Weierstraß-Institut
für Angewandte Analysis und Stochastik

Leibniz-Institut im Forschungsverbund Berlin e. V.

Preprint ISSN 2198-5855

Efficient approximation of high-dimensional exponentials by tensor networks

Martin Eigel[1], Nando Farchmin[2], Sebastian Heidenreich[2], Philipp Trunschke[3]

submitted: May 27, 2021

[1] Weierstrass Institute
Mohrenstr. 39
10117 Berlin
Germany
E-Mail: martin.eigel@wias-berlin.de

[2] Physikalisch Technische Bundesanstalt
Abbestr. 2-12
10587 Berlin
Germany
E-Mail: nando.farchmin@ptb.de
sebastian.heidenreich@ptb.de

[3] Technische Universität Berlin
Institut für Mathematik
Straße des 17. Juni 136
10623 Berlin
Germany
E-Mail: trunschke@math.tu-berlin.de

No. 2844
Berlin 2021

2020 *Mathematics Subject Classification.* 35R60, 47B80, 60H35, 65C20, 65N12, 65N22, 65J10.

Key words and phrases. Uncertainty quantification, dynamical system approximation, Petrov–Galerkin, a posteriori error bounds, tensor product methods, tensor train format, holonomic functions, Bayesian likelihoods, log-normal random field.

M. Eigel acknowledges the partial support by the DFG SPP 1886 “Polymorphic Uncertainty Modelling for the Numerical Design of Structures”. N. Farchmin has received funding form the German Central Innovation Program (ZIM) No. ZF4014017RR7. P. Trunschke acknowledges support by the Berlin International Graduate School in Model and Simulation based Research (BIMoS). The authors would like to thank Maren Casfor, Michael Götte, Robert Lasarzik, Mathias Oster, Leon Sallandt and Reinhold Schneider for fruitful discussions.

Edited by
Weierstraß-Institut für Angewandte Analysis und Stochastik (WIAS)
Leibniz-Institut im Forschungsverbund Berlin e. V.
Mohrenstraße 39
10117 Berlin
Germany

Fax: +49 30 20372-303
E-Mail: preprint@wias-berlin.de
World Wide Web: http://www.wias-berlin.de/

Efficient approximation of high-dimensional exponentials by tensor networks

Martin Eigel, Nando Farchmin, Sebastian Heidenreich, Philipp Trunschke

ABSTRACT. In this work a general approach to compute a compressed representation of the exponential $\exp(h)$ of a high-dimensional function h is presented. Such exponential functions play an important role in several problems in Uncertainty Quantification, e.g. the approximation of log-normal random fields or the evaluation of Bayesian posterior measures. Usually, these high-dimensional objects are intractable numerically and can only be accessed pointwise in sampling methods. In contrast, the proposed method constructs a functional representation of the exponential by exploiting its nature as a solution of an ordinary differential equation. The application of a Petrov–Galerkin scheme to this equation provides a tensor train representation of the solution for which we derive an efficient and reliable a posteriori error estimator. Numerical experiments with a log-normal random field and a Bayesian likelihood illustrate the performance of the approach in comparison to other recent low-rank representations for the respective applications. Although the present work considers only a specific differential equation, the presented method can be applied in a more general setting. We show that the composition of a generic holonomic function and a high-dimensional function corresponds to a differential equation that can be used in our method. Moreover, the differential equation can be modified to adapt the norm in the a posteriori error estimates to the problem at hand.

1. INTRODUCTION

In this work we consider the problem of approximating the exponential $u = \exp(h)$ of a differentiable multivariate function $h(y)$. Before presenting the new approach, we first illustrate the relevance of this often challenging task by two examples from the field of Uncertainty Quantification (UQ).

First, consider the steady state diffusion in a porous medium also known as Darcy's problem. This is a standard benchmark problem in forward UQ and can be modeled by the second order linear partial differential equation

$$-\operatorname{div}(\kappa \nabla w) = f, \tag{1.1}$$

where w is the concentration of some substance, the diffusion coefficient κ determines the mobility of the particles and f describes sources or sinks. In practical applications the diffusion coefficient often takes the form $\kappa(x, y) = \exp(\gamma(x, y))$ where the function γ depends on the spatial coordinate x as well as a random parameter y that models uncertainties. A popular approach for solving equation (1.1) is the *stochastic Galerkin* (SG) method (cf. [1, 2]) as it can be interpreted as an extension of the *finite element method* (FEM) to parametric random PDEs. In comparison to simpler sampling estimates such as different variants of the Monte Carlo method, the convergence of the SG method is potentially much faster since it exploits inherent structural properties of the considered problem, in particular anisotropic sparsity that can be captured in an appropriate generalized polynomial chaos basis. Moreover, when using SG methods a reliable a posteriori error estimator can be computed, leading to a quasi optimal iterative construction of the discrete space as introduced in [3, 4].

To apply the SG method to the variational formulation of equation (1.1), a functional representation of the diffusion coefficient κ is required, which is provided by our proposed approach.

A second related example is the representation of some posterior probability density π_δ in the context of Bayesian inference [5, 6] via the exponential of the log-likelihood $\ell(y; \delta)$ and the prior π_0 as

$$\frac{\mathrm{d}\pi_\delta}{\mathrm{d}\pi_0}(y) \propto \exp(\ell(y; \delta)).$$

In UQ this task arises for example in the parameter reconstruction of model data via inverse problems [7–9]. If a functional representation of $\exp(\ell(y; \delta))$ can be constructed, it may for instance be

DOI 10.20347/WIAS.PREPRINT.2844 Berlin 2021

used to efficiently generate independent posterior samples [10] or to compute high-dimensional quantities of interest such as moments or marginals [11].

The preceding two examples highlight the benefits of a functional representation of the occurring exponentials, which however is difficult to obtain. A classical approach to represent any multivariate function $u : \mathbb{R}^M \to \mathbb{R}$ numerically is to choose a set of d polynomial basis functions $\{p_i^m(y_m)\}_{i=0,\dots,d-1}$ for each coordinate y_m and each $m = 1, \dots M$, and then apply the expansion

$$u(y) \approx \sum_{i_1=0}^{d-1} \cdots \sum_{i_M=0}^{d-1} c[i_1, \dots, i_M]\, p_{i_1}^1(y_1) \cdots p_{i_M}^M(y_M).$$

Note that in both examples mentioned above, the dimension M of the parameter vector y may vary for practical problems from just a few to well over a hundred. This renders a standard product basis representation like this unfeasible since d^M coefficients would be required. One remedy to this problem is to use a sparse representation of the coefficient tensor c. This works well as long as the function u can be sparsely represented in the chosen product basis. However, to represent the exponential function $u(y) := \exp(h(y))$ higher-order multivariate polynomials are necessary and theoretical results indicate [12] that the number of required coefficients in a sparse representation quickly becomes intractable in practice. To mitigate this problem tensor network representations [13] have been successfully applied to both examples. This work focuses in particular on the *tensor train* (TT) format [14], also know as matrix product states, as a technique to compress the high-dimensional coefficient tensor c. Given such a representation of the diffusion coefficient κ it is easy to apply the SG algorithm [15]. If the likelihood function $\exp(\ell(y;\delta))$ is given in TT format quantities of interest [11] can be computed promptly and samples can be draw efficiently [10].

Concerning the diffusion coefficient κ there exist several approaches to represent c of (1) in TT format. In [15] an exact representation is developed for the case that $\gamma(x, y)$ is an affine function in the parameters y. However, since the construction is tailored to the specific structure of γ, it is not straightforward to apply in other settings. As an alternative, it is possible to construct an approximation in TT format by exploiting the connection of mean an variance of κ and its exponent γ [16] while also providing bounds for the approximation error in the Frobenius norm. The use of numerical quadrature in [16] is replaced by a block cross approximation algorithm, which constructs the TT format for κ from a few evaluations of the entry-wise formula in [17] to improve efficiency. A hybrid algorithm to approximate κ is presented in [18]. This algorithm iteratively approximates c via the *alternating linear scheme (ALS)* [19–21] and replaces the dense iterates by a sparse approximation obtained via cross approximation, which exploits and preserves the block diagonal structure of the discretized operator in stochastic collocation schemes. A drawback of numerical quadrature is that a large number of function evaluations is necessary or strong regularity assumptions have to be made. Using cross approximation could be problematic because — to the knowledge of the authors — no convergence guarantees exist.

