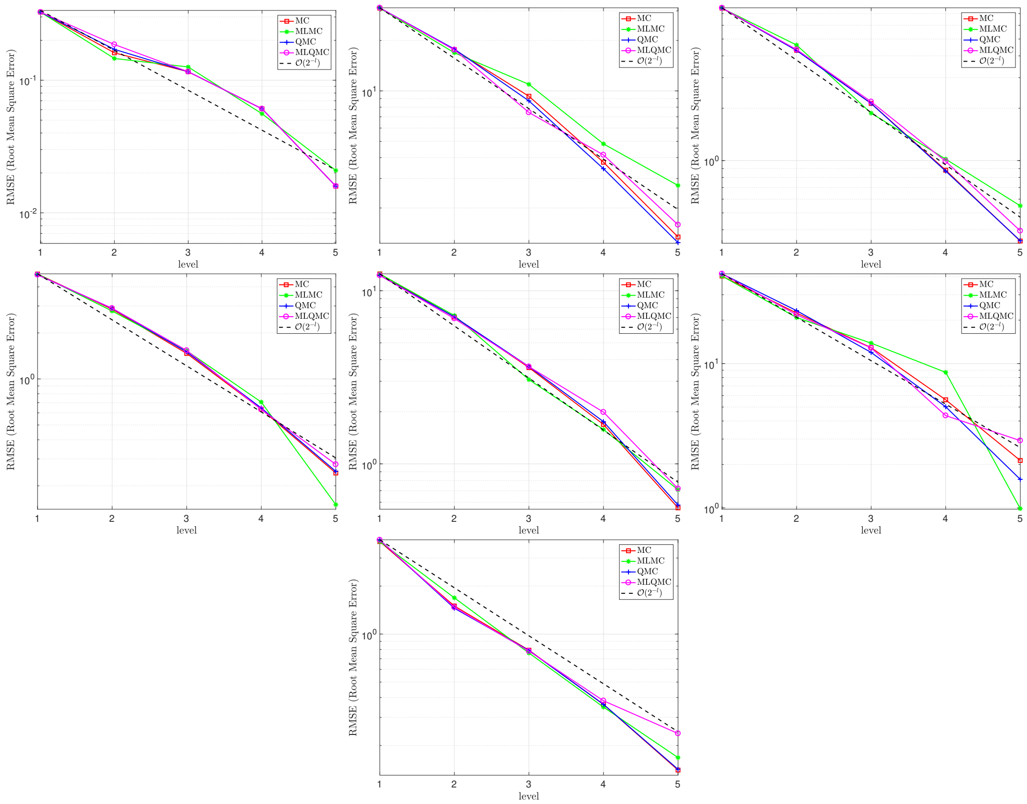


Figure 15: Convergence of the action potential at the locations specified in Figure 14 following the order going from left to right.

5.3.2 Activation time

We start by selecting points at equivalent geodesic distance from the stimulus location. These points are shown in Figure 16. The geodesic distance is calculated by solving an eikonal problem with a zero initial condition on the originating point [28], i.e. the stimulus in our case. The graphs showing the convergence of the activation times for these locations are reported in Figure 17, with all of them showing the expected rate of convergence for all quadrature methods tested.

