Isogeometric analysis of diffusion problems on random surfaces

Wei Huang [*], Michael Multerer

Euler Institute, Università della Svizzera italiana, Via alla Santa 1, 6962 Lugano, Switzerland

A B S T R A C T

In this article, we discuss the numerical solution of diffusion equations on random surfaces within the isogeometric framework. We describe in detail, how diffusion problems on random surfaces can be modelled and how quantities of interest may be derived. In particular, we employ a low rank approximation algorithm for the high-dimensional space-time correlation of the random solution based on an online singular value decomposition, cp. [7]. Extensive numerical studies are performed to validate the approach. In particular, we consider complex computational geometries originating from surface triangulations. The latter can be recast into the isogeometric context by transforming them into quadrangulations using the procedure from [41] and a subsequent approximation by NURBS surfaces.

© 2022 The Author(s). Published by Elsevier B.V. on behalf of IMACS. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

1. Introduction

Many problems in science and engineering can be modelled by partial differential equations (PDEs). Especially, PDEs on surfaces appear in a variety of applications, such as computer graphics [14,44,46], chemical engineering [1,21,48] or biology [22,23,39] and there already exists a multitude of computational methods for their numerical solution. In [19,20], the surface finite element method has been developed aiming at explicit surface representations by means of meshes. Alternatively, a level set based PDE solver is discussed in [4,5] and the closest point method, addressing closest point representations of surfaces, is proposed in [36,37,42]. A general solver for PDEs on point clouds can be found in [33]. Due to the wide-spread use and the effectiveness of Non-Uniform Rational Basis Spline (NURBS) based surfaces in industry, an Isogeometric Analysis (IGA), see [13,30], based framework for the solution of PDEs on surfaces was introduced in [31].

The striking advantage of IGA over the classical finite element method is the use of an exact geometry representation. In particular, the same basis is chosen to represent the geometry and the solution. This way, in IGA, the apriori L^2-error and H^1-error satisfy the optimal convergence rates $\mathcal{O}(h^{p+1})$ and $\mathcal{O}(h^p)$ respectively for elliptic PDEs on surfaces, cf. [2,15,45]. Herein, h and p are the mesh size and the polynomial order of the NURBS basis functions respectively. The L^2-error for the IGA of the parabolic problems is explored in [49]. In this article, we shall adopt the IGA framework for solving diffusion PDEs on random surfaces.

As we have outlined, PDEs on deterministic surfaces are already well explored. In many applications, however, such as chemical engineering and biology, the computational domain can not be measured exactly or may be subject to manufacturing tolerances. Indeed, within this article, we will employ an interpolation approach to recast arbitrary surface triangulations into the isogeometric framework, which might itself be considered a source of epistemic uncertainty. In such situations, the

---

* Corresponding author.
E-mail addresses: wei.huang@usi.ch (W. Huang), michael.multerer@usi.ch (M. Multerer).
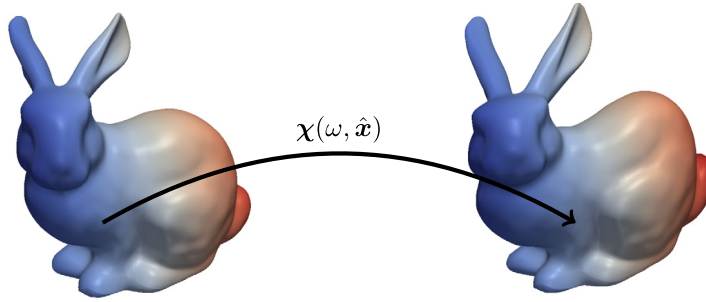
Fig. 1. Realization of a randomly deformed surface.

computational domain is a major source of uncertainty that needs to be accounted for. Uncertainty quantification of PDEs on random domains and surfaces has previously been considered in [11,16,26,47] and in the context of IGA in [3,17]. To model random surfaces, we follow the approach presented in [26] based on the Karhunen-Loève expansion of random deformation fields and exploit that the knowledge of this field at the surface is sufficient for the modelling of the random surface itself, cp. [17]. Having the random deformation field at our disposal, quantities of interest (QoI), such as the expectation and the correlation, can be defined with respect to the reference configuration. As the computation of the expectation is straightforward, we will particularly focus here on the computation of the space-time correlation, whose size may easily exceed the memory of a modern computer. Therefore, we devise a suitable low rank approximation, based on the online singular value decomposition suggested in [7]. We remark that similar low rank approximation methods have already been studied in the context of reduced basis methods, see [12,32,43]. However, in contrast to these works, our goal is not to devise a surrogate model for the underlying PDE, but rather to find an efficient means to represent the correlation.

The rest of this article is structured as follows. In Section 2, we introduce the diffusion problem and the random model under consideration. In Section 3, we discuss the isogeometric representation of random domains and briefly recall the algorithm from [41] for transferring surface triangulations into quadrangulations. Section 4 then discusses the variational formulation. Especially, thanks to the isogeometric setting we can compute realizations of the solution directly in the spatial configuration. Section 4 is devoted to the computation of quantities of interest, particularly to the low rank approximation of the solution's correlation. In Section 6, extensive numerical studies are presented and concluding remarks are stated in Section 7.

2. Problem formulation

Let $(\Omega, \mathcal{F}, \mathbb{P})$ denote a complete probability space, where Ω is the sample space, $\mathcal{F}$ denotes the σ-field of events, and $\mathbb{P}$ is a probability measure. As model problem, we consider the diffusion equation on a closed random surface, i.e.,

$$\begin{cases} \partial_t u(\omega, \boldsymbol{x}, t) - \Delta_S u(\omega, \boldsymbol{x}, t) = f(\omega, \boldsymbol{x}), & \boldsymbol{x} \in S(\omega) \\ u(\omega, \boldsymbol{x}, 0) = u_0(\omega, \boldsymbol{x}), & \boldsymbol{x} \in S(\omega) \end{cases} \quad \text{for } \mathbb{P}\text{-a.e. } \omega \in \Omega. \tag{1}$$

Herein, Δ_S denotes the Laplace-Beltrami operator, f is a source on the surface, and u_0 denotes the initial condition. As the surface is closed, we do not need to impose any boundary conditions. Uniqueness of the solution is then obtained by considering mean zero functions. For the treatment of non-trivial boundary conditions in the random domain case, we refer to [25].

For the modelling of the random surface, we assume the existence of a Lipschitz continuous reference surface S_0 and a random deformation field

$$\boldsymbol{\chi} : \Omega \times S_0 \to \mathbb{R}^3$$

such that there exists a constant $C_{\mathrm{uni}} > 0$ with

$$\|\boldsymbol{\chi}(\omega)\|_{C^1(S_0;\mathbb{R}^3)}, \|\boldsymbol{\chi}^{-1}(\omega)\|_{C^1(S(\omega);\mathbb{R}^3)} \le C_{\mathrm{uni}} \quad \text{for } \mathbb{P}\text{-a.e. } \omega \in \Omega \tag{2}$$

and

$$S(\omega) = \boldsymbol{\chi}(\omega, S_0) \quad \text{for } \mathbb{P}\text{-a.e. } \omega \in \Omega.$$

We refer to Fig. 1 for a realization of such a random deformation field. Note that it would in principle also be possible to consider random space-time deformation fields, as they have recently been introduced in [24], in a similar fashion.

Remark 2.1. For a sufficiently smooth surface, e.g. Lipschitz continuous, the uniformity condition (2) guarantees that the Sobolev spaces $H^1(S(\omega))$ and $H^1(S_0)$ are equivalent independently of the particular realization of the random parameter,

see for example [26]. A more general framework is considered in [16], where the bounds on the deformation field hold only pathwise. The latter particularly covers the situation of log-normal deformation fields.

For our model (1), we make the assumption that the initial condition and the heat source on the random surface are given in material coordinates and not subject to uncertainty themselves, i.e.,

$$u_0(\omega, \boldsymbol{x}) := \hat{u}_0\big(\boldsymbol{\chi}^{-1}(\omega, \boldsymbol{x})\big), \quad f(\omega, \boldsymbol{x}) := \hat{f}\big(\boldsymbol{\chi}^{-1}(\omega, \boldsymbol{x})\big)$$

for two deterministic functions $\hat{f}, \hat{u}_0 \colon S_0 \to \mathbb{R}$. We remark that it is also possible to consider initial conditions and heat sources described in spatial coordinates. In this case, they need to be defined on the hold-all domain $\bigcup_{\omega \in \Omega} \boldsymbol{\chi}(\omega, S_0)$. Furthermore, one may to consider the situation where the data are subject to uncertainty themselves, which typically results in a much higher dimensionality of the problem, see e.g. [38]. Finally, for a given solution u on $S(\omega)$, we denote the pulled back solution by

$$\hat{u}(\omega, \hat{\boldsymbol{x}}, t) := u\big(\boldsymbol{\chi}(\omega, \hat{\boldsymbol{x}}), t\big).$$

Based on the pulled back solution, we compute QoI's, such as the expected solution and its space-time correlation. More precisely, for any fixed time points t and t', we define these quantities according to

$$\mathbb{E}[\hat{u}(\omega, \hat{\boldsymbol{x}}, t)] = \int_{\Omega} \hat{u}(\omega, \hat{\boldsymbol{x}}, t) \mathrm{d}\mathbb{P}(\omega) \in H^1(S_0) \tag{3}$$

and

$$\mathrm{Cor}[\hat{u}]\big((\hat{\boldsymbol{x}}, t), (\hat{\boldsymbol{x}}', t')\big) = \int_{\Omega} \hat{u}(\omega, \hat{\boldsymbol{x}}, t)\hat{u}(\omega, \hat{\boldsymbol{x}}', t') \mathrm{d}\mathbb{P}(\omega) \in H^1(S_0) \otimes H^1(S_0), \tag{4}$$

respectively.