To circumvent the dimension independent but slow convergence rate of Monte Carlo sampling in Bayesian inference, measure transport approaches [22–24] gained a lot of popularity in recent years. These approaches allow for a fast generation of independent samples from the posterior, but generally do not supply a functional representation thereof. Recent results to gain such a functional representation in TT format are presented in [10, 11], which is motivated by the low-rank representability of Gaussian densities [25]. However, the approximation task itself can be quite challenging since one needs to guarantee small Lipschitz constants of the transport map [26] to achieve numerical stability.

The above mentioned methods for the representation of an exponential are either highly intrusive or are at least tailored to a specific problem structure. Furthermore, a direct approach of a solely sample based non-intrusive reconstruction using e.g. tensor recovery [27] or cross approximation [28, 29]

requires an infeasible amount of training data or an unrealistically good initial guess to obtain similar approximation results as the other methods.

In this paper we propose to compute the exponential of a function given in TT format by solving the system

$$\nabla u - u\,\nabla h = f, \quad u(y_0) = 0$$

of *ordinary differential equations* (ODE) via Galerkin projection. This makes it possible to derive an approximation of the exponential and establish error bounds via an equivalence of the discrete residuum and an energy norm. The system operator and right-hand side can be discretized efficiently in the TT format which enables to solve the problem using the ALS method. We restrict our examinations mainly to first order systems and to the representation of the exponential in the TT format. However, a specific choice of the tensor network or the ODE is not necessary and both may be adapted to other applications. In fact, we argue that our approach is not limited to the reconstruction of exponentials, but can be extended to other functions which constitute the solution of a linear homogeneous differential equation with polynomial coefficients, namely holonomic functions. These functions are a powerful tool in computer algebra since they are smooth, can be represented by a finite amount of data and satisfy several closure properties. By our work we extend these properties by adding an algorithmic representability in the TT format and hope that this will provide a new and practical way to represent functions in the ever more important field of high-dimensional numerical computations.

The rest of the paper is structured as follows. After introducing basic notation in Section 1.1 , Section 2 presents the general framework of our approach. We describe the system of differential equations that we solve and the realization of the Galerkin scheme for the univariate case. Subsequently, we define the energy norm and show equivalence of this norm to the discrete residual. We conclude the section by generalizing the univariate results to the multivariate case and establish a theoretical foundation to apply our approach to other types of holonomic functions. Section 3 recalls the basic workings of the ALS method and constructs a dimension independent low-rank representation of the system operator and right-hand side in the TT format. Additionally, we comment on the algorithmic realization of our method and give detail on some numerical intricacies in Section 3.3. Numerical applications are discussed in Section 4. There we explain our choice of discretization and error evaluation and give further details on the Darcy problem and Bayesian likelihoods before we discuss our numerical results. The paper concludes with an outlook on further applications of the suggested approach as well as possible future research.

1.1. **Preliminaries.** In the following we introduce the notation used throughout the paper. If not specified differently, sets are denoted by calligraphic letters. Denote by $[n] \subset \mathbb{N}_0 := \mathbb{N} \cup \{0\}$ the set of integers $\{0, \dots, n-1\}$. For sets $\mathcal{X}, \mathcal{Y}$, we denote by $C^{k,\alpha}(\mathcal{X}; \mathcal{Y})$ the space of k-times differentiable functions from $\mathcal{X}$ to $\mathcal{Y}$ with α-Hölder-continuous k^{th} derivative. Denote by $L^p(\mathcal{X}, \rho; \mathcal{Y})$ the weighted L^p-space for any $p \in \mathbb{N}$, weight function ρ, and two sets $\mathcal{X}, \mathcal{Y}$. The same notation is used for Sobolev spaces $W^{k,p}(\mathcal{X}, \rho; \mathcal{Y})$ and we write $H^k(\mathcal{X}, \rho; \mathcal{Y}) = W^{k,2}(\mathcal{X}, \rho; \mathcal{Y})$. If $\rho \equiv 1$ we omit the weight function and if $\mathcal{Y} = \mathbb{R}$, we omit the image space. The unit sphere of any set $\mathcal{X}$ is defined by

$$S(\mathcal{X}) := \{v \in \mathcal{X} \colon \|v\|_{\mathcal{X}} = 1\}.$$

For any Hilbert space $\mathcal{X}$ and any subspace $\mathcal{Y} \subseteq \mathcal{X}$, the orthogonal projection onto $\mathcal{Y}$ is denoted by $P_{\mathcal{Y}} : \mathcal{X} \to \mathcal{Y}$. We use standard notation for multiindices $\mu, \nu \in \mathbb{N}_0^M$ and additionally abbreviate sums over multiindices by

$$\sum_{\nu=1}^{\mu} := \sum_{\nu_1=1}^{\mu_1} \cdots \sum_{\nu_M=1}^{\mu_M}.$$

Regular letters are used to notate standard non-discretized operators and functions. The discretized versions of objects are denoted by boldface symbols. For any orthonormal set of functions $\{P_\mu\}_{\mu \in [d]^M} \subset$

$L^2(\mathbb{R}^M, \rho)$ we define the finite dimensional subspace

$$\mathcal{V}_d := \operatorname{span}\{P_\mu \colon \mu \in [d]^M\} \subset L^2(\mathbb{R}^M, \rho).$$

Then, functions $w \in \mathcal{V}_d$ can be expressed by

$$w(x) = \sum_{\mu \in [d]^M} \boldsymbol{w}[\mu] P_\mu(x) \qquad \text{with} \qquad \boldsymbol{w} \in \mathbb{R}^{d \times \cdots \times d}.$$

Hence, the finite dimensional space $\mathcal{V}_d$ is isomorphic to the space of coefficient tensors $\mathbb{R}^{d^M}$.

Note that the size of the coefficient tensor $\boldsymbol{w}$ grows exponentially with the order M. This is commonly referred to as the *curse of dimensionality*. To mitigate this exponential dependence on M, we employ a low-rank decomposition of the tensor $\boldsymbol{w}$. There are many tensor decompositions discussed in the literature [13, 30–32]. Mainly due to its simplicity and the wide availability in numerical libraries we have chosen the *tensor train* (TT) format for our derivations. It is one of the best-studied tensor formats in numerical mathematics (cf. [14, 21, 25, 33]). We can efficiently represent all the tensors in our method in this format and the employed optimization algorithm is has been used reliably in many applications. We want to stress however that the approach presented in this work can be applied to any other tensor decomposition.

In the following we provide a brief overview of the notation used with the tensor train format. For further details, we refer the reader to [33, 34] and the references therein. The TT representation of a tensor $\boldsymbol{w} \in \mathbb{R}^{d^M}$ is given as

$$\boldsymbol{w}[\mu] = \sum_{k=1}^{r} \prod_{m=1}^{M} \boldsymbol{w}_m[k_m, \mu_m, k_{m+1}] \qquad \text{for any } \mu \in [d]^M,$$

with order three component tensors $\boldsymbol{w}_m \in \mathbb{R}^{r_m \times d \times r_{m+1}}$. Here, $r = (r_0, \dots, r_{M+1})$, with $r_0 = r_{M+1} = 1$. If all ranks r_m are minimal, this is called *tensor train decomposition* of $\boldsymbol{w}$ with TT rank r. The degrees of freedom of a TT representation are defined by

$$\text{tt-dofs}(\boldsymbol{w}) = \sum_{m=1}^{M-1} \left(r_m d r_{m+1} - r_{m+1}^2 \right) + r_M d, \tag{1.2}$$

which shows that the complexity of tensor trains behaves like $\mathcal{O}(M d \hat{r}^2)$ for $\hat{r} = \max\{r_1, \dots, r_M\}$. In contrast to full tensor representations with complexity $\mathcal{O}(d^M)$, tensor trains depend only linearly on the order M. As a result, the TT format is especially efficient for a small maximal rank $\hat{r}$.