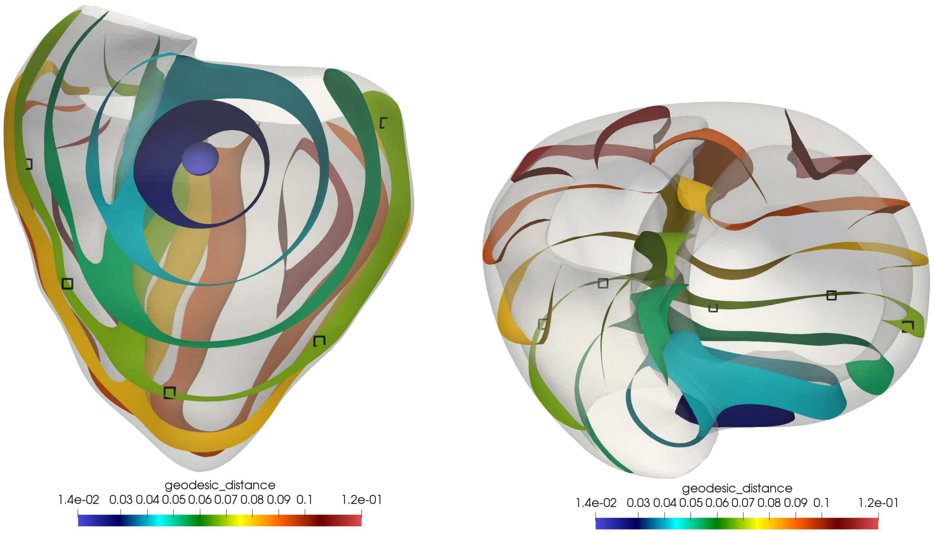

geodesic_distance
1.4e-02 0.03 0.04 0.05 0.06 0.07 0.08 0.09 0.1 1.2e-01
geodesic_distance
1.4e-02 0.03 0.04 0.05 0.06 0.07 0.08 0.09 0.1 1.2e-01

Figure 16: Locations selected at equivalent geodesic distance from the stimulus.

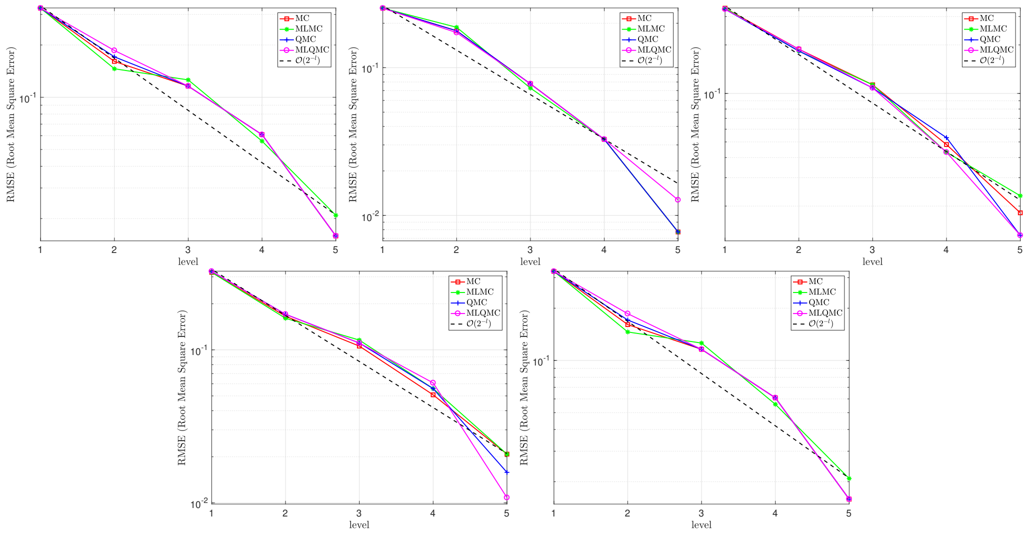


Figure 17: Convergence of the activation time at the locations specified in Figure 16 following the order going from left to right.

We next select locations at the circumference of the left ventricle. These are shown in Figure 18. This electrical signal, when propagated to the chest, is exactly what is perceived clinically (on a electrocardiogram monitor). Mathematically, it is possible to map the surface potential to the chest by solving an additional diffusion problem, see [6, 7]. As we have several discretization levels, we need to ensure that these points are well-defined on each one of them. The convergence graphs for the activation times at these locations, which validate the the expected rate of convergence for all quadrature methods tested, are reported in Figure 19.

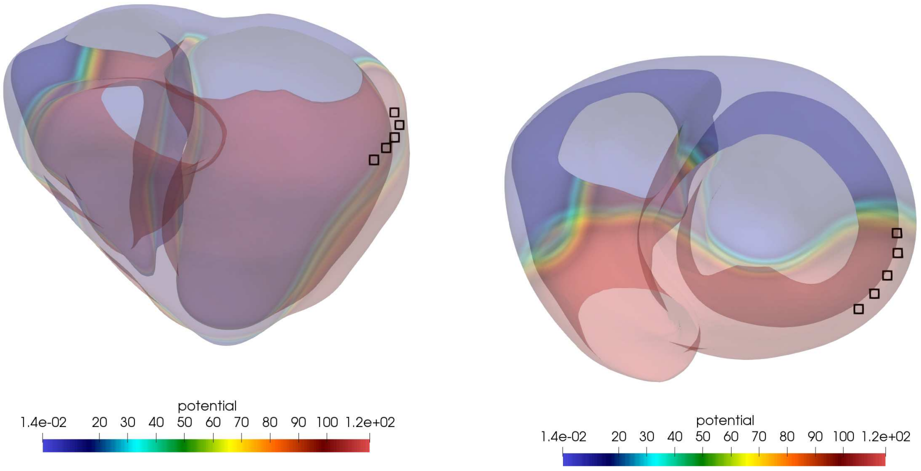


Figure 18: Locations selected at the periphery of the heart surface.