3. Isogeometric representation of random domains

3.1. Representation of random deformation fields

In this section we briefly recall the ideas presented in [17] and adapt them to our settings. As has been argued there, it is sufficient to compute the Karhunen-Loève expansion of the random deformation field exclusively with respect to the random surface and no volume discretization is required at all. Hence, given the expected deformation field

$$\mathbb{E}[\boldsymbol{\chi}] \colon S_0 \to \mathbb{R}^d$$

and its matrix valued covariance function

$$\mathrm{Cov}[\boldsymbol{\chi}] \colon S_0 \times S_0 \to \mathbb{R}^{d \times d},$$

we can compute the surface Karhunen-Loève expansion

$$\boldsymbol{\chi}(\omega, \hat{\boldsymbol{x}}) = \mathbb{E}[\boldsymbol{\chi}](\hat{\boldsymbol{x}}) + \sum_{k=1}^{\infty} \sqrt{\lambda_k} \boldsymbol{\chi}_k(\hat{\boldsymbol{x}}) Y_k(\omega), \quad \hat{\boldsymbol{x}} \in S_0. \tag{5}$$

Herein, the tuples $\{(\lambda_k, \boldsymbol{\chi}_k)\}_k$ are the eigenpairs of the covariance operator

$$\mathcal{C} \colon [L^2(S_0)]^3 \to [L^2(S_0)]^3,$$
$$(\mathcal{C}\boldsymbol{v})(\hat{\boldsymbol{x}}) := \int_{S_0} \mathrm{Cov}[\boldsymbol{\chi}](\hat{\boldsymbol{x}}, \hat{\boldsymbol{x}}') \boldsymbol{v}(\hat{\boldsymbol{x}}') \mathrm{d}\sigma_{\hat{\boldsymbol{x}}'} \tag{6}$$

and, for $\lambda_k \neq 0$, the centred and uncorrelated random variables $\{Y_k\}_k$ are given according to

$$Y_k(\omega) := \frac{1}{\sqrt{\lambda_k}} \int_{S_0} \big(\boldsymbol{\chi}(\omega, \hat{\boldsymbol{x}}) - \mathbb{E}[\boldsymbol{\chi}](\hat{\boldsymbol{x}})\big)^{\mathsf{T}} \boldsymbol{\chi}_k(\hat{\boldsymbol{x}}) \mathrm{d}\sigma_{\hat{\boldsymbol{x}}}.$$

The existence of the Karhunen-Loève expansion is guaranteed by the uniformity condition (2), which ensures $\boldsymbol{\chi} \in L^2\big(\Omega; [L^2(S_0)]^3\big)$ and $\mathrm{Cov}[\boldsymbol{\chi}] \in [L^2(S_0)]^3 \otimes [L^2(S_0)]^3$ and thus gives rise to the Hilbert-Schmidt operator $\mathcal{C}$ from (6), see also [26]. For the numerical realization of the Karhunen-Loève expansion in the random domain case, we refer to [27,38].

In practice, however, the random variables $\{Y_k\}_k$ are not known explicitly and need to be estimated. We make the common model assumption that the random variables $\{Y_k\}_k$ are independent and uniformly distributed with $\mathcal{U}(-1, 1)$ for all k. For numerical computations, the Karhunen-Loève expansion has to be truncated after $m \in \mathbb{N}$ terms, where m has to be chosen to meet the overall accuracy. Then, by identifying each random variable Y_k by its image $y_k \in [-1, 1]$, we arrive at the parametric deformation field

$$\boldsymbol{\chi}(\boldsymbol{y}, \hat{\boldsymbol{x}}) = \mathbb{E}[\boldsymbol{\chi}](\hat{\boldsymbol{x}}) + \sum_{k=1}^{m} \sqrt{\lambda_k} \boldsymbol{\chi}_k(\hat{\boldsymbol{x}}) y_k, \quad \boldsymbol{y} \in \Gamma := [-1, 1]^m. \tag{7}$$

The parametric deformation field gives rise to the parametric surfaces

$$S(\boldsymbol{y}) = \{\boldsymbol{\chi}(\boldsymbol{y}, \hat{\boldsymbol{x}}) : \hat{\boldsymbol{x}} \in S_0\}. \tag{8}$$

In the next subsection, we define corresponding NURBS representations and the associated ansatz spaces.

3.2. Random NURBS surfaces

We start by recalling the basic notions of isogeometric analysis, restricting ourselves to function spaces constructed via locally quasi-uniform p-open knot vectors as in [8].

Definition 3.1. Let p and k with $0 \le p < k$. A *locally quasi uniform p-open knot vector* is a tuple

$$\Xi = \left[\xi_0 = \cdots = \xi_p \le \cdots \le \xi_k = \cdots = \xi_{k+p}\right] \in [0, 1]^{k+p+1}$$

with $\xi_0 = 0$ and $\xi_{k+p} = 1$ such that there exists a constant $\theta \ge 1$ with $\theta^{-1} \le h_j \cdot h_{j+1}^{-1} \le \theta$ for all $p \le j < k$, where $h_j := \xi_{j+1} - \xi_j$. The B-spline basis $\{b_j^p\}_{0 \le j < k}$ is then recursively defined according to

$$b_j^p(x) = \begin{cases} \mathbb{1}_{[\xi_j, \xi_{j+1})} & \text{if } p = 0, \\ \frac{x - \xi_j}{\xi_{j+p} - \xi_j} b_j^{p-1}(x) + \frac{\xi_{j+p+1} - x}{\xi_{j+p+1} - \xi_{j+1}} b_{j+1}^{p-1}(x) & \text{else,} \end{cases}$$

where $\mathbb{1}_A$ refers to the characteristic function of the set A. The corresponding spline space is defined as $\mathcal{S}^p(\Xi) := \operatorname{span}(\{b_j^p\}_{j<k})$.

Spline spaces in two spatial dimensions are obtained by a tensor product construction. More precisely, we define the spaces

$$\mathcal{S}^{\boldsymbol{p}} := \mathcal{S}^p(\Xi) \otimes \mathcal{S}^p(\Xi). \tag{9}$$

With regard to the knot vector Ξ, sets of the form $[\xi_i, \xi_{i+1}] \times [\xi_j, \xi_{j+1}]$ are referred to as *elements*. The corresponding mesh size will be denoted by h. For further details, we refer to [40] and the references therein.

In what follows, we shall adopt the usual isogeometric setting for the representation of the surface S_0, i.e., we assume that S_0 can be decomposed into several smooth *patches* $S_0 = \bigcup_{i=1}^{M} S_0^{(i)}$, where the intersection $S_0^{(i)} \cap S_0^{(i')}$ consists at most of a common vertex or a common edge for $i \ne i'$. Each patch $S_0^{(i)}$ is represented by an invertible NURBS mapping

$$\boldsymbol{s}_i : [0, 1]^2 \to S_0^{(i)} \quad \text{with} \quad S_0^{(i)} = \boldsymbol{s}_i([0, 1]^2) \quad \text{for } i = 1, 2, \ldots, M. \tag{10}$$

Herein, the functions $\boldsymbol{s}_i$ are of the form

$$\boldsymbol{s}_i(x_1, x_2) := \sum_{0=i_1}^{k_1} \sum_{0=i_2}^{k_2} \frac{\boldsymbol{c}_{i_1, i_2} b_{i_1}^p(x_1) b_{i_2}^p(x_2) w_{i_1, i_2}}{\sum_{j_1=0}^{k_1-1} \sum_{j_2=0}^{k_2-1} b_{j_1}^p(x_1) b_{j_2}^p(x_2) w_{j_1, j_2}}$$

for control points $\boldsymbol{c}_{i_1, i_2} \in \mathbb{R}^3$ and weights $w_{i_1, i_2} > 0$. As a consequence, the resulting patches $S_0^{(i)}$ are at most of class C^{p-1}. In order to obtain a consistent surface representation, we moreover assume that parametrizations sharing a common edge coincide at this edge except for orientation.

The subdivision of the surface S_0 into patches, directly induces a corresponding subdivision of the random surface according to $S(\boldsymbol{y}) = \bigcup_{i=1}^{M} S^{(i)}(\boldsymbol{y})$ with $S^{(i)}(\boldsymbol{y}) := (\boldsymbol{\chi}(\boldsymbol{y}) \circ \boldsymbol{s}_i)([0, 1]^2)$. By representing the mappings $\boldsymbol{\chi}(\boldsymbol{y}) \circ \boldsymbol{s}_i$ using NURBS, it can be achieved that $S^{(i)}(\boldsymbol{y})$ is again a NURBS surface, see [17]. As we typically have to compute many realizations of the parametric surface to obtain acceptable approximations of the QoI's, it is in practice advisable to reinterpolate each patch and to compute the Karhunen-Loève expansion only with respect to a set of interpolation points on $[0, 1]^2$, see [27].
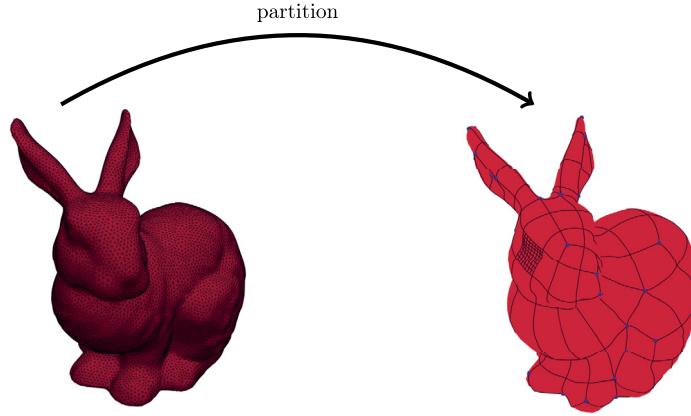
Fig. 2. Partitioning of the Stanford bunny into 179 quadrilateral patches.

As usual, we define the ansatz space on S_0 by patchwise lifting the space $\mathcal{S}^{\boldsymbol{p}}$ to S_0, i.e., $V_h := \big\{ v \in C(S_0) : \big(v|_{S_0^{(i)}} \circ \boldsymbol{s}_i \big) \in \mathcal{S}^{\boldsymbol{p}} \big\}$. Analogously, we introduce the parametric spaces $V_h(\boldsymbol{y}) := \big\{ v \in C\big(S(\boldsymbol{y})\big) : \big(v|_{S_0^{(i)}} \circ \boldsymbol{\chi}(\boldsymbol{y}) \circ \boldsymbol{s}_i \big) \in \mathcal{S}^{\boldsymbol{p}}(\boldsymbol{\Xi}) \big\}$. Given a B-spline basis $\widehat{\boldsymbol{\Phi}} := [\hat{\varphi}_1, \hat{\varphi}_2, \dots, \hat{\varphi}_{N_{\boldsymbol{p}, \boldsymbol{\Xi}}}]$ of V_h, we denote the lifted basis in $V_h(\boldsymbol{y})$ by

$$\varphi_i := (\hat{\varphi}_i \circ \boldsymbol{\chi}^{-1}) \quad \text{for } i = 1, 2, \dots, N_{\boldsymbol{p}, \boldsymbol{\Xi}}.$$

3.3. Partitioning surfaces into quadrilateral patches

In practice, the surface representation (10) is often not directly available for more complex geometries. However, the partitioning of surface meshes into conforming quadrilateral patches is well solved, see for example [6,10,34,35,41] and the references therein. The procedure suggested there is based on determining a suitable set of nodes called *singularities* and connecting these by a set of arcs called *separatrices*, such that the resulting partition consists of multiple valid conforming quadrilateral patches.