In a similar fashion, for any $d, q \in \mathbb{N}$ we can express linear operators $W \colon \mathcal{V}_q \to \mathcal{V}_d$ in the tensor train format. For this recall that the application of W to $v \in \mathcal{V}_q$ reads

$$Wv(x) = \sum_{\mu \in [d]^M} \sum_{\nu \in [q]^M} \boldsymbol{W}[\mu, \nu] \boldsymbol{v}[\nu] P_\mu(x).$$

The TT representation of the tensor operator $\boldsymbol{W} \colon \mathbb{R}^{q^M} \to \mathbb{R}^{d^M}$ is thus determined by

$$\boldsymbol{W}[\mu, \nu] = \sum_{k=1}^{r} \prod_{m=1}^{M} \boldsymbol{W}_m[k_m, \mu_m, \nu_m, k_{m+1}] \qquad \text{for any } \mu \in [d]^M \text{ and } \nu \in [q]^M,$$

with the order four component tensors $\boldsymbol{W}_m \in \mathbb{R}^{r_m \times d \times q \times r_{m+1}}$. The TT decomposition always exists and can be computed in polynomial time using the hierarchical singular value decomposition (SVD) [21]. A truncated hierarchical SVD leads to quasi-optimal approximations of the TT decomposition in the Frobenius norm [14, 35–37]. This can also be applied to tensors which are already represented in the TT format to obtain a TT decomposition with a lower rank. This process is referred to as *rounding*. Note that sums and products can be computed efficiently in the TT format [34],

which is crucial for the proposed method. Moreover, many of the occurring tensors are of the form $\boldsymbol{W} = \sum_{m=1}^{M} \boldsymbol{B}_m$, where, for any $\mu \in [d]^M$,

$$\boldsymbol{B}_m[\mu] = \sum_{k=1}^{r} \prod_{n=1}^{m-1} \boldsymbol{U}_n[k_n, \mu_n, k_{n+1}] \boldsymbol{C}_m[k_m, \mu_m, k_{m+1}] \prod_{n=m+1}^{M} \boldsymbol{V}_n[k_n, \mu_n, k_{n+1}].$$

Such tensors are said to have a *Laplace-like* structure [38] and are representable with a particularly low rank.

2. Approximation of exponentials via Galerkin projection

In this section we demonstrate how the exponential of a function can be approximated by using a Galerkin projection and derive computable bounds for the approximation error. However, the presented orthogonal projection as well as the derived a posteriori error bounds can in principle be adapted to a broader class of multivariate holonomic functions. These are functions that constitute solutions of a system of linear differential equations with polynomial coefficients as discussed in Section 2.3.

For any given exponent h, we construct a system of differential equations that has $\exp h$ as a unique solution and then use a Galerkin projection to construct an approximation to $\exp h$. This also allows us to harvest well established results of the Galerkin method to obtain an a posteriori error control of the approximation.

We start with the description of the approach for univariate functions, subsequently derive upper and lower bounds of the approximation error in terms of the residual, and eventually generalize our results to the multivariate case.

2.1. Approximation of univariate exponentials.

Let ρ be the standard Gaussian density and assume that the exponent in $C^1(\mathbb{R}) \cap L^2(\mathbb{R}, \rho)$ can be approximated by

$$h(y) = \sum_{j=0}^{d_h - 1} \boldsymbol{h}[j] p_j(y), \tag{2.1}$$

where $\{p_j\}_{j=0}^{\infty}$ form an orthonormal basis in $L^2(\mathbb{R}, \rho)$. Consider the linear initial value problem

$$\begin{aligned} u' - u\, h' &= 0, \\ u(y_0) &= \exp h(y_0), \end{aligned} \tag{2.2}$$

for an arbitrary $y_0 \in \mathbb{R}$. It is easy to verify that $u = \exp h$ is the unique solution to (2.2). For $f(y) = \exp(h(y_0))h'(y)$, the problem with inhomogeneous initial condition (2.2) is equivalent to

$$\begin{aligned} u' - u\, h' &= f, \\ u(y_0) &= 0, \end{aligned} \tag{2.3}$$

in the sense that u is the solution of (2.3) if and only if $u + \exp h(y_0)$ is the solution of (2.2). Although the choice of $y_0 \in \mathbb{R}$ is arbitrary, it is advisable to choose the initial point such that $\rho(y_0) \gg 0$ to avoid numerical precision issues. Because of this and for the sake of simplicity, we assume $y_0 = 0$ in the following.

Let $\mathcal{X} = \{u \in H^2(\mathbb{R}, \rho) \colon u(y_0) = 0\}$ and define $B(v) = v' - vh'$. The variational form of (2.3) then reads: Find $u \in \mathcal{X}$ such that

$$(B(u), v)_{\mathcal{V}} = (f, v)_{\mathcal{V}} \qquad \text{for all } v \in \mathcal{V}, \tag{2.4}$$

where $(\bullet, \bullet)_{\mathcal{V}}$ denotes the inner product in $\mathcal{V} := L^2(\mathbb{R}, \rho)$.

For the Galerkin approximation of (2.4), we define the ansatz space $\mathcal{V}_\mathrm{a} := \operatorname{span}\{p_j : j \in [d_a] \setminus \{0\}\}$ and the test space $\mathcal{V}_\mathrm{t} := \mathcal{V}_{d_t}$ for polynomial degrees $d_a, d_t \in \mathbb{N}_{>0}$. With these spaces we can define discretized versions of B and f by

$$\boldsymbol{B}_{ij} := (B(p_{j+1}), p_i)_\mathcal{V} \qquad \text{and} \qquad \boldsymbol{f}_i := (f, p_i)_\mathcal{V} \qquad i \in [d_\mathrm{t}],\ j \in [d_\mathrm{a}].$$

Consequently, the discretization of (2.4) reads

$$\boldsymbol{B}\boldsymbol{u} = \boldsymbol{f}.$$

Lemma 2.1. *The operator $B : \mathcal{X} \to \mathcal{V}$ and the discretized operator $\boldsymbol{B} : \mathbb{R}^{d_a} \to \mathbb{R}^{d_t}$ are injective for $d_\mathrm{t} > d_\mathrm{a}$, i.e. $\ker B = \{0\}$ and $\ker \boldsymbol{B} = \{0\}$.*

Proof. The first assertion holds if and only if the solution $u = 0$ of the ODE $B(u) = u' - h'u = 0$ is unique. Since $\mathcal{X} \subset C^{1,0}(\mathbb{R})$ we can consider this equation in the classical sense. The claim follows because h' is locally Lipschitz continuous and the ODE satisfies the conditions for the Picard–Lindelöf theorem.
To prove the second assertion, first assume that $\deg(h) = 0$. Then $Bu = u'$, which can only be zero when u is constant. Since $p_0 \notin \mathcal{V}_\mathrm{a}$ this can only be the case for $u = 0$. Now assume $\deg(h) > 0$. Then $\deg(h'u) \geq \deg(u)$ and $\deg(u') = \deg(u) - 1$. Consequently, $\deg(B(u)) \geq \deg(u)$ and $B(u) = 0$ implies $\deg(u) = 0$. The only polynomial of degree 0 in $\mathcal{V}_\mathrm{a}$ is 0. This concludes the proof. □

Lemma 2.1 demonstrates the advantage of considering the problem with homogeneous initial conditions (2.4). The condition $u(y_0) = \exp h(y_0)$ effectively reduces the dimension of the solution space by 1. Solving with homogeneous initial conditions over the space $\mathcal{X}$ makes this explicit and ensures that the operator B is injective. This is not the case when considering $B : H^1(\mathbb{R}, \rho) \to L^2(\mathbb{R}, \rho)$.

In the following we assume that $d_\mathrm{t} = d_\mathrm{a} + d_\mathrm{h} - 1$. This ensures that $B(u_\mathrm{a}) \in \mathcal{V}_\mathrm{t}$ for any $u_\mathrm{a} \in \mathcal{V}_\mathrm{a}$ and that $f \in \mathcal{V}_\mathrm{t}$. Since the bound $\deg(B(u)) \leq \deg(u) + \deg(h) - 1$ is sharp, this is the smallest natural number with this property. The resulting system $\boldsymbol{B}\boldsymbol{u} = \boldsymbol{f}$ is overdetermined and can only be solved in a least-squares sense. This can be obtained by performing a QR-factorization of the form

$$\boldsymbol{B} = \begin{pmatrix} \boldsymbol{Q} & \boldsymbol{Q}_\perp \end{pmatrix} \begin{pmatrix} \boldsymbol{R} \\ 0 \end{pmatrix}$$

and by solving the regular linear system $\boldsymbol{Q}^\mathsf{T}\boldsymbol{B}\boldsymbol{u} = \boldsymbol{Q}^\mathsf{T}\boldsymbol{f}$. From this it can be seen that solving $\boldsymbol{B}\boldsymbol{u} = \boldsymbol{f}$ is equivalent to a standard Galerkin method with the reduced test space $\tilde{\mathcal{V}}_\mathrm{t} = Q\mathcal{V}_\mathrm{t}$. This test space is optimal in the sense that it minimizes the residual over $\tilde{\mathcal{V}}_\mathrm{t}^\perp$.