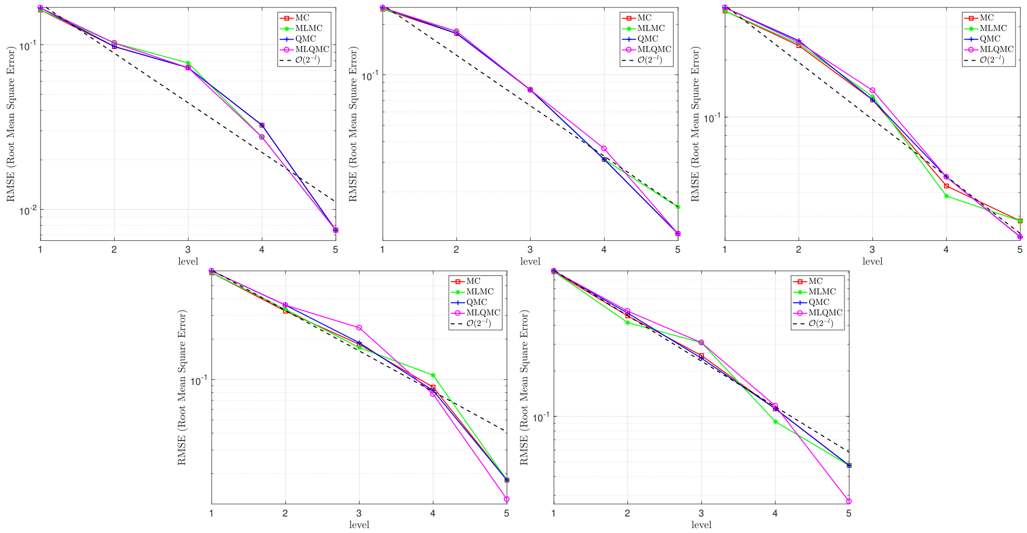


Figure 19: Convergence of the activation time at the locations specified in Figure 18 following an order going from the furthest of the (final state) travelling wave to the nearest.

6 Conclusion

In this article, we have considered the monodomain equation from cardiac electrophysiology with the Fitz-Hugh Nagumo model and an anisotropic conductivity tensor that can account for increased diffusion along the direction of the heart fibers. Modelling the heart fibers as a random vector field, by means of the Karhunen-Loéve expansion from given expectation and covariance vector fields of the heart fibers, we arrive at a parametric monodomain equation. Thus, common quantities of interest such as the action potential and the activation time then also are subject to this uncertainty and we therefore aim at computing their statistics, which amounts to the evaluation of a high-dimensional integral.

To enable the approximate computation of the high-dimensional integral, we propose to combine a space-time discretisation of the monodomain equation using finite elements in space and the Crank–Nicolson method in time, which yields a method with good parallel scalability, in a multilevel manner with dimension robust quadrature methods. The resulting scheme is fully parallelized in space, time and stochastics.

Our numerical experiments show that the approach is feasible and that the considered quadrature methods consistently satisfy their theoretical convergence rates. This indicates that in the settings of our numerical experiments the more restrictive regularity requirements for the QMC quadrature are fulfilled, and that the mixed regularity requirement for the multilevel quadrature methods MLMC and MLQMC are fulfilled, as well. The results show that we can significantly improve the amount of work required for a certain error by using the QMC and MLQMC methods instead of the MC and MLMC methods. It is important to note, that as the QMC and MLQMC quadrature methods using Halton points are essentially a MC or MLMC method where the random sequence of sample points is replaced with fewer Halton points, the increased performance does not require any additional non-trivial implementation modifications to be made when changing from the MC and MLMC methods to the QMC and MLQMC methods.