The initial quadrangulation for the Stanford bunny considered in this paper has been taken from [41]. The procedure there is described as follows: The input is a triangular surface mesh of the geometry endowed with a *cross field*, which consists of two orthogonal vectors lying on tangent space of each triangle. The singularities are determined by localizing points where no local parametrization can be constructed, see the nodes in Fig. 2. The separatrices are propagated by tracing geodesic paths directly on the triangular mesh starting from a singularity towards an incident singularity. By constraining the valence of all singularities to be neither 3 or 5 and avoiding tangential crossings, the separatrices automatically divide the surface into a conforming quad layout, see Fig. 2. Having quadrangulation and patches described in terms of separatrices at our disposal, we finally fit a NURBS surface by using the partitioned quadrangulation as control mesh for each patch. Because the control points match on the edges between any two incident quad patches, the parametric surface is C^0 globally and C^{p-1} patchwise, where p is the order of the NURBS representation.

4. Variational formulation

For the discretization of the model problem, we employ the method of lines. Hence, for fixed $t > 0$, the discrete variational formulation of the diffusion problem on random surfaces reads:

find $u_h(\boldsymbol{y}) \in V_h(\boldsymbol{y})$ such that

$$\int\limits_{S(\boldsymbol{y})} \frac{\partial u_h(\boldsymbol{y})}{\partial t} v_h(\boldsymbol{y}) \mathrm{d}\sigma + \int\limits_{S(\boldsymbol{y})} \langle \nabla_S u_h(\boldsymbol{y}), \nabla_S v_h(\boldsymbol{y}) \rangle \mathrm{d}\sigma = \int\limits_{S(\boldsymbol{y})} f(\boldsymbol{y}) v_h(\boldsymbol{y}) \mathrm{d}\sigma$$

for all test functions $v_h(\boldsymbol{y}) \in V_h(\boldsymbol{y})$. Note that we have omitted here the dependency of u_h on $\boldsymbol{x}$ and t. Now, inserting the basis representations of $u_h(\boldsymbol{y})$ and $v_h(\boldsymbol{y})$, respectively, this translates to the linear system

$$\boldsymbol{M}(\boldsymbol{y}) \frac{\partial}{\partial t} \boldsymbol{u}(\boldsymbol{y}, \boldsymbol{x}, t) + \boldsymbol{A}(\boldsymbol{y}) \boldsymbol{u}(\boldsymbol{y}, \boldsymbol{x}, t) = \boldsymbol{f}(\boldsymbol{x}, t), \quad \boldsymbol{u}(\boldsymbol{y}, \boldsymbol{x}, 0) = \boldsymbol{u}_0.$$

Herein,

$$\boldsymbol{M}(\boldsymbol{y}) = [(\varphi_i, \varphi_j)_{L^2(S(\boldsymbol{y}))}]_{i,j}$$

denotes the finite element mass matrix, while

$$\boldsymbol{A}(\boldsymbol{y}) = [(\nabla_S \varphi_i, \nabla_S \varphi_j)_{L^2(S(\boldsymbol{y}))}]_{i,j}$$

is the finite element stiffness matrix. Note that the mass matrix and the stiffness matrix are independent of the time, however, they depend on the random parameter $\boldsymbol{y}$. Since we assume that the initial condition and the source term are given in material coordinates, their coefficients $\boldsymbol{f}(t)$ and $\boldsymbol{u}_0$ are independent of the random parameter $\boldsymbol{y}$. Moreover, we remark that, due to the definition of the spline basis on the randomly deformed domain, the coefficient vector $\boldsymbol{u}$ in material and spatial coordinates coincides.

Next, we introduce an appropriate discretization for the time: we employ the θ-scheme with a uniform time discretization of step size Δt. Thus, we obtain the linear system

$$(\boldsymbol{M} + \Delta t \theta \boldsymbol{A})\boldsymbol{u}(t_{i+1}) = (\boldsymbol{M} - \Delta t(1 - \theta)\boldsymbol{A})\boldsymbol{u}(t_i) + \Delta t(1 - \theta)\boldsymbol{f}(t_i) + \Delta t \theta \boldsymbol{f}(t_{j+1}),$$

where we have dropped the parameter dependency for a more compact notation.

We close this paragraph by citing the corresponding error estimate. To this end, we recall that the surface S_0 consists of M smooth patches. A uniform mesh is thus obtained by uniform refinement of the parameter domain $[0,1]^2$ for each patch and lifting the subdivided parameter domain to the surface. Hence, subdividing each patch j times yields to a mesh of mesh size $h \sim 2^{-j}$. For fixed $\boldsymbol{y}$, we obtain the following convergence result from [49] under the IGA-θ stability condition introduced therein.

Theorem 4.1. *For any fixed $\boldsymbol{y} \in [-1,1]^m$, let $T > 0$, $f \in C^0([0,T]; L^2(S(\boldsymbol{y})))$, $\frac{\partial f}{\partial t} \in L^2(S(\boldsymbol{y}))$, $u(0,\boldsymbol{x}) \in H^{p+1}(S(\boldsymbol{y}))$ and the solution u is such that*

$$u \in C^0\left([0,T]; H^{p+1}(S(\boldsymbol{y}))\right) \text{ and } \frac{\partial u}{\partial t} \in L^1\left([0,T]; H^{p+1}(S(\boldsymbol{y}))\right).$$

The approximation $u_h(t_i)$ satisfies $\frac{\partial u_h}{\partial t}(0) \in L^2(S(\boldsymbol{y}))$. Then, the pointwise L^2-error for the θ-scheme satisfies

$$
\begin{aligned}
&\|u(t_i) - u_h(t_i)\|_{L^2(S(\boldsymbol{y}))} \\
&\leq C_\theta \bigg[h^{p+1} \bigg(|u(0)|_{H^{p+1}(S(\boldsymbol{y}))} + \left\| \frac{\partial u}{\partial t} \right\|_{L^1([0,T]; H^{p+1}(S(\boldsymbol{y})))} \bigg) \\
&\quad + \Delta t \bigg(\left\| \frac{\partial u_h}{\partial t}(0) \right\|_{L^2(S(\boldsymbol{y}))} + \left\| \frac{\partial f}{\partial t} \right\|_{L^2([0,T]; L^2(S(\boldsymbol{y})))} \bigg) \bigg], \quad C_\theta > 0.
\end{aligned}
\tag{11}
$$

In the particular case of the Crank-Nicolson method, with additional assumptions $\frac{\partial^2 f}{\partial t^2} \in L^2([0,T] \times S(\boldsymbol{y}))$ and $\frac{\partial^2 u_h}{\partial t^2} \in L^2(S(\boldsymbol{y}))$ we obtain

$$
\begin{aligned}
&\|u(t_i) - u_h(t_i)\|_{L^2(S(\boldsymbol{y}))} \\
&\leq C_{1/2} \bigg[h^{p+1} \bigg(|u(0)|_{H^{p+1}(S(\boldsymbol{y}))} + \left\| \frac{\partial u}{\partial t} \right\|_{L^1([0,T]; H^{p+1}(S(\boldsymbol{y})))} \bigg) \\
&\quad + (\Delta t)^2 \bigg(\left\| \frac{\partial^2 u_h}{\partial t^2}(0) \right\|_{L^2(S(\boldsymbol{y}))} + \left\| \frac{\partial^2 f}{\partial t^2} \right\|_{L^2([0,T]; L^2(S(\boldsymbol{y})))} \bigg) \bigg],
\end{aligned}
\tag{12}
$$

where $u(t_i)$ refers to the exact solution at time t_i, $u_h \in V_h(\boldsymbol{y})$ is the Galerkin approximation and p is the minimum of p_1 and p_2, cp. (9). Herein, C_θ and $C_{1/2}$ are independent of Δt and h.

5. Computation of quantities of interest

5.1. Quantities of interest

We consider quantities of interest that are defined with respect to the unperturbed reference surface S_0. There holds for an arbitrary time dependent function $u_h(t) \in V_h(\boldsymbol{y})$ that

$$
\begin{aligned}
\hat{u}_h(\boldsymbol{y}, \hat{\boldsymbol{x}}, t) &= \left(u_h(t) \circ \boldsymbol{\chi}\right)(\boldsymbol{y}, \hat{\boldsymbol{x}}) = \sum_{i=1}^{N_{\boldsymbol{p},\Xi}} u_i(\boldsymbol{y}, t)(\varphi_i \circ \boldsymbol{\chi})(\boldsymbol{y}, \hat{\boldsymbol{x}}) \\
&= \sum_{i=1}^{N_{\boldsymbol{p},\Xi}} u_i(\boldsymbol{y}, t)\hat{\varphi}_i(\hat{\boldsymbol{x}}) = \widehat{\boldsymbol{\Phi}}\boldsymbol{u}(\boldsymbol{y}, t)
\end{aligned}
$$

with the time and parameter dependent coefficient vector

$$\boldsymbol{u}(\boldsymbol{y}, t) := [u_1(\boldsymbol{y}, t), u_2(\boldsymbol{y}, t), \ldots, u_{N_{\boldsymbol{p}, \Xi}}(\boldsymbol{y}, t)]^{\mathsf{T}}.$$

This means that, by construction, the coefficients of the basis representations in $\mathcal{S}_{\boldsymbol{p}, \Xi}(S(\boldsymbol{y}))$ and $\mathcal{S}_{\boldsymbol{p}, \Xi}(S_0)$ coincide. Consequently, we obtain the following two equations for the expectation and correlation respectively,

$$\mathbb{E}[\hat{u}_h(t)](\hat{\boldsymbol{x}}) = \int_{\Omega} \widehat{\boldsymbol{\Phi}}(\hat{\boldsymbol{x}}) \boldsymbol{u}(\boldsymbol{y}, t) \mathrm{d}\mathbb{P}(\omega) = \widehat{\boldsymbol{\Phi}}(\hat{\boldsymbol{x}}) \int_{\Omega} \boldsymbol{u}(\boldsymbol{y}, t) \mathrm{d}\mathbb{P}(\omega) = \widehat{\boldsymbol{\Phi}}(\hat{\boldsymbol{x}}) \mathbb{E}[\boldsymbol{u}(t)],$$

and

$$\mathrm{Cor}[\hat{u}_h]\big((\hat{\boldsymbol{x}}, t), (\hat{\boldsymbol{x}}', t')\big) = \widehat{\boldsymbol{\Phi}}(\hat{\boldsymbol{x}}) \mathbb{E}[\boldsymbol{u}(t) \boldsymbol{u}^{\mathsf{T}}(t')] \widehat{\boldsymbol{\Phi}}^{\mathsf{T}}(\hat{\boldsymbol{x}}').$$