2.2. **A posteriori error bounds.** We are now interested in relating the discrete residual of the variational form (2.4) to the error in an appropriate norm. As it turns out, the dynamical system (2.4) induces a norm which we may use for this purpose.

We begin by considering the following lemma about injective linear operators.

Lemma 2.2. *Let $\mathcal{V}$ be a normed space and $W : \mathcal{X} \to \mathcal{V}$ be an injective linear operator. Then $\|v\|_W := \|W(v)\|_\mathcal{V}$ defines a norm on $\mathcal{X}$.*

Proof. Absolute homogeneity and the triangle inequality follow directly from the linearity of W and from the fact that $\| \bullet \|_\mathcal{V}$ is a norm. To show positive definiteness, let $v \in \mathcal{X}$ be such that $\|W(v)\|_\mathcal{V} = 0$. With the injectivity of W, which yields $\ker W = \{0\}$, there directly follows $v \equiv 0$. □

Since B is injective by Lemma 2.1, it follows from Lemma 2.2 that $\| \bullet \|_B$ is a norm. In the broader context of variational methods for elliptic PDEs, we refer to this norm as the *energy norm* induced by the system (2.4). With this we are subsequently able to prove that the discrete residual $\|\boldsymbol{f} - \boldsymbol{B}\boldsymbol{u}_\mathrm{a}\|_2$ is

equivalent to the error $\|u-u_\mathrm{a}\|_B$ in the energy norm for any discrete object $u_\mathrm{a} \in \mathcal{V}_\mathrm{a}$. This is important for mainly two reasons. First, by minimizing the discrete residual in the least-squares sense, we get a guarantee that the distance to the exact solution in the energy norm is minimized as well. Second, for any discrete approximation $u_\mathrm{a} \in \mathcal{V}_\mathrm{a}$, the discrete residual represents a reliable and efficient estimator for the approximation error in the energy norm. This in principle allows to adaptively control the number of steps of an iterative solver without any computational overhead.

To show the equivalence, we first establish the relation of the discrete and continuous residual in the following lemma.

Lemma 2.3. *Let $\mathcal{V}$ be a Hilbert space and $\mathcal{V}_\mathrm{t} \subseteq \mathcal{V}$. It holds for any $R \in \mathcal{V}^*$ that*

$$\|R\|_{\mathcal{V}_\mathrm{t}^*} = \|P_{\mathcal{V}_\mathrm{t}} r\|_{\mathcal{V}},$$

where $r \in \mathcal{V}$ denotes the Riesz representative of R in $\mathcal{V}$ and $P_{\mathcal{V}_\mathrm{t}}$ is the orthogonal projection $\mathcal{V} \to \mathcal{V}_\mathrm{t}$.

Proof. With the orthogonal projection $P_{\mathcal{V}_\mathrm{t}}$ and the dual pairing $\langle \bullet, \bullet \rangle_{\mathcal{V}_\mathrm{t}^*, \mathcal{V}_\mathrm{t}}$, we have

$$\|R\|_{\mathcal{V}_\mathrm{t}^*} = \sup_{v \in S(\mathcal{V}_\mathrm{t})} |\langle R, v \rangle_{\mathcal{V}_\mathrm{t}^*, \mathcal{V}_\mathrm{t}}| = \sup_{v \in S(\mathcal{V}_\mathrm{t})} |\,(r, v)_{\mathcal{V}}\,| = \sup_{v \in S(\mathcal{V}_\mathrm{t})} |\,(P_{\mathcal{V}_\mathrm{t}} r, v)_{\mathcal{V}}\,|.$$

Since the supremum in the last equation is attained for $v = \|P_{\mathcal{V}_\mathrm{t}} r\|_{\mathcal{V}}^{-1} P_{\mathcal{V}_\mathrm{t}} r \in S(\mathcal{V}_\mathrm{t})$, this directly shows the claim. □

As a consequence, it holds that $\|\mathcal{R}(v_\mathrm{a})\|_{\mathcal{V}_\mathrm{t}^*} = \|\boldsymbol{f} - \boldsymbol{B}\boldsymbol{v}_\mathrm{a}\|_2$, where for any $w \in \mathcal{X}$, $\mathcal{R}(w) := (f - B(w), \bullet)_{\mathcal{V}} \in \mathcal{V}^*$ is the residual of (2.4). What remains is to prove the equivalence of the continuous residual and the energy error, which is achieved with the following theorem.

Theorem 2.4. *Let $u \in \mathcal{X}$ be the unique solution of* (2.4)*, let $v_\mathrm{a} \in \mathcal{V}_\mathrm{a}$ be arbitrary and assume that $B(\mathcal{V}_\mathrm{a}) \subseteq \mathcal{V}_\mathrm{t}$. Then it holds that*

$$\|\mathcal{R}(v_\mathrm{a})\|_{\mathcal{V}^*} = \|u - v_\mathrm{a}\|_B \quad \textit{and} \quad \|\mathcal{R}(v_\mathrm{a})\|_{\mathcal{V}_\mathrm{t}^*} \leq \|u - v_\mathrm{a}\|_B \leq \|\mathcal{R}(v_\mathrm{a})\|_{\mathcal{V}_\mathrm{t}^*} + \|P_{\mathcal{V}_\mathrm{t}^\perp} f\|_{\mathcal{V}}.$$

Proof. The first assertion follows from the definition of the energy norm and the residual, i.e.

$$\|\mathcal{R}(v_\mathrm{a})\|_{\mathcal{V}^*} = \|r\|_{\mathcal{V}} = \|B(u - v_\mathrm{a})\|_{\mathcal{V}} = \|u - v_\mathrm{a}\|_B,$$

where $r = f - B(v_\mathrm{a})$ is the Riesz representative of the residual $\mathcal{R}(v_\mathrm{a})$ in $\mathcal{V}$. To show the first inequality of the second assertion, note that Lemma 2.3 directly yields

$$\|\mathcal{R}(v_\mathrm{a})\|_{\mathcal{V}_\mathrm{t}^*} = \|P_{\mathcal{V}_\mathrm{t}} r\|_{\mathcal{V}} = \|P_{\mathcal{V}_\mathrm{t}} B(u - v_\mathrm{a})\|_{\mathcal{V}} \leq \|B(u - v_\mathrm{a})\|_{\mathcal{V}} = \|u - v_\mathrm{a}\|_B.$$

The second inequality holds since $B(v_\mathrm{a}) \in \mathcal{V}_\mathrm{t}$, which implies $(r, P_{\mathcal{V}_\mathrm{t}^\perp} v)_{\mathcal{V}} = (f, P_{\mathcal{V}_\mathrm{t}^\perp} v)_{\mathcal{V}}$ and thus

$$\|\mathcal{R}(v_\mathrm{a})\|_{\mathcal{V}^*} \leq \sup_{v \in S(\mathcal{V})} |\,(r, P_{\mathcal{V}_\mathrm{t}} v)_{\mathcal{V}}\,| + \sup_{v \in S(\mathcal{V})} |(r, P_{\mathcal{V}_\mathrm{t}^\perp} v)_{\mathcal{V}}| = \|\mathcal{R}(v_\mathrm{a})\|_{\mathcal{V}_\mathrm{t}^*} + \|P_{\mathcal{V}_\mathrm{t}^\perp} f\|_{\mathcal{V}}. \qquad \square$$

Remark 2.5. *Theorem 2.4 is similar to well known results for a posteriori error control in the context of elliptic PDEs [39–41] in the sense that estimating the residual in the dual norm of the discrete space $\mathcal{V}_\mathrm{t}$ introduces an additional data oscillation term. To guarantee the efficiency of the residual estimator, the right-hand side f hence has to be resolved adequately. However, even if the data oscillation fails to be of higher order, it is always strictly efficient in the sense that*

$$\|P_{\mathcal{V}_\mathrm{t}^\perp} f\|_{\mathcal{V}} = \|P_{\mathcal{V}_\mathrm{t}^\perp} B(u - v_\mathrm{a})\|_{\mathcal{V}} \leq \|u - v_\mathrm{a}\|_B.$$

Note that in our setting the right-hand side f can be chosen rather freely. Hence, without loss of generality we may assume that the data oscillation term can be neglected in applications. Indeed, for the choice $f(y) = \exp h(y_0) h'(y)$ it holds that $f \in \mathcal{V}_\mathrm{t}$ and thus $\|P_{\mathcal{V}_\mathrm{t}^\perp} f\|_{\mathcal{V}} = 0$.