Lastly, the numerical experiments on the realistic heart geometry additionally show the utility of the multilevel estimator using the quadrature differences instead of solution differences,

$$\mathrm{QoI}_L^{\mathrm{ML}}[u] := \sum_{l=0}^{L} (\mathcal{Q}_l - \mathcal{Q}_{l-1})\big(\mathcal{F}_{L-l}[u](\cdot)\big),$$

combined with the usage of non-nested meshes when considering involved space geometries.

7 Acknowledgements

The authors would like to thank the Swiss National Science Foundation (SNSF) for their support through the project “Multilevel Methods and Uncertainty Quantification in Cardiac Electrophysiology” in collaboration with the University of Basel (grant agreement SNSF-205321_169599). The authors also gratefully acknowledge the support of the Center of Computational Medicine in Cardiology and in particular Dr. Simone Pezzuto for providing the heart geometry and many fruitful discussions.

References

[1] Andrea Barth, Christoph Schwab, and Nathaniel Zollinger. Multi-level Monte Carlo finite element method for elliptic PDEs with stochastic coefficients. *Numerische Mathematik*, 119 (1):123–161, 2011.

[2] Jason D. Bayer, Robert C. Blake, Gernot Plank, and Natalia A Trayanova. A novel rule-based algorithm for assigning myocardial fiber orientation to computational heart models. *Annals of Biomedical Engineering*, 40(10):2243–2254, 2012.

[3] Seif Ben Bader, Pietro Benedusi, Alessio Quaglino, Patrick Zulian, and Rolf Krause. Space-time multilevel monte carlo methods and their application to cardiac electrophysiology. *Journal of Computational Physics*, 433:110164, 2021.

[4] Pietro Benedusi. *Parallel space-time multilevel methods with application to electrophysiology: theory and implementation.* PhD thesis, Università della Svizzera italiana, 2020.

[5] Pietro Benedusi, Carlo Garoni, Rolf Krause, Xiaozhou Li, and Stefano Serra-Capizzano. Space-Time FE-DG Discretization of the Anisotropic Diffusion Equation in Any Dimension: The Spectral Symbol. *SIAM Journal on Matrix Analysis and Applications*, 39(3):1383–1420, 2018.

[6] Muriel Boulakia, Serge Cazeau, Miguel A Fernández, Jean-Frédéric Gerbeau, and Nejib Zemzemi. Mathematical modeling of electrocardiograms: a numerical study. *Annals of Biomedical Engineering*, 38(3):1071–1097, 2010.

[7] Miguel A Fernández and Nejib Zemzemi. Decoupled time-marching schemes in computational cardiac electrophysiology and ecg numerical simulation. *Mathematical biosciences*, 226(1):58–75, 2010.

[8] Richard FitzHugh. Impulses and physiological states in theoretical models of nerve membrane. *Biophysical Journal*, 1(6):445–466, 1961.

[9] Martin J. Gander. 50 years of time parallel time integration. In *Multiple Shooting and Time Domain Decomposition Methods*, pages 69–113. Springer, 2015.

[10] Thomas Gerstner and Stefan Heinz. Dimension- and time-adaptive multilevel Monte Carlo methods. In Jochen Garcke and Michael Griebel, editors, *Sparse Grids and Applications*, volume 88 of *Lecture Notes in Computational Science and Engineering*, pages 107–120, Berlin-Heidelberg, 2012. Springer.

[11] Michael B. Giles. Multilevel Monte Carlo path simulation. *Operations Research*, 56(3): 607–617, 2008.

[12] Michael B. Giles. Multilevel Monte Carlo methods. *Acta Numerica*, 24:259–328, 2015.

[13] Michael Griebel and Helmut Harbrecht. A note on the construction of L-fold sparse tensor product spaces. *Constructive Approximation*, 38(2):235–251, 2013.

[14] Michael Griebel, Helmut Harbrecht, and Michael D. Multerer. Multilevel quadrature for elliptic parametric partial differential equations in case of polygonal approximations of curved domains. *SIAM Journal on Numerical Analysis*, 58(1):684–705, 2020.

[15] Helmut Harbrecht and Marc Schmidlin. Multilevel methods for uncertainty quantification of elliptic PDEs with random anisotropic diffusion. *Stochastic Partial Differential Equations: Analysis and Computations*, 8(1):54–81, 2020.