Hence, the quantities of interest under consideration can be approximated by computing the corresponding quantities of interest of the coefficient vectors, i.e., $\mathbb{E}[\boldsymbol{u}(t)]$ and $\mathbb{E}[\boldsymbol{u}(t) \boldsymbol{u}^{\mathsf{T}}(t')]$. Numerically, we approximate those quantities by applying suitable quadrature formulas in the parameter, such as the Monte Carlo method and the quasi-Monte Carlo quadrature based on Halton points, cf. [9]. Thus, we end up with approximations

$$\mathbb{E}[\hat{u}_h(t)] \approx \widehat{\boldsymbol{\Phi}}\bigg(\frac{1}{N_q} \sum_{i=1}^{N_q} \boldsymbol{u}(\boldsymbol{\xi}_i, t)\bigg), \tag{13}$$

and

$$\mathrm{Cor}[\hat{u}_h]\big((\hat{\boldsymbol{x}}, t), (\hat{\boldsymbol{x}}', t')\big) \approx \widehat{\boldsymbol{\Phi}}(\hat{\boldsymbol{x}})\bigg(\frac{1}{N_q} \sum_{i=1}^{N_q} \boldsymbol{u}(\boldsymbol{\xi}_i, t) \boldsymbol{u}(\boldsymbol{\xi}_i, t')^{\mathsf{T}}\bigg) \widehat{\boldsymbol{\Phi}}^{\mathsf{T}}(\hat{\boldsymbol{x}}'), \tag{14}$$

where $\boldsymbol{\xi}_i \in \Gamma_m := [-1, 1]^m$, $i = 1, \ldots, N_q$, are the sample points. Quantities of interest based on linear functionals of the solution can be dealt with in a similar fashion. In summary, we remark that the computation of quantities of interest amounts to high dimensional quadrature problems for the coefficient vector of the basis representation in $\mathcal{S}_{\boldsymbol{p}, \Xi}(S_0)$.

For the computation of the expectation, we only need to store one coefficient vector for each desired time point. In total, the required memory size is thus $\mathcal{O}(N_t N_{\boldsymbol{p}, \Xi})$. In the case of the correlation, however, the situation is much worse, as we need to store an $(N_t N_{\boldsymbol{p}, \Xi}) \times (N_t N_{\boldsymbol{p}, \Xi})$ matrix, resulting in a memory size of $\mathcal{O}(N_t^2 N_{\boldsymbol{p}, \Xi}^2)$. This matrix easily exceeds the memory size of a large computer if N_t or $N_{\boldsymbol{p}, \Xi}$ is large. This motivates the low rank representation of the correlation, which is discussed in the subsequent paragraph.

5.2. Low rank approximation of the correlation

For the cost efficient storage of the correlation, we employ an online algorithm that keeps track of the most important directions for the storage of the correlation matrix. To this end, we consider the online singular value decomposition (SVD) suggested in [7]. Given the sequence of time-dependent coefficient vectors $\{\boldsymbol{u}(\boldsymbol{\xi}_i, t)\}_{i=1}^{N_q}$ defined by the evaluation at the sample points, we employ the online singular value decomposition for each discrete time step. To outline the algorithm, we define the matrices

$$\boldsymbol{W}_\ell(t_i) := [\boldsymbol{u}^{(1)}(t_i), \ldots, \boldsymbol{u}^{(\ell)}(t_i)] := [\boldsymbol{u}(\boldsymbol{\xi}_1, t_i), \ldots, \boldsymbol{u}(\boldsymbol{\xi}_\ell, t_i)] \in \mathbb{R}^{N_{\boldsymbol{p}, \Xi} \times \ell},$$

which contain the first ℓ samples at time $t = t_i$. The coefficient matrix of the correlation of the solution between the times t_i and $t_{i'}$ for the first ℓ samples is thus given by

$$\boldsymbol{C}_\ell(t_i, t_{i'}) = \frac{1}{\ell} \boldsymbol{W}_\ell(t_i) \boldsymbol{W}_\ell^{\mathsf{T}}(t_{i'}).$$

We denote the SVD of $\boldsymbol{W}_\ell(t_i)$ by

$$\boldsymbol{W}_\ell(t_i) = \boldsymbol{U}_\ell(t_i) \boldsymbol{\Sigma}_\ell(t_i) \boldsymbol{V}_\ell^{\mathsf{T}}(t_i).$$

In the ℓ-th step, the vector $\boldsymbol{u}^{(\ell)}(t_i)$ is used to produce a potential new column vector of the matrix $\boldsymbol{U}_\ell(t_i)$. Accounting for the situation of limited memory, we only consider the first $k \leq N_q$ most important columns of $\boldsymbol{U}_\ell(t_i)$. Here, the importance is determined by the magnitude of the associated singular value stored in the matrix $\boldsymbol{\Sigma}_\ell(t_i)$. As we only wish to compute the correlation, we do not need to explicitly store the right-orthonormal matrices $\boldsymbol{V}_\ell(t_i)$. Instead, we rather store a low dimensional coefficient matrix $\boldsymbol{L}_\ell(t_i, t_{i'}) := \frac{1}{\ell} \boldsymbol{V}_\ell^{\mathsf{T}}(t_i) \boldsymbol{V}_\ell(t_{i'})$ between the times t_i and $t_{i'}$. Given that we only keep track of the first k most important vectors, the size of each $\boldsymbol{L}_\ell(t_i, t_{i'})$ is never larger than $k \times k$.

In what follows, we will discuss how to compute the low rank approximation of the space-time coefficient matrix in detail by adapting the idea from [7]. Given a new sample $\boldsymbol{u}^{(\ell)}(t_i)$ for $i = 1, \cdots, N_t$, the first step is computing the coefficient vector and the residual at each discrete time t_i by projecting the new sample solution onto the subspace spanned by the orthonormal basis $\boldsymbol{U}_{\ell-1}(t_i)$. This way, we obtain the coefficients

$$\boldsymbol{c}(t_i) = \boldsymbol{U}_{\ell-1}^{\mathsf{T}}(t_i)\boldsymbol{u}^{(\ell)}(t_i)$$

and the remainder

$$\hat{\boldsymbol{r}}(t_i) = \boldsymbol{u}^{(\ell)}(t_i) - \boldsymbol{U}_{\ell-1}(t_i)\boldsymbol{c}(t_i).$$

The associated normalized remainder is given by

$$\boldsymbol{r}(t_i) = \frac{\hat{\boldsymbol{r}}(t_i)}{r(t_i)} \quad \text{with } r(t_i) := \|\hat{\boldsymbol{r}}(t_i)\|_2.$$

Next, we pad $\boldsymbol{U}_{\ell-1}(t_i)$ by $\boldsymbol{r}(t_i)$ and $\boldsymbol{\Sigma}_{\ell-1}(t_i)$ by adding $\boldsymbol{c}(t_i)$ and $r(t_i)$ according to

$$\hat{\boldsymbol{U}}_\ell(t_i) := \begin{bmatrix} \boldsymbol{U}_{\ell-1}(t_i) & \boldsymbol{r}(t_i) \end{bmatrix}$$

and

$$\hat{\boldsymbol{\Sigma}}_\ell(t_i) := \begin{bmatrix} \boldsymbol{\Sigma}_{\ell-1}(t_i) & \boldsymbol{c}(t_i) \\ \boldsymbol{0} & r(t_i) \end{bmatrix}.$$

Especially, we have $\boldsymbol{W}_\ell(t_i) = \hat{\boldsymbol{U}}_\ell(t_i)\hat{\boldsymbol{\Sigma}}_\ell(t_i)\hat{\boldsymbol{V}}_\ell^{\mathsf{T}}(t_i)$, where

$$\hat{\boldsymbol{V}}_\ell(t_i) := \begin{bmatrix} \boldsymbol{V}_{\ell-1}(t_i) & \boldsymbol{0} \\ \boldsymbol{0} & 1 \end{bmatrix}.$$

In order to store the low rank approximation of the space-time coefficient matrix, we need to update the small matrix $\boldsymbol{L}_\ell(t_i, t_{i'})$ between t_i and $t_{i'}$ for $i, i' = 1, \cdots, N_t$. To this end, we pad it according to

$$\hat{\boldsymbol{L}}_\ell(t_i, t_{i'}) := \frac{1}{\ell}\hat{\boldsymbol{V}}_\ell^{\mathsf{T}}(t_i)\hat{\boldsymbol{V}}_\ell(t_{i'}) = \frac{\ell-1}{\ell}\begin{bmatrix} \boldsymbol{L}_{\ell-1}(t_i, t_{i'}) & \boldsymbol{0} \\ \boldsymbol{0} & \frac{1}{\ell-1} \end{bmatrix}.$$

In the final step, we diagonalize $\hat{\boldsymbol{\Sigma}}_\ell(t_i)$, which amounts to the SVD of a low dimensional matrix according to $\hat{\boldsymbol{\Sigma}}_\ell(t_i) = \boldsymbol{P}_\ell(t_i)\boldsymbol{\Sigma}_\ell(t_n)\boldsymbol{Q}_\ell^{\mathsf{T}}(t_i)$ and perform the updates

$$\boldsymbol{U}_\ell(t_i) = \hat{\boldsymbol{U}}_\ell(t_i)\boldsymbol{P}_\ell(t_i), \tag{15}$$

and

$$\boldsymbol{L}_\ell(t_i, t_{i'}) = \boldsymbol{Q}_\ell^{\mathsf{T}}(t_i)\hat{\boldsymbol{L}}_\ell(t_i, t_{i'})\boldsymbol{Q}_\ell(t_{i'}).$$

If the rank of $\boldsymbol{U}_\ell(t_i)$ is larger than the maximally allowed number k, we truncate $\boldsymbol{U}_\ell(t_i)$ by choosing the left most k columns, given that the singular values are stored in decreasing order. Moreover, we truncate $\boldsymbol{\Sigma}_\ell(t_i)$ by keeping the top-left $k \times k$ block. Finally, we truncate $\boldsymbol{L}_\ell(t_i, t_{i'})$ by keeping the top most k rows if the rank of $\boldsymbol{U}_\ell(t_i)$ is larger than k and truncate $\boldsymbol{L}_\ell(t_i, t_{i'})$ by keeping the left most k columns if the rank of $\boldsymbol{U}_\ell(t_{i'})$ is larger than k. The procedure is summarized in Algorithm 1.