The following is an observation on how certain properties of the exponent h such as regularity influence the boundedness of the energy error with respect to other more meaningful or practical norms.

Corollary 2.6. *Let $u \in \mathcal{X}$ be the solution of* (2.4) *and let $h \in W^{1,\infty}(\mathbb{R})$. Then there exists $C > 0$ such that*

$$\|u - v_\mathrm{a}\|_B \leq C\|u - v_\mathrm{a}\|_{H^1(\mathbb{R},\rho)} \qquad \text{for all } v_a \in \mathcal{V}_\mathrm{a}.$$

If additionally there exists $\varepsilon > 0$, such that either $h'(y) \leq \frac{y}{2} + \varepsilon$ or $\frac{y}{2} + \varepsilon \leq h'(y)$ for all $y \in \mathbb{R}$, then for $c = \frac{1}{\varepsilon}$ it holds that

$$\|u - v_a\|_{L^2(\mathbb{R},\rho)} \leq c\|u - v_\mathrm{a}\|_B \qquad \text{for all } v_a \in \mathcal{V}_\mathrm{a}.$$

Proof. The upper bound follows directly from the definition of the H^1 norm and the essential boundedness of h' for $C = \sqrt{2}\max\{1, \|h'\|_{L^\infty(\mathbb{R})}\}$. To show the lower bound, let $\hat{c}(y) = \frac{y}{2} - h'(y)$. From the assumptions on h' it follows that $|\hat{c}(y)| \geq \varepsilon$ for all $y \in \mathbb{R}$. Integrating by parts yields

$$(v', v)_{L^2(\mathbb{R},\rho)} = (\rho, v'v) = \frac{1}{2}\left(\rho, (v^2)'\right) = -\frac{1}{2}\left(\rho', v^2\right) = \frac{1}{2}\left(y, v^2\right)_{L^2(\mathbb{R},\rho)}.$$

In combination with the boundedness of $\hat{c}$ this implies

$$|\,(B(v), v)_{L^2(\mathbb{R},\rho)}\,| = |\,\left(\hat{c}(y), v^2\right)_{L^2(\mathbb{R},\rho)}\,| \geq \varepsilon\|v\|^2_{L^2(\mathbb{R},\rho)}.$$

Since $\mathcal{V} = L^2(\mathbb{R}, \rho)$, the residual can be bounded from below. For any $v \in \mathcal{X}$ it holds that

$$\begin{aligned}\|\mathcal{R}(v)\|_{\mathcal{V}^*} &= \sup_{w\in\mathcal{V}\setminus\{0\}} \frac{|\,(B(u-v), w)_\mathcal{V}\,|}{\|w\|_\mathcal{V}} \geq \frac{|\,(B(u-v), u-v)_\mathcal{V}\,|}{\|u-v\|_\mathcal{V}} \\ &\geq \varepsilon\|u - v\|_\mathcal{V}.\end{aligned}$$

Theorem 2.4 then concludes the proof with $\|u - v_a\|_{L^2(\mathbb{R},\rho)} \leq \varepsilon^{-1}\|\mathcal{R}(v_\mathrm{a})\|_{\mathcal{V}^*} = \varepsilon^{-1}\|u - v_\mathrm{a}\|_B$ for all $v_\mathrm{a} \in \mathcal{V}_\mathrm{a}$. □

Remark 2.7. *The choice of the dynamical system for the approximation of u is not unique and it determines the induced energy norm. A different choice of dynamical system may thus lead to more reasonable assumptions on the exponent h then suggested by Corollary 2.6 to obtain bounds of the energy error by different norms.*

To illustrate this, consider for $x \in D = (0, 1)$ the second order ODE

$$\begin{aligned} u'' &= (h'' + (h')^2)\, u \qquad &&\text{in } D, \\ u(y) &= \exp h(y) &&\text{on } \partial D. \end{aligned}$$

Homogenization and standard arguments for elliptic PDEs yield $\|u - v_a\|_B \approx \|u - v_a\|_{H^1_0}$ for all $v_a \in \mathcal{V}_\mathrm{a} \subset \mathcal{X} = H^1_0(D)$ if there exist $0 < \check{h} \leq \hat{h} < \infty$ such that $\check{h} \leq (h'' + (h')^2) \leq \hat{h}$. This is the case for many affine and quadratic exponents.

2.3. **Generalization to multivariate exponentials.** In the multivariate setting we assume that for some $M, d_h \in \mathbb{N}$ the exponent h is given by an expansion

$$h(y) = \sum_{\mu\in[d_\mathrm{h}]^M} \boldsymbol{h}[\mu]P_\mu(y), \quad \text{where} \quad P_\mu(y) = \prod_{m=1}^{M} p_{\mu_m}(y_m). \tag{2.5}$$

Here, the orthonormal basis $\{P_\mu\}_{\mu\in\mathbb{N}_0^M}$ of the space $L^2(\mathbb{R}^M, \varrho)$ for $\varrho(y) = \prod_{m=1}^M \rho(y_m)$ is chosen as the tensorization of the univariate orthonormal basis $\{p_j\}_{j\in\mathbb{N}_0}$ of $L^2(\mathbb{R}; \rho)$.

The aim of this section is to generalize the univariate results from Section 2.2 to the multivariate setting by considering a *gradient system* of differential equations.

Definition 2.8 (Gradient system)**.** *A system of first order linear differential equations of the form*

$$\textit{Find } u \in C^1(\mathbb{R}^M) \textit{ such that} \quad \nabla u(y) = A(y)u(y) + F(y),$$

with $A, F \in C(\mathbb{R}^M; \mathbb{R}^M)$ *is called a* gradient system *with* M component equations

$$\partial_m u(y) = A_m(y)u(y) + F_m(y).$$

For $M = 2$, the simple example $A \equiv 0$ and $F(y_1, y_2) = (y_2 \ 0)^\intercal$ shows that an arbitrary gradient system may not have a solution. However, the following theorem guarantees that the existence of a solution to a gradient system implies its uniqueness under suitable assumptions.

Theorem 2.9. *Consider the gradient system* $\nabla u(y) = A(y)u(y) + F(y)$ *on the domain* Ω *with initial condition* $u(y^0) = 0$*. Assume that* Ω *is star-shaped with center* y^0 *and that* $\sup_{y\in\tilde{\Omega}} \|A(y)\| < \infty$ *for any closed subset* $\tilde{\Omega} \subseteq \Omega$*. Then, if there exists a classical solution* u *to the gradient system, it is unique.*

Proof. To prove uniqueness, let u_1 and u_2 be two solutions to the gradient system. For an arbitrary $y \in \mathbb{R}^M$ and $m = 0, \dots, M$ define $y^m[t] := (y_1, \dots, y_{m-1}, t, y^0_{m+1}, \dots, y^0_M)$ and $y^m := y^m[y_m]$ and note that $y^{m+1}[y^0_m] = y^m$. Now consider the ordinary differential equation $\partial_t u(y^1[t]) = A_1(y^1[t]) + F_1(y^1[t])$ with initial condition $u(y^1[y^0_1]) = u(y^0) = 0$. The assumption that $\sup_{y\in\tilde{\Omega}} \|A(y)\| < \infty$ for any closed subset $\tilde{\Omega} \subseteq \Omega$ guarantees that the conditions of the Picard–Lindelöf theorem are satisfied and the solution is (globally) unique. Since both u_1 and u_2 satisfy the equation it follows that $u_1(y^1[t]) = u_2(y^1[t])$ for all $t \in \mathbb{R}$ and in particular $u_1(y^1) = u_2(y^1)$. This argument can be iterated. By considering the equation $\partial_t u(y^{m+1}[t]) = A_{m+1}(y^{m+1}[t]) + F_{m+1}(y^{m+1}[t])$ with initial condition $u(y^{m+1}[y^0_m]) = u(y^m) = u_1(y^m) = u_2(y^m)$ it follows that $u_1(y^{m+1}) = u_2(y^{m+1})$. Finally, $u_1(y) = u_1(y^M) = u_2(y^M) = u_2(y)$. This implies $u_1 \equiv u_2$ since $y \in \mathbb{R}^M$ was arbitrary. □