[16] Helmut Harbrecht, Michael Peters, and Reinhold Schneider. On the low-rank approximation by the pivoted Cholesky decomposition. *Applied Numerical Mathematics*, 62:28–440, 2012.

[17] Helmut Harbrecht, Michael Peters, and Markus Siebenmorgen. On multilevel quadrature for elliptic stochastic partial differential equations. In Jochen Garcke and Michael Griebel, editors, *Sparse Grids and Applications*, volume 88 of *Lecture Notes in Computational Science and Engineering*, pages 161–179. Springer, Berlin-Heidelberg, 2012.

[18] Helmut Harbrecht, Michael Peters, and Markus Siebenmorgen. Efficient approximation of random fields for numerical applications. *Numerical Linear Algebra with Applications*, 22 (4):596–617, 2015.

[19] Helmut Harbrecht, Michael Peters, and Marc Schmidlin. Uncertainty quantification for PDEs with anisotropic random diffusion. *SIAM J. Numer. Anal.*, 55(2):1002–1023, 2017.

[20] Stefan Heinrich. The multilevel method of dependent tests. In *Advances in stochastic simulation methods (St. Petersburg, 1998)*, Statistics for Industry and Technology, pages 47–61. Birkhäuser, Boston, MA, 2000.

[21] Lukas Herrmann and Christoph Schwab. Multilevel quasi-Monte Carlo Uncertainty Quantification for advection-diffusion-reaction. In *Monte Carlo and Quasi-Monte Carlo Methods. MCQMC 2018, Rennes, France, July 1–6*, volume 324 of *Springer Proceedings in Mathematics & Statistics*, pages 31–67. Springer Nature Switzerland, Cham, Switzerland, 2020.

[22] E. Hille and R. S. Phillips. *Functional Analysis and Semi-Groups*, volume 31. American Mathematical Society, Providence, 1957.

[23] Alan L. Hodgkin and Andrew F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology*, 117 (4):500–544, 1952.

[24] Daniel E. Hurtado and Duvan Henao. Gradient flows and variational principles for cardiac electrophysiology: toward efficient and robust numerical simulations of the electrical activity of the heart. *Computer Methods in Applied Mechanics and Engineering*, 273:238–254, 2014.

[25] William A. Light and Elliott W. Cheney. *Approximation Theory in Tensor Product Spaces*, volume 1169 of *Lecture Notes in Mathematics*. Springer, New York, 1985.

[26] Eleanor McDonald and Andy Wathen. A simple proposal for parallel computation over time of an evolutionary process with implicit time stepping. In *Numerical Mathematics and Advanced Applications ENUMATH 2015*, pages 285–293. Springer, 2016.

[27] Walter T. Miller and David B. Geselowitz. Simulation studies of the electrocardiogram. i. the normal heart. *Circulation Research*, 43(2):301–315, 1978.

[28] Simone Pezzuto, Alessio Quaglino, and Mark Potse. On sampling spatially-correlated random fields for complex geometries. In Yves et al. Coudière, editor, *International Conference on Functional Imaging and Modeling of the Heart*, pages 103–111, Cham, Switzerland, 2019. Springer Nature Switzerland.

[29] Mark Potse, Bruno Dubé, Jacques Richer, Alain Vinet, and Ramesh M Gulrajani. A comparison of monodomain and bidomain reaction-diffusion models for action potential propagation in the human heart. *IEEE Transactions on Biomedical Engineering*, 53(12):2425–2435, 2006.

[30] Alessio Quaglino, Seif Ben Bader, and Rolf Krause. Sloth: A Python library for UQ. Software, 2017.

[31] Marc Schmidlin. *Multilevel Quadrature Methods and Regularity Analysis for Semilinear PDEs with Random Data*. PhD Thesis, Faculty of Science, University of Basel, 2021. (In preparation).

[32] Markus Siebenmorgen. *Quadrature methods for elliptic PDEs with random diffusion*. PhD thesis, Faculty of Science, University of Basel, 2015.

[33] Patrick Zulian, Alena Kopaničáková, Maria Chiara Giuseppina Nestola, Andreas Fink, Nur Fadel, Victor Magri, Teseo Schneider, Eric Botter, Jan Mankau, and Rolf Krause. Utopia: A C++ embedded domain specific language for scientific computing. Software, 2016. URL `https://bitbucket.org/zulianp/utopia`.