We arrive at a tall-and-skinny matrix $\boldsymbol{U}_\ell(t_i)$ whose columns are used as low rank basis for time t_i. In particular, the matrix of singular values $\boldsymbol{\Sigma}_\ell(t_i)$ gives us a means to track the importance of each basis vector. Moreover, the matrix $\boldsymbol{L}_\ell(t_i, t_{i'})$ represents the correlation structure between time points t_i and $t_{i'}$ in a compressed manner. In summary, we end up with the low rank approximation

$$\boldsymbol{C}_{N_q}(t_i, t_{i'}) \approx \tilde{\boldsymbol{C}}_{N_q}(t_i, t_{i'}) := \boldsymbol{U}_{N_q}(t_i)\boldsymbol{\Sigma}_{N_q}(t_i)\boldsymbol{L}_{N_q}(t_i, t_{i'})\boldsymbol{\Sigma}_{N_q}^{\mathsf{T}}(t_{i'})\boldsymbol{U}_{N_q}^{\mathsf{T}}(t_{i'}).$$

The corresponding low rank space time correlation is hence given by

$$\mathrm{Cor}[\hat{u}_h]\big((\hat{\boldsymbol{x}}, t_i), (\hat{\boldsymbol{x}}', t_{i'})\big) \approx \widehat{\boldsymbol{\Phi}}(\hat{\boldsymbol{x}})\tilde{\boldsymbol{C}}_{N_q}(t_i, t_{i'})\widehat{\boldsymbol{\Phi}}^{\mathsf{T}}(\hat{\boldsymbol{x}}').$$

We conclude this paragraph by discussing the computational cost of the low rank representation. Storing the matrices $\boldsymbol{L}(t_i, t_{i'})$, $i, i' = 1, \ldots, N_t$ results in cost of $\mathcal{O}(N_t k^2)$, while the cost for storing the bases $\boldsymbol{U}(t_i)$ is of cost $\mathcal{O}(N_t N_{\boldsymbol{p}, \Xi} k)$. Assuming $k \le N_{\boldsymbol{p}, \Xi}$, the storage cost is hence of order $\mathcal{O}(N_t N_{\boldsymbol{p}, \Xi} k)$, which is a huge reduction compared to the original cost of $\mathcal{O}(N_t^2 N_{\boldsymbol{p}, \Xi}^2)$. The computational cost for obtaining the low rank approximation is comprised of the projection steps of cost $\mathcal{O}(N_q N_t N_{\boldsymbol{p}, \Xi} k)$, while the cost of the required SVD in each step is $\mathcal{O}(N_q N_t^2 k^3)$ and the cost of the update formula (15) is $\mathcal{O}(N_t N_{\boldsymbol{p}, \Xi} N_q k^2)$. This amounts to an overall computation cost of $\mathcal{O}(N_q N_t N_{\boldsymbol{p}, \Xi} k^2 + N_q N_t^2 k^3)$, which for moderate k is again much smaller than the original cost of $\mathcal{O}(N_q (N_t N_{\boldsymbol{p}, \Xi})^2)$.

Algorithm 1: Low rank approximation of the space time correlation.

Input: stream of solutions $\{u(t, \xi_i)\}_{i=1}^{N_q}$
Output: $U(t_i)$, $\Sigma(t_i)$ and $L(t_i, t_{i'})$
for $j = 1$ **to** N_q **do**
   for $i = 1$ **to** N_t **do**
      $\hat{r}(t_i) = u(t_i, \xi_i)$;
      if $j = 1$ **then**
         $[U(t_i), \Sigma(t_i), V(t_i)] = \mathrm{SVD}(\hat{r}(t_i))$;
      else
         $c(t_i) = U^{\top}(t_i)\hat{r}(t_i)$
         $\hat{r}(t_i) = u_i(t_i) - U(t_i)c(t_i)$
         $r(t_i) = \|\hat{r}(t_i)\|_2$
         $r(t_i) = \hat{r}(t_i)/r(t_i)$
         $\hat{U}(t_i) = [U(t_i), r(t_i)]$;
         $\hat{\Sigma}(t_i) = [\Sigma(t_i), c(t_i); 0, r(t_i)]$;
         $[P(t_i), \Sigma(t_i), Q(t_i)] = \mathrm{SVD}(\hat{\Sigma}(t_i))$;
         $U(t_i) = \hat{U}(t_i)P(t_i)$;
      end
   end
   for $i = 1$ **to** N_t **do**
      for $i' = i$ **to** N_t **do**
         if $j = 1$ **then**
            $L(t_i, t_{i'}) = V^{\top}(t_i)V(t_{i'})$;
         else
            $\hat{L}(t_i, t_{i'}) = \frac{j-1}{j}[L(t_i, t_{i'}), 0; 0, \frac{1}{j-1}]$;
            $L(t_i, t_{i'}) = Q^{\top}(t_i)\hat{L}(t_i, t_{i'})Q(t_{i'})$;
         end
         if $j > k$ **then**
            $L(t_i, t_{i'}) = L(t_i, t_{i'})(1 : k, 1 : k)$
         end
      end
      if $j > k$ **then**
         $U(t_i) = U(t_i)(:, 1 : k)$
         $\Sigma(t_i) = \Sigma(t_i)(1 : k, 1 : k)$
      end
   end
end

---

6. Numerical experiments

6.1. Convergence of the spatial solver

All computations for this article have been performed with the Boundary Element Method Based Engineering Library (`Bembel`), see [18]. In particular, we have extended `Bembel` to also support local operators, like the Laplace-Beltrami operator, and added a class for the fast deformation of computational geometries. We numerically validate our implementation for the diffusion equation by showing that the theoretical convergence rates from Theorem 4.1 are obtained. At first, we consider a diffusion equation on the unit sphere, i.e.,

$$\begin{cases} \partial_t u(\boldsymbol{x}, t) - \Delta_S u(\boldsymbol{x}, t) = 0, & \boldsymbol{x} \in \mathbb{S}^2, \\ u(\boldsymbol{x}, 0) = x_3, & \boldsymbol{x} \in \mathbb{S}^2. \end{cases} \tag{16}$$

The corresponding exact solution is $u(\boldsymbol{x}, t) = e^{-2t}x_3$. Fig. 3 shows the corresponding convergence plot for a representation of the sphere with $M = 6$ patches and a uniform refinement up to level 4 and $\Delta t = 10^{-5}$, evaluated for the final time $T = 1$. For the time stepping we use the Crank-Nicolson method and we consider $p = 1, 2, 3$. It can be seen that the theoretical rate from Theorem 4.1 is achieved. Indeed, the observed rate appears to be even higher.

Next, in view of the previous numerical results and Theorem 4.1, we fix $p = 2$ and $\Delta t = 10^{-3}$, such that the convergence is not limited by the time discretization. We consider the diffusion problem

$$\begin{cases} \partial_t u(\boldsymbol{x}, t) - \Delta_S u(\boldsymbol{x}, t) = \sin(\pi x_1)\sin(\pi x_2)\sin(\pi x_3), & \boldsymbol{x} \in S, \\ u(\boldsymbol{x}, 0) = x_3, & \boldsymbol{x} \in S, \end{cases}$$

for three different geometries. Namely, these are the unit sphere (bounding box: $[-1, 1]^3$), a pipe geometry (bounding box: $[0, 2.4] \times [-0.3, 0.3]^2$) and the Stanford bunny (bounding box: $[-0.95, 0.61] \times [0.33, 1.86] \times [-0.61, 0.59]$). Vizualisations of the solutions for $T = 1$ can be found in Fig. 4.
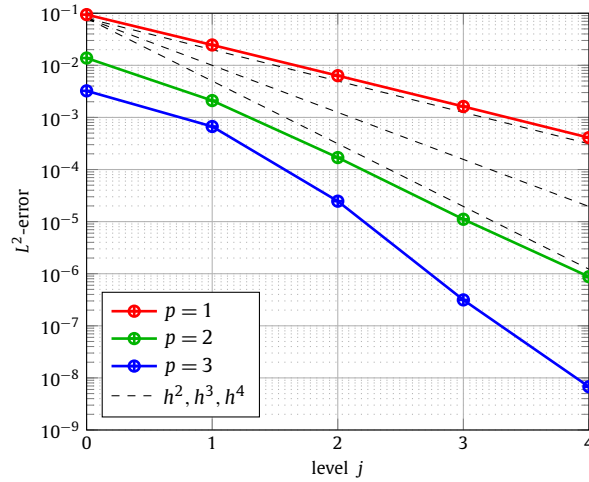
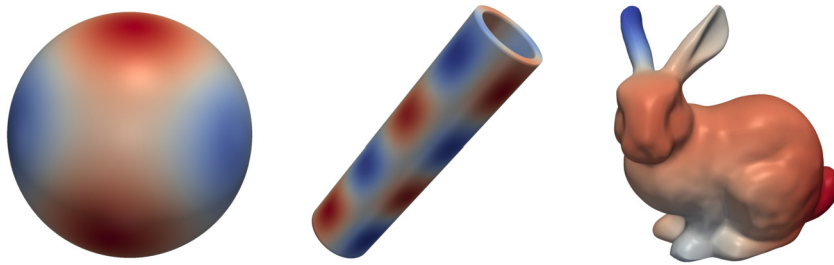**Fig. 3.** L^2-error versus the level j on the unit sphere.

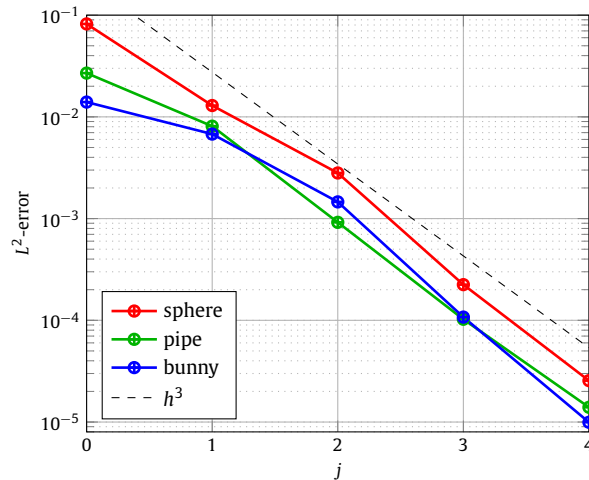**Fig. 4.** Visualization of the solutions at time $T = 1$.

Fig. 5. Average L^2-error decrease for $p = 2$ and the different geometries under consideration.