We now define the multivariate formulation of (2.3) as

$$\begin{aligned} \nabla u - u\,\nabla h &= f, \\ u(y_0) &= 0, \end{aligned} \tag{2.6}$$

where we choose $f(y) = \exp(h(y_0))\,\nabla h(y)$ and set $y_0 = 0 \in \mathbb{R}^M$ as before. Observe that $u(y) = \exp h(y)$ is a classical solution of (2.6) and since the gradient system satisfies the conditions of Theorem 2.9, this solution is unique. For $k = \lceil \frac{M}{2} \rceil + 1$, let $\mathcal{X} = \{u \in H^k(\mathbb{R}^M, \varrho) \colon u(y_0) = 0\}$ and $\mathcal{V} = L^2(\mathbb{R}^M, \varrho)^M$ and define the operator $B : \mathcal{X} \to \mathcal{V}$ by $B(v) = \nabla v - v\,\nabla h$. Here, $\lceil x \rceil = \min\{n \in \mathbb{Z} \colon n \geq x\}$ denotes the ceiling function. The variational form of this equation then reads: Find $u \in \mathcal{X}$ such that

$$(B(u)_m, v)_{L^2(\mathbb{R}^M,\varrho)} = (f_m, v)_{L^2(\mathbb{R}^M,\varrho)} \qquad \text{for all } m \in [M] \text{ and } v \in L^2(\mathbb{R}^M, \varrho). \tag{2.7}$$

To formulate the Galerkin approximation of equation (2.7), we define the ansatz space $\mathcal{V}_\mathrm{a}$ by

$$\mathcal{V}_\mathrm{a} := \operatorname{span}\{P_\mu \,:\, \mu \in [d_\mathrm{a}]^M \setminus \{0\}\},$$

and set $\mathcal{V}_{d_t}$ as test space. Note that $\mathcal{V}_{d_t}$ is the test space for the operators B_m. The test space for the complete operator B is given by the Cartesian product $\mathcal{V}_\mathrm{t} := \mathcal{V}_{d_t}^M$. As above, we denote the discretized versions of u by $\boldsymbol{u}$ and define the discretization of B and f as

$$\boldsymbol{B} = \begin{pmatrix} \boldsymbol{B}_1 & \dots & \boldsymbol{B}_M \end{pmatrix}^\intercal \qquad \text{and} \qquad \boldsymbol{f} = \begin{pmatrix} \boldsymbol{f}_1 & \dots & \boldsymbol{f}_M \end{pmatrix}^\intercal$$

for

$$\boldsymbol{B}_m[\mu, \nu] = (B(P_\nu)_m, P_\mu)_{L^2(\mathbb{R}^M,\varrho)} \qquad \text{and} \qquad \boldsymbol{f}_m[\mu] = (f_m, P_\mu)_{L^2(\mathbb{R}^M,\varrho)},$$

where $\mu \in [d_\mathrm{t}]^M$ and $\nu \in [d_\mathrm{a}]^M \setminus \{0\}$.

Remark 2.10. *Observe that by the Sobolev inequality we have $H^k(U) \hookrightarrow C^1(U)$ for $k = \lceil \frac{M}{2} \rceil + 1$ and that $Z_U := \inf_{y\in U} \varrho(y) > 0$ for any bounded open subset of $\mathbb{R}^M$ with C^1 boundary. Then, for any $f \in H^k(\mathbb{R}^M, \varrho)$,*

$$\|f\|^2_{H^k(U)} = \sum_{|\mu|\leq k} \int_U |f^{(\mu)}(y)|^2 \,\mathrm{d}y \leq \sum_{|\mu|\leq k} \int_U |f^{(\mu)}(y)|^2 Z_U^{-1} \varrho(y) \,\mathrm{d}y = Z_U^{-1} \|f\|^2_{H^k(U,\varrho)}.$$

Thus, $f \in H^k(U)$ and consequently $f \in C^1(U)$ for any U in a countable covering of $\mathbb{R}^M$ by open sets with C^1 boundary. This shows that $f \in C^1(\mathbb{R}^M)$ and hence every function in $\mathcal{X}$ is differentiable in the classical sense. This means that the weak solution of (2.7) *coincides with the classical solution and that B is injective. Choosing $d_t \geq d_a + d_h - 1$ ensures $B(u_\mathrm{a}) \in \mathcal{V}_\mathrm{t}$ for all $u_\mathrm{a} \in \mathcal{V}_\mathrm{a}$. Then $\boldsymbol{B}$ is the matrix representation of the restriction of B onto $\mathcal{V}_\mathrm{a}$ and thus injective.*

Remark 2.10 guarantees that the energy norm is well-defined and implies that $\|\mathcal{R}(v_\mathrm{a})\|_{\mathcal{V}_\mathrm{t}^*} = \|\boldsymbol{f} - \boldsymbol{B}\boldsymbol{v}_\mathrm{a}\|_2$ where the residual is again defined by $\mathcal{R}(v_\mathrm{a}) := (f - B(v_\mathrm{a}), \bullet)_\mathcal{V}$. To show equivalence of the residual to the energy norm and to obtain a posteriori error control note that Lemma 2.3 and Theorem 2.4 also hold for the multivariate case.

To conclude this section we note that it is possible to generalize Theorem 2.9 to a larger set of what we refer to as multivariate holonomic functions, which is shown in the following proposition.

Proposition 2.11. *Let w be a holonomic function and h be a polynomial. Then $w \circ h$ is the unique solution to a gradient system.*

Proof. Recall that a holonomic function w of order r is the solution to an rth order homogeneous linear differential equation with polynomial coefficients. This means that there exist matrices $A(t), B(t) \in \mathbb{R}^{r\times r}$ such that

$$A(t)v'(t) + B(t)v(t) = 0$$

and $w = v_1$. Let $\partial_m f(y)$ denote the partial derivative of the function f with respect to y_m in y and define $V := v \circ h$. Then $\partial_m V(y) = \partial_m h(y) v'(h(y))$ for any $y \in \mathbb{R}^M$. This means that V is the solution to the system of ordinary differential equations

$$A(h(y))\partial_m V(y) + \partial_m h(y) B(h(y)) V(y) = 0.$$

To show uniqueness, let $\xi \in C^1(\mathbb{R}; \mathbb{R}^M)$ and observe that the preceding system of equations implies

$$A(h(\xi(t)))\partial_m V(\xi(t))\xi_m'(t) + \partial_m h(\xi(t)) B(h(\xi(t))) V(\xi(t))\xi_m'(t) = 0$$

for all $m = 1, \dots, M$. Summing over m, the equation can be reformulated equivalently as

$$A(h(\xi(t)))(V(\xi(t)))' + (h(\xi(t)))' B(h(\xi(t)))V(\xi(t)) = 0$$

or in shorter notation as $[A \circ h \circ \xi][V \circ \xi]' + [h \circ \xi]'[B \circ h \circ \xi][V \circ \xi] = 0$. This is a first order homogeneous linear differential equation for the function $V \circ \xi = v \circ [h \circ \xi]$. If h is a polynomial and if $\xi(t) = \xi_y(t) := y_0 + (y - y_0)t$, this is a first order homogeneous linear differential equation with polynomial coefficients and exhibits a unique solution. Thus, if we are given an initial condition $V(y_0) = V_0$ and we set the initial condition $[V \circ \xi](0) = V_0$ then $V(y) = [V \circ \xi](1)$ is uniquely defined for any $y \in \mathbb{R}^M$. □

3. Low-rank representation of the operator equation

This section is concerned with a low-rank discretization of the multivariate system $\boldsymbol{B}\boldsymbol{u} = \boldsymbol{f}$ in order to make computations become feasible. We first briefly illustrate how the ALS can be employed to solve the high-dimensional linear system $\boldsymbol{B}\boldsymbol{u} = \boldsymbol{f}$ in the TT format.