For the convergence study, we compute the average L^2-error evaluated at the time grid $t = 0.001, 0.002, \ldots, 1$, i.e.,

$$e_{L^2}^{(j)} = \frac{1}{N_t} \sum_{i=1}^{N_t} \left\| u_J(t_i) - u_j(t_i) \right\|_{L^2(S)}, \quad N_t = 1000,$$

where we use the numerical solution on level $J = 5$ as ground truth. The corresponding error plot can be found in Fig. 5. As can be seen, the error is almost divided by 8 when the level j increases by one, which reflects the expected convergence rate of $h_j \sim 2^{-3j}$.
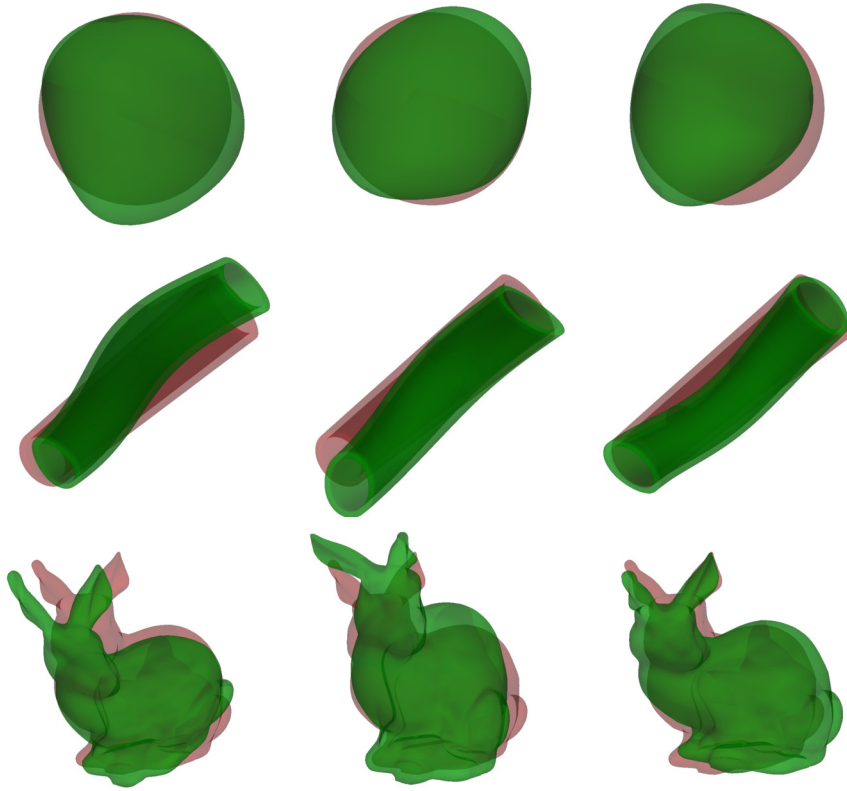
Fig. 6. Different realizations of the random deformation field for the three surfaces under consideration. The transparent red ones are the reference surfaces, while green ones are the deformed surfaces.

6.2. Convergence of the low rank approximation

In this part, we numerically test the convergence of the proposed algorithm with respect to the predefined rank k, the generalization error and the stability of the algorithm with respect to the magnitude of the domain perturbation. Now we consider the diffusion problem

$$\begin{cases} \partial_t u(\boldsymbol{y}, \boldsymbol{x}, t) - \Delta_S u(\boldsymbol{y}, \boldsymbol{x}, t) = \sin(\pi x_1)\sin(\pi x_2)\sin(\pi x_3), & \boldsymbol{x} \in S(\boldsymbol{y}), \\ u(\boldsymbol{y}, \boldsymbol{x}, 0) = x_3, & \boldsymbol{x} \in S(\boldsymbol{y}). \end{cases}$$

As before, we consider the randomly deformed sphere, pipe and Stanford bunny. For the time discretization, we use again the Crank-Nicolson method with time steps $\Delta t = 0.001$ for the sphere and the pipe and $\Delta t = 0.01$ for the Stanford bunny. For the spatial discretization, we set the level and the polynomial order of the basis functions with $j = 4$ for the sphere and the pipe and $j = 3$ for the Stanford bunny. Moreover, we always choose the polynomial degree $p = 2$. For the sphere and pipe, this setting results in an accuracy of 10^{-5}, see Fig. 5. For the Stanford bunny, the computational cost is much larger than for the sphere and pipe because of the more complex geometry. Therefore, we reduce the level j by one, resulting in an accuracy of 10^{-4}, see Fig. 5. In all cases, the choice of the time step ensures that the error is not dominated by the time discretization. This leads to $N_{\boldsymbol{p}, \boldsymbol{\Xi}} = 1736, 6936$, and 14501 for the sphere, pipe and the Stanford bunny, respectively, and results in a spatial discretization error about 10^{-4} for all (undeformed) geometries, cp. previous example.

For the random deformation field, we consider the covariance function

$$\mathbb{E}[\boldsymbol{\chi}](\hat{\boldsymbol{x}}) = \hat{\boldsymbol{x}}, \quad \mathrm{Cov}[\boldsymbol{\chi}](\hat{\boldsymbol{x}}, \hat{\boldsymbol{x}}') = 10^{-2} \begin{bmatrix} e^{-50r^2} & 0 & 10^{-4}e^{-0.5r^2} \\ 0 & e^{-50r^2} & 0 \\ 10^{-4}e^{-0.5r^2} & 0 & e^{-50r^2} \end{bmatrix}$$

with $r := \|\hat{\boldsymbol{x}} - \hat{\boldsymbol{x}}'\|_2$. The random field is computed by the pivoted Cholesky decomposition, cf. [27–29], with an accuracy of 10^{-4}. This leads to the parameter dimensions 282, 252, 264 for the sphere, pipe and Stanford bunny, respectively. Different realizations of the random surfaces are depicted in Fig. 6.

The expectations and the standard deviations at the end of time $T = 1$ are displayed on the reference surfaces in Fig. 7.
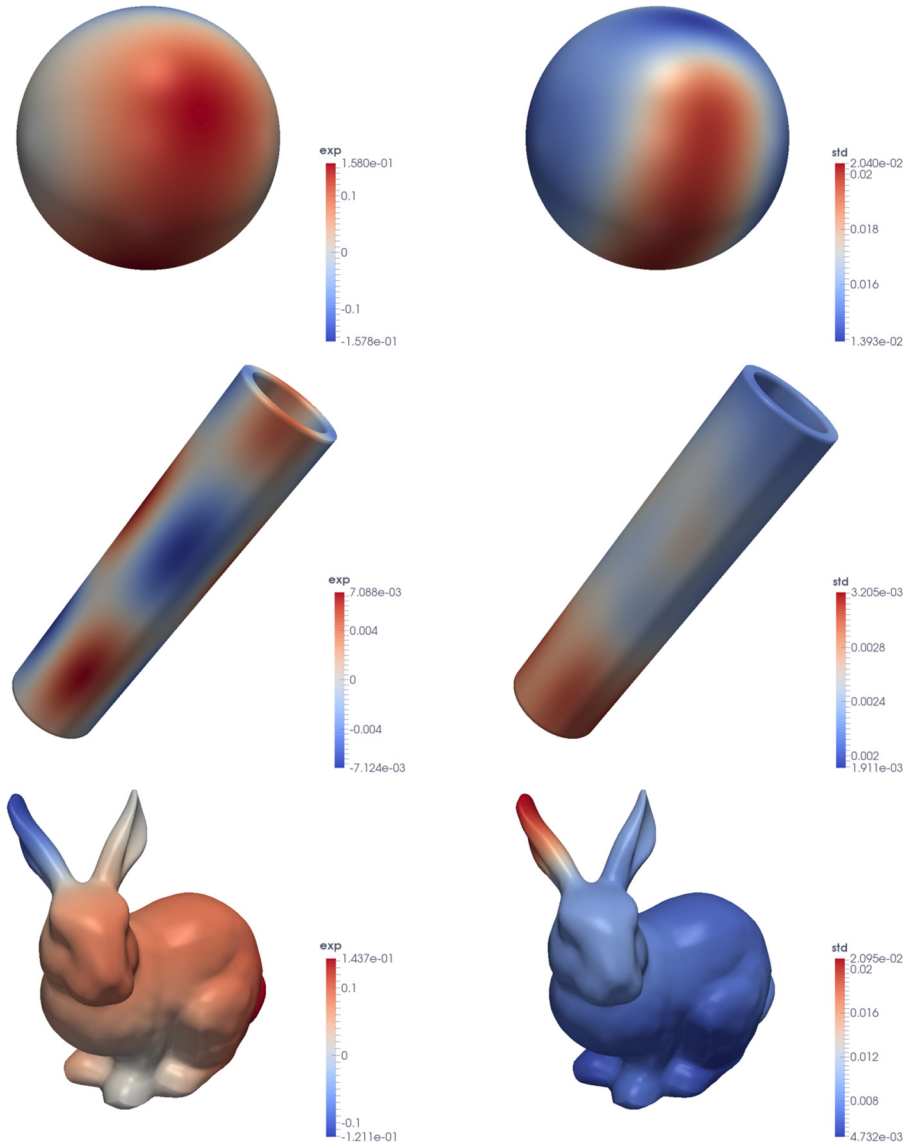
Fig. 7. Expectations of the solutions (left) and standard deviations (right) at time $T = 1$ for the three surfaces.

As can be seen, the ratio between the standard deviation and the expectation is around 13% for the sphere and the Stanford bunny, while it is about 40% for the pipe.

Next, we examine the convergence of the low rank representation. As we cannot store the full coefficient matrix of the space-time correlation as a reference, we only compute the diagonal blocks $C_{N_q}(t_i, t_i)$ for $t_i = 0.1i$, $i = 1, \ldots, 10$, and compare them to the low rank approximation $\tilde{C}_{N_q}(t_i, t_i)$. We set the number of samples to $N_q = 8192$ for the Monte Carlo and the quasi-Monte Carlo method based on Halton points. For the assessment of the error, we measure the relative Frobenius norm of the difference between the ground truth and the low rank via

$$e_F := \frac{1}{10} \sum_{i=1}^{10} \frac{\|C(t_i, t_i) - \tilde{C}(t_i, t_i)\|_F}{\|C(t_i, t_i)\|_F}.$$

The left hand side of Fig. 8 shows the error e_F in case of the Monte Carlo method, while the right hand side provides it for the quasi-Monte Carlo method. In both cases, the error rapidly decays for increasing k. As can be seen, the errors for both methods are comparable, however, they are slightly smaller for the quasi-Monte Carlo method.

The availability of the full space-time correlation facilitates the visualization and analysis of the correlation between different time points. In particular, due to the particular structure of the low rank approximation, we can directly identify the principal components between any two evaluated times, i.e., $U_\ell^\mathsf{T}(t_i) \tilde{C}(t_i, t_{i'}) U_\ell(t_{i'})$, $\ell \leq k$. For simplicity, we only show
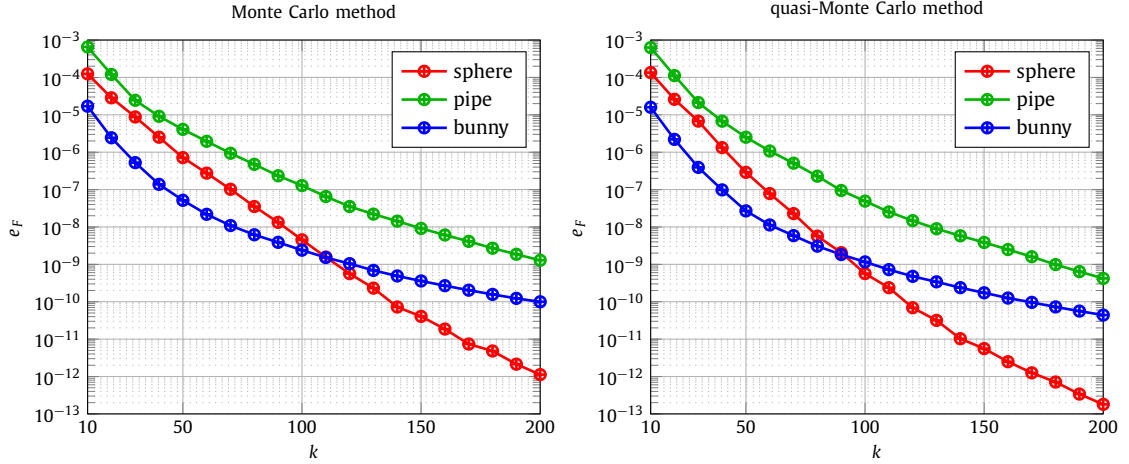
Fig. 8. Convergence of the low rank approximation for the Monte Carlo method and the quasi-Monte Carlo method.
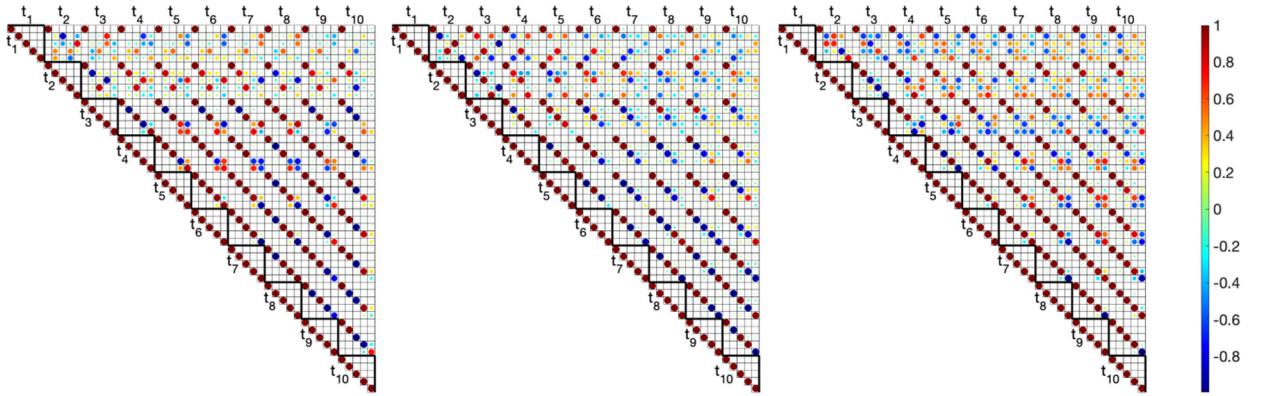

Fig. 9. Correlation coefficients for the first 5 principal components and the time points t_i for the sphere (left), the pipe (middle) and the Stanford bunny (right) in case of the quasi-Monte Carlo method.

the upper triangular part of the correlation matrix for the coefficients of the first 5 principal components at each time in Fig. 9 for the case of the quasi-Monte Carlo method. The radii of the circles inside each small block stand for the absolute values of the corresponding correlation coefficients. From the figures, the correlation matrix within the same time step is a 5×5 diagonal matrix (outlined by the black lines in the figures), since we use an orthogonal basis given by $\boldsymbol{U}(t_i)$ at each evaluated time. On the other hand, the solution at different times are highly correlated to each other.

Next, we consider the generalization error given the low rank bases $\boldsymbol{U}(t_i)$. To this end, we introduce the relative ℓ^2-error at the time point t_i, i.e.,

$$e_{\ell^2}(t_i) = \frac{1}{N_q} \sum_{j=1}^{N_q} \frac{\left\| \boldsymbol{u}(\tilde{\boldsymbol{\xi}}_j, t_i) - \boldsymbol{U}(t_i)\boldsymbol{U}(t_i)^{\mathsf{T}}\boldsymbol{u}(\tilde{\boldsymbol{\xi}}_j, t_i) \right\|_2}{\left\| \boldsymbol{u}(\tilde{\boldsymbol{\xi}}_j, t_i) \right\|_2},$$

for independently drawn Monte Carlo samples $\boldsymbol{u}(\tilde{\boldsymbol{\xi}}_j, t_i)$, $j = 1, \ldots N_q$, with $N_q = 8192$. As a measure for the generalization error, we consider now the average error

$$e_{\ell^2} = \frac{1}{N_t} \sum_{i=1}^{N_t} e_{\ell^2}(t_i).$$

The generalization error is displayed in Fig. 10. Particularly for the sphere, the generalization error rapidly decreases for an increasing rank k, where the Monte Carlo method performs slightly better here. For $k = 200$ the average ℓ^2-error is lowest for the sphere and around $4 \cdot 10^{-6}$ for both methods.

Finally, we examine the stability of the low rank approximation with respect to the magnitude of the random perturbation. To this end, we introduce the parameter α which steers the impact of the random perturbation according to
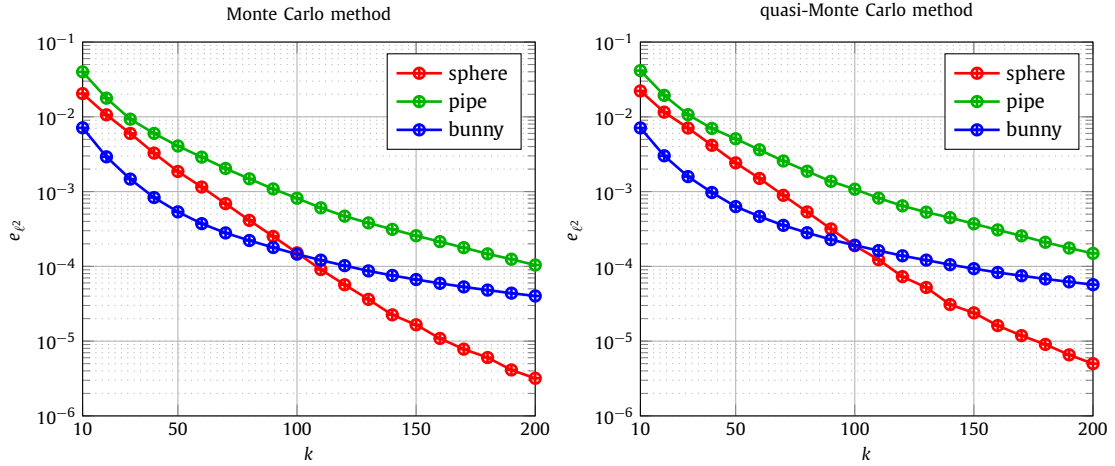
Fig. 10. Generalization experiments on 8192 test samples generated by the Monte Carlo method.
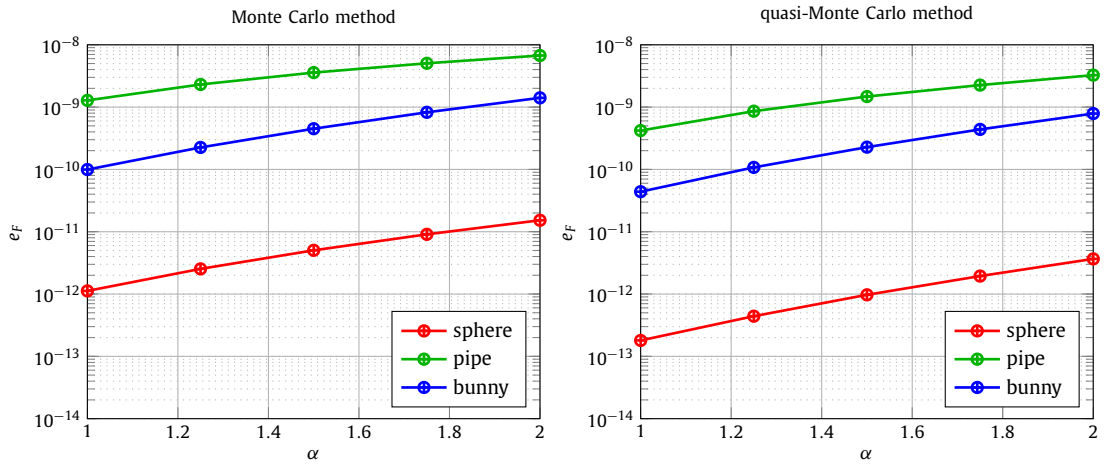

Fig. 11. Low rank approximation error with respect to the magnitude of the random perturbation.

$$\boldsymbol{\chi}(\boldsymbol{y}, \hat{\boldsymbol{x}}) = \mathbb{E}[\boldsymbol{\chi}](\hat{\boldsymbol{x}}) + \alpha \sum_{k=1}^{m} \sqrt{\lambda_k} \boldsymbol{\chi}_k(\hat{\boldsymbol{x}}) y_k, \quad \boldsymbol{y} \in \Gamma := [-1, 1]^m.$$

We fix $k = 200$ and consider $\alpha = 1, 1.2, \dots, 2$. This results in a maximal possible relative displacement between 20% and 40% for the sphere, a maximal possible relative displacement between 21% and 42% for the pipe, and a maximal possible relative displacement between 27% and 54% for the Stanford bunny. Fig. 11 depicts the corresponding error e_F for the covariance approximation.

As can be seen, both methods are rather robust with respect to the increase of the displacement, where the quasi-Monte Carlo method behaves slightly better.

The generalization error with respect to α is shown in Fig. 12.

As for the low rank error, the generalization error only moderately increases with an increase of the random perturbation, where here the Monte Carlo method performs slightly better.