3.1. **The Alternating Linear Scheme.** For the representation of a function $v_\mathrm{a} \in \mathcal{V}_\mathrm{a}$ in the tensor train format, we use the space $\mathcal{V}_{d_\mathrm{a}}$ and enforce the homogenous boundary condition $v_\mathrm{a}(y_0) = 0$ via a regularizer. Since the initial point y_0 can be chosen arbitrarily, it would be disproportionate to strictly enforce the condition $u(y_0) = 0$ when solving $\boldsymbol{B}\boldsymbol{u} = \boldsymbol{f}$. Instead, we incorporate the initial condition by means of a regularization term.

Denote by $\boldsymbol{P} \in \mathbb{R}^{d_a^M}$ the vector of basis functions evaluated in y_0, i.e. $\boldsymbol{P}[\mu] := P_\mu(y_0)$ for any $\mu \in [d_a]^M$. Since $\|\boldsymbol{B}\boldsymbol{u} - \boldsymbol{f}\|_2^2 = \sum_{m=1}^M \|\boldsymbol{B}_m\boldsymbol{u} - \boldsymbol{f}_m\|_2^2$, the regularized problem reads

$$\operatorname*{argmin}_{\boldsymbol{u}\in\mathbb{R}^{d_a^M}} \quad \|\boldsymbol{P}\boldsymbol{u}\|_2^2 + \sum_{m=1}^M \|\boldsymbol{B}_m\boldsymbol{u} - \boldsymbol{f}_m\|_2^2. \tag{3.1}$$

Note that the convergence of the residual implies that the initial condition is enforced. Inspired by the ALS, this functional can be minimized in an alternating fashion. Recall from Section 1.1 that the tensor $\boldsymbol{u}$ can be written in the tensor train format as $\boldsymbol{u} := \boldsymbol{Q}_k\boldsymbol{C}_k$ where $\boldsymbol{C}_k$ is the kth component tensor and $\boldsymbol{Q}_k$ an operator that represents the contraction of this component tensor with the remaining tensor network. The ALS then solves (3.1) by optimizing

$$\operatorname*{argmin}_{\boldsymbol{C}_k\in\mathbb{R}^{r_k\times d_a\times r_{k+1}}} \quad \|\boldsymbol{P}\boldsymbol{Q}_k\boldsymbol{C}_k\|_2^2 + \sum_{m=1}^M \|\boldsymbol{B}_m\boldsymbol{Q}_k\boldsymbol{C}_k - \boldsymbol{f}_m\|_2^2 \tag{3.2}$$

cyclically for each $k = 1, \dots, M$, until some convergence criterion is satisfied. Each cycle is referred to as an ALS iteration step or sweep. The first order optimality condition of the optimization problem (3.2) reads

$$\left(\boldsymbol{Q}_k^\mathsf{T}\boldsymbol{P}^\mathsf{T}\boldsymbol{P}\boldsymbol{Q}_k + \sum_{m=1}^M \boldsymbol{Q}_k^\mathsf{T}\boldsymbol{B}_m^\mathsf{T}\boldsymbol{B}_m\boldsymbol{Q}_k\right)\boldsymbol{C}_k = \sum_{m=1}^M \boldsymbol{Q}_k^\mathsf{T}\boldsymbol{B}_m^\mathsf{T}\boldsymbol{f}_m.$$

Given the operator and right-hand side

$$\boldsymbol{W} := \boldsymbol{P}^\mathsf{T}\boldsymbol{P} + \sum_{m=1}^M \boldsymbol{B}_m^\mathsf{T}\boldsymbol{B}_m \qquad \text{and} \qquad \boldsymbol{b} := \sum_{m=1}^M \boldsymbol{B}_m^\mathsf{T}\boldsymbol{f}_m, \tag{3.3}$$

it is easy to find the minimum in (3.2) by cyclically solving the linear system

$$\boldsymbol{Q}_k^\mathsf{T}\boldsymbol{W}\boldsymbol{Q}_k\boldsymbol{C}_k = \boldsymbol{Q}_k^\mathsf{T}\boldsymbol{b} \tag{3.4}$$

for each $k = 1, \dots, M$.

3.2. **Low-rank representation of operator and right-hand side.** To construct an efficient representation of $\boldsymbol{W}$ and $\boldsymbol{b}$, we start by assembling $\boldsymbol{B}_m$ and $\boldsymbol{f}_m$. For this we first define the partial derivative operator

$$\boldsymbol{D}_m = \boldsymbol{I}^{\otimes(m-1)} \otimes \boldsymbol{D} \otimes \boldsymbol{I}^{\otimes(M-m)} \tag{3.5}$$

with the univariate differentiation operator $\boldsymbol{D}[i,j] := (p_i, p_j')_\mathcal{V}$. Now assume that the coefficient tensor $\boldsymbol{h}$ of (2.5) can be represented in the TT format as

$$\boldsymbol{h}[\mu] = \sum_{k=1}^r \prod_{j=1}^M \boldsymbol{h}_j[k_j, \mu j, k_{j+1}]$$

and define the multiplication operator

$$\boldsymbol{H}_m[\mu,\nu] = \sum_{k=1}^r \prod_{j=1}^M \boldsymbol{H}_{m,j}[k_j, \mu_j, \nu_j, k_{j+1}] \tag{3.6}$$

with the component tensors

$$\boldsymbol{H}_{m,j}[k_j,\mu_j,\nu_j,k_{j+1}] = \sum_{i_1=1}^{d_\mathrm{h}} \boldsymbol{\tau}[\mu_j,\nu_j,i_1]\boldsymbol{h}_j[k_j,i_1,k_{j+1}] \qquad \text{for } j \neq m \text{ and} \tag{3.7}$$

$$\boldsymbol{H}_{m,m}[k_m,\mu_m,\nu_m,k_{m+1}] = \sum_{i_1=1}^{d_\mathrm{h}}\sum_{i_2=1}^{d_\mathrm{h}} \boldsymbol{\tau}[\mu_m,\nu_m,i_1]\boldsymbol{D}[i_1,i_2]\boldsymbol{h}_m[k_m,i_2,k_{m+1}], \tag{3.8}$$

where $\boldsymbol{\tau}[i,j,k] = (p_i p_j, p_k)_{\mathcal{V}}$ denotes the triple product tensor. Since the operators (3.5) are of rank 1 and the operators (3.6) are of rank r, the operators $\boldsymbol{B}_m := \boldsymbol{D}_m - \boldsymbol{H}_m$ are of rank $(r+1)$ and $\boldsymbol{f}_m := \boldsymbol{D}_m \boldsymbol{h}$ are of rank r. For the initial condition, note that $\boldsymbol{P} = \boldsymbol{P}_1 \otimes \cdots \otimes \boldsymbol{P}_M$ constitutes a rank one tensor with component tensors $\boldsymbol{P}_j = (p_0(y_{0,j}),\dots,p_{d_a}(y_{0,j}))^\intercal$.

A naive computation of the sums in (3.3) would result in representation ranks that increase linearly in the number of parameters M. However, we can exploit the structure of the operators $\boldsymbol{W}$ and right-hand side $\boldsymbol{b}$ which resembles the structure of Laplace-like operators [38]. This allows to bound the ranks of $\boldsymbol{W}$ and $\boldsymbol{b}$ independent of the dimension. For any $j = 1,\dots,M$ and all $m_1, m_2 \in [M]\setminus\{j\}$, note that the component tensors of $\boldsymbol{B}_{m_1}$ and $\boldsymbol{B}_{m_2}$ satisfy $\boldsymbol{B}_{m_1,j} = \boldsymbol{B}_{m_2,j}$. To emphasize this, we write $\boldsymbol{C}_j := \boldsymbol{B}_{m,j}$ for some $m \neq j$. Then the component tensors of $\boldsymbol{W}$ are given by

$$\boldsymbol{W}_1 = \begin{bmatrix} \boldsymbol{P}_1^\intercal \boldsymbol{P}_1 & \boldsymbol{B}_{1,1}^\intercal \boldsymbol{B}_{1,1} & \boldsymbol{C}_1^\intercal \boldsymbol{C}_1 \end{bmatrix}, \tag{3.9}$$

$$\boldsymbol{W}_j = \begin{bmatrix} \boldsymbol{P}_j^\intercal \boldsymbol{P}_j & 0 & 0 \\ 0 & \boldsymbol{C}_j^\intercal \boldsymbol{C}_j & 0 \\ 0 & \boldsymbol{B}_{j,j}^\intercal \boldsymbol{B}_{j,j} & \boldsymbol{C}_j^\intercal \boldsymbol{C}_j \end{bmatrix} \qquad \text{for } j = 2,\dots,M-1, \tag{3.10}$$

$$\boldsymbol{W}_M = \begin{bmatrix} \boldsymbol{P}_M^\intercal \boldsymbol{P}_M \\ \boldsymbol{C}_M^\intercal \boldsymbol{C}_M \\ \boldsymbol{B}_{M,M}^\intercal \boldsymbol{B}_{M,M} \end{bmatrix}. \tag{3.11}$$

From this it is easy to see that the rank of $\boldsymbol{W}$ is given by $2(r+1)^2 + 1$ and is thus independent of M. In an analogous way, the component tensors of $\boldsymbol{b}$ are given by

$$\boldsymbol{b}_1 = \begin{bmatrix} \boldsymbol{B}_{1,1}^\intercal \boldsymbol{f}_{1,1} & \boldsymbol{C}_1^\intercal \boldsymbol{g}_1 \end{bmatrix}, \quad \boldsymbol{b}_j = \begin{bmatrix} \boldsymbol{C}_j^\intercal \boldsymbol{g}_j & 0 \\ \boldsymbol{B}_{j,j}^\intercal \boldsymbol{f}_{j,j} & \boldsymbol{C}_j^\intercal \boldsymbol{g}_j \end{bmatrix} \quad \text{and} \quad \boldsymbol{b}_M = \begin{bmatrix} \boldsymbol{C}_M^\intercal \boldsymbol{g}_M \\ \boldsymbol{B}_{M,M}^\intercal \boldsymbol{f}_{M,M} \end{bmatrix}, \tag{3.12}$$

where again $j = 2,\dots,M-1$ and we again define $\boldsymbol{g}_j := \boldsymbol{f}_{m,j}$ for some $m \neq j$. This shows that $\boldsymbol{b}$ can be represented in the TT format with rank $2r(r+1)$ and is independent of the dimension M as well.

Finally, observe that the constant function $\exp h(y_0)$ can be represented by a TT tensor of rank one. This means that the TT representation of the solution $u + \exp h(y_0)$ to the original problem can be computed in a straight-forward manner and the rank will increase by at most one.

3.3. **Algorithmic realization.** In the following we discuss some intricacies that arise in the application of an ALS to compute the Galerkin approximation (3.4) in the TT format. The method itself is rather straight-forward and we provide pseudo-code in Algorithm 1. Investigating the algorithm, the method $\mathtt{ALSsweep}(\boldsymbol{W},\boldsymbol{b},\boldsymbol{u}_\mathrm{a})$ in line 8 realizes one complete sweep of the ALS algorithm (3.2), i.e. it solves the local linear system (3.4) for each component tensor $C_1,\dots,C_M$.

To assess the convergence of the algorithm, we need to compute the discrete residual $\|\boldsymbol{B}\boldsymbol{u} - \boldsymbol{f}\|_2$ (Lines 6 and 9). For this, the operator $\boldsymbol{B}$ and right-hand side $\boldsymbol{f}$ (Line 4) have to be assembled in addition to $\boldsymbol{W}$ and $\boldsymbol{b}$. The assembly can be performed in a dimension independent low-rank manner similar to (3.9)–(3.12) since $\boldsymbol{B}$ and $\boldsymbol{f}$ inherit the same Laplace-like structure as $\boldsymbol{W}$ and $\boldsymbol{b}$. In particular,

the TT components of $\boldsymbol{B}$ and $\boldsymbol{f}$ are given by

$$\boldsymbol{B}_{\mathrm{TT},1} = \begin{bmatrix} \boldsymbol{B}_{1,1} & \boldsymbol{C}_1 \end{bmatrix}, \quad \boldsymbol{B}_{\mathrm{TT},j} = \begin{bmatrix} \boldsymbol{C}_j & 0 \\ \boldsymbol{B}_{j,j} & \boldsymbol{C}_j \end{bmatrix} \quad \text{and} \quad \boldsymbol{B}_{\mathrm{TT},M} = \begin{bmatrix} \boldsymbol{C}_M \\ \boldsymbol{B}_{M,M} \end{bmatrix} \tag{3.13}$$

and

$$\boldsymbol{f}_{\mathrm{TT},1} = \begin{bmatrix} \boldsymbol{f}_{1,1} & \boldsymbol{g}_1 \end{bmatrix}, \quad \boldsymbol{f}_{\mathrm{TT},j} = \begin{bmatrix} \boldsymbol{g}_j & 0 \\ \boldsymbol{f}_{j,j} & \boldsymbol{g}_j \end{bmatrix} \quad \text{and} \quad \boldsymbol{f}_{\mathrm{TT},M} = \begin{bmatrix} \boldsymbol{g}_M \\ \boldsymbol{f}_{M,M} \end{bmatrix}. \tag{3.14}$$

Algorithm 1: Low-rank exponential approximation via Galerkin projection (`ExpTT`)

Input: TT representation of the exponent h; ansatz space dimension d_a; initial point y_0; stopping tolerance ε; maximum number of iterations N_{ITER}

Output: TT approximation $\boldsymbol{u}_a$ of $\exp h$; discrete relative residual res

1 Build operators $\boldsymbol{D}_m$ and $\boldsymbol{H}_m$ for $m = 1, \dots, M$ according to (3.5) and (3.7)–(3.8);
2 Use $\boldsymbol{D}_m$, $\boldsymbol{H}_m$ to assemble cores $\boldsymbol{B}_{m,m}$, $\boldsymbol{C}_m$, $\boldsymbol{P}_m$, $\boldsymbol{f}_{m,m}$ and $\boldsymbol{g}_m$;
3 Construct low-rank operator $\boldsymbol{W}$ and right-hand side $\boldsymbol{b}$ according to (3.9)–(3.12);
4 Construct low-rank operator $\boldsymbol{B}$ and right-hand side $\boldsymbol{f}$ according to (3.13)–(3.14);
5 Initialize random start vector $\boldsymbol{u}_a$ of dimension d_a;
6 Set $\mathrm{res} \leftarrow \|\boldsymbol{B}\boldsymbol{u}_a - \boldsymbol{f}\|_2 / \|\boldsymbol{f}\|_2$;
7 **for** $j = 1, \dots, N_{\mathrm{ITER}}$ **do**
8 $\quad \boldsymbol{u}_a \leftarrow \mathtt{ALSsweep}(\boldsymbol{W}, \boldsymbol{b}, \boldsymbol{u}_a)$;
9 $\quad \mathrm{res} \leftarrow \|\boldsymbol{B}\boldsymbol{u}_a - \boldsymbol{f}\|_2 / \|\boldsymbol{f}\|_2$;
10 $\quad$ **Break** if $\|\boldsymbol{W}\boldsymbol{u}_a - \boldsymbol{b}\|_2 \leq \varepsilon \|\boldsymbol{b}\|_2$;
11 Build constant TT tensor $\boldsymbol{c} = \exp h(y_0)$;
12 Set $\boldsymbol{u}_a \leftarrow \boldsymbol{u}_a + \boldsymbol{c}$;
13 **return** $\boldsymbol{u}_a$, res

Although our theory guarantees a (quasi-)best approximation for any polynomial exponent h, the resulting exponential might require large ansatz space dimensions and ranks. This is a general problem of approximation methods and results in larger memory requirements and increase computational costs.

We propose to circumvent this problem by utilizing a simple trick. For a given scaling $s \in \mathbb{N}_{>0}$, we apply Algorithm 1 to the scaled exponent $\tilde{h} := s^{-1}h$ and compute the sought exponential via $\exp(h(y)) = \exp(\tilde{h}(y))^s$. Since $\exp \circ \tilde{h}$ grows at a slower rate than $\exp \circ h$, this reduces the required ansatz space dimension. The resulting pseudo code is presented in Algorithm