7. Conclusion

In this article, we have presented an isogeometric approach to solving diffusion problems on random surfaces. Especially, we have described in detail, how diffusion problems on random surfaces can be modelled by means of random deformation fields in the isogeometric context and how quantities of interest may be derived. Moreover, we have employed an online low rank approximation algorithm for the high-dimensional space-time correlation of the random solution. Extensive numerical studies on complex geometries have been performed. The numerical results corroborate the efficacy of the presented methodology. Finally, the solver for the Laplace-Beltrami operator will be added to the isogeometric boundary element library `Bembel`.
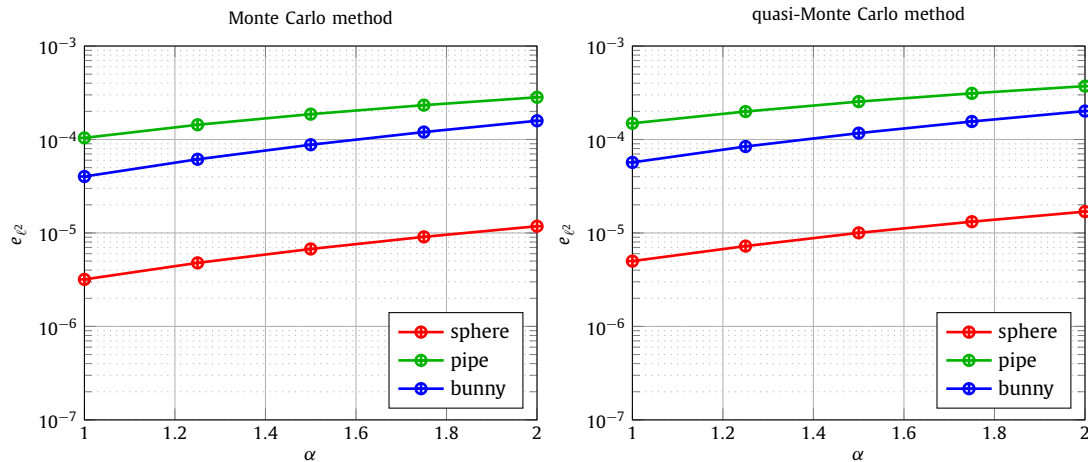
Fig. 12. Generalization error with respect to the magnitude of the randomness.

References

[1] D. Bassett, P. Webber, Diffusion of single adatoms of platinum, iridium and gold on platinum surfaces, Surf. Sci. 70 (1) (1978) 520–531.

[2] Y. Bazilevs, L.B. da Veiga, J. Cottrell, T. Hughes, G. Sangalli, Isogeometric analysis: approximation, stability and error estimates for h-refined meshes, Math. Models Methods Appl. Sci. 16 (07) (2006) 1031–1090.

[3] J. Beck, L. Tamellini, R. Tempone, IGA-based multi-index stochastic collocation for random PDEs on arbitrary domains, Comput. Methods Appl. Mech. Eng. 351 (2019) 330–350.

[4] M. Bertalmıo, L. Cheng, S. Osher, G. Sapiro, Variational problems and partial differential equations on implicit surfaces, J. Comput. Phys. 174 (2) (2001) 759–780.

[5] M. Bertalmio, G. Sapiro, L. Cheng, S. Osher, A framework for solving surface partial differential equations for computer graphics applications, CAM Report 00-43, UCLA, Mathematics Department, 2000.

[6] J. Born, P. Schmidt, L. Kobbelt, Layout embedding via combinatorial optimization, Comput. Graph. Forum 40 (2) (2021) 277–290.

[7] M. Brand, Incremental singular value decomposition of uncertain data with missing values, in: European Conference on Computer Vision, Springer, 2002, pp. 707–720.

[8] A. Buffa, J. Dölz, S. Kurz, S. Schöps, R. Vázquez, F. Wolf, Multipatch approximation of the de Rham sequence and its traces in isogeometric analysis, Numer. Math. 144 (1) (2020) 201–236.

[9] R.E. Caflisch, Monte Carlo and quasi-Monte Carlo methods, Acta Numer. 7 (1998) 1–49.

[10] M. Campen, Partitioning surfaces into quad patches, in: A. Bousseau, D. Gutierrez (Eds.), EG 2017 - Tutorials. The Eurographics Association, 2017.

[11] J.E. Castrillón-Candás, J. Xu, A stochastic collocation approach for parabolic PDEs with random domain deformations, Comput. Math. Appl. 93 (2021) 32–49.

[12] P. Chen, A. Quarteroni, G. Rozza, Reduced basis methods for uncertainty quantification, SIAM/ASA J. Uncert. Quant. 5 (1) (2017) 813–869.

[13] J.A. Cottrell, T.J. Hughes, Y. Bazilevs, Isogeometric Analysis: Toward Integration of CAD and FEA, John Wiley & Sons, 2009.

[14] K. Crane, C. Weischedel, M. Wardetzky, Geodesics in heat: a new approach to computing distance based on heat flow, ACM Trans. Graph. 32 (5) (2013) 1–11.

[15] L. Dedè, A. Quarteroni, Isogeometric analysis for second order partial differential equations on surfaces, Comput. Methods Appl. Mech. Eng. 284 (2015) 807–834.

[16] A. Djurdjevac, Linear parabolic problems in random moving domains, SIAM/ASA J. Uncert. Quant. 9 (2) (2021) 848–879.

[17] J. Dölz, H. Harbrecht, C. Jerez-Hanckes, M. Multerer, Isogeometric multilevel quadrature for forward and inverse random acoustic scattering, Comput. Methods Appl. Mech. Eng. 388 (2022) 114242.

[18] J. Dölz, H. Harbrecht, S. Kurz, M. Multerer, S. Schöps, F. Wolf, Bembel: the fast isogeometric boundary element C++ library for Laplace, Helmholtz, and electric wave equation, SoftwareX 11 (2020) 100476.

[19] G. Dziuk, Finite elements for the Beltrami operator on arbitrary surfaces, in: Partial Differential Equations and Calculus of Variations, Springer, 1988, pp. 142–155.

[20] G. Dziuk, C.M. Elliott, Finite element methods for surface PDEs, Acta Numer. 22 (2013) 289–396.

[21] G. Ehrlich, K. Stolt, Surface diffusion, Annu. Rev. Phys. Chem. 31 (1) (1980) 603–637.

[22] C.M. Elliott, B. Stinner, C. Venkataraman, Modelling cell motility and chemotaxis with evolving surface finite elements, J. R. Soc. Interface 9 (76) (2012) 3027–3044.

[23] J. Faraudo, Diffusion equation on curved surfaces. I. Theory and application to biological membranes, J. Chem. Phys. 116 (13) (2002) 5831–5841.

[24] L. Gander, R. Krause, M. Multerer, S. Pezzuto, Space-time shape uncertainties in the forward and inverse problem of electrocardiography, Int. J. Numer. Methods Biomed. Eng. 37 (10) (2021) e3522.

[25] R. Gantner, M. Peters, Higher-order quasi-Monte Carlo for Bayesian shape inversion, SIAM/ASA J. Uncert. Quant. 6 (2) (2018) 707–736.

[26] H. Harbrecht, M. Multerer, M. Siebenmorgen, Analysis of the domain mapping method for elliptic diffusion problems on random domains, Numer. Math. 134 (4) (2016) 823–856.

[27] H. Harbrecht, M. Multerer, R. von Rickenbach, Isogeometric shape optimization for scaffold structures, Comput. Methods Appl. Mech. Eng. 391 (2021) 114552.

[28] H. Harbrecht, M. Peters, R. Schneider, On the low-rank approximation by the pivoted Cholesky decomposition, Appl. Numer. Math. 62 (2012) 28–440.

[29] H. Harbrecht, M. Peters, M. Siebenmorgen, Efficient approximation of random fields for numerical applications, Numer. Linear Algebra Appl. 22 (4) (2015) 596–617.

[30] T.J.R. Hughes, J.A. Cottrell, Y. Bazilevs, Isogeometric analysis: cad, finite elements, nurbs, exact geometry and mesh refinement, Comput. Methods Appl. Mech. Eng. 194 (39–41) (2005) 4135–4195.

[31] B. Jüttler, A. Mantzaflaris, R. Perl, M. Rumpf, On numerical integration in isogeometric subdivision methods for PDEs on surfaces, Comput. Methods Appl. Mech. Eng. 302 (2016) 131–146.

[32] Y. Kazashi, F. Nobile, E. Vidličková, Stability properties of a projector-splitting scheme for dynamical low rank approximation of random parabolic equations, Numer. Math. 149 (4) (2021) 973–1024.

[33] J. Liang, H. Zhao, Solving partial differential equations on point clouds, SIAM J. Sci. Comput. 35 (3) (2013) A1461–A1486.

[34] M. Lyon, M. Campen, L. Kobbelt, Quad layouts via constrained T-mesh quantization, Comput. Graph. Forum 40 (2) (2021) 305–314.

[35] M. Lyon, M. Campen, L. Kobbelt, Simpler quad layouts using relaxed singularities, Comput. Graph. Forum 40 (5) (2021) 169–179.

[36] C. Macdonald, S. Ruuth, The implicit closest point method for the numerical solution of partial differential equations on surfaces, SIAM J. Sci. Comput. 31 (6) (2010) 4330–4350.

[37] C. Macdonald, S.J.S.J. Ruuth, Level set equations on surfaces via the closest point method, J. Sci. Comput. 35 (2) (2008) 219–240.

[38] M. Multerer, A note on the domain mapping method with rough diffusion coefficients, Appl. Numer. Math. 145 (2019) 283–296.

[39] I. Novak, F. Gao, Y. Choi, D. Resasco, J. Schaff, B. Slepchenko, Diffusion on a curved surface coupled to diffusion in the volume: application to cell biology, J. Comput. Phys. 226 (2) (2007) 1271–1290.

[40] L. Piegl, W. Tiller, The NURBS Book, Springer Science & Business Media, 1996.

[41] N. Pietroni, E. Puppo, G. Marcias, R. Scopigno, P. Cignoni, Tracing field-coherent quad layouts, Comput. Graph. Forum 35 (7) (2016) 485–496.

[42] S. Ruuth, B. Merriman, A simple embedding method for solving partial differential equations on surfaces, J. Comput. Phys. 227 (3) (2008) 1943–1961.

[43] C. Spannring, S. Ullmann, J. Lang, A weighted reduced basis method for parabolic PDEs with random data, in: International Conference on Computational Engineering, Springer, 2017, pp. 145–161.

[44] L. Tian, C. Macdonald, S. Ruuth, Segmentation on surfaces with the closest point method, in: 2009 16th IEEE International Conference on Image Processing (ICIP), IEEE, 2009, pp. 3009–3012.

[45] L.D. Veiga, A. Buffa, J. Rivas, G. Sangalli, Some estimates for h–p–k-refinement in isogeometric analysis, Numer. Math. 118 (2) (2011) 271–305.

[46] C. Wu, J. Deng, F. Chen, Diffusion equations over arbitrary triangulated surfaces for filtering and texture applications, IEEE Trans. Vis. Comput. Graph. 14 (3) (2008) 666–679.

[47] D. Xiu, D. Tartakovsky, Numerical methods for differential equations in random domains, SIAM J. Sci. Comput. 28 (3) (2006) 1167–1185.

[48] V.P. Zhdanov, Elementary Physicochemical Processes on Solid Surfaces, Springer Science & Business Media, 2013.

[49] S. Zhu, L.L. Dedè, A. Quarteroni, Isogeometric analysis and proper orthogonal decomposition for parabolic problems, Numer. Math. 135 (2) (2017) 333–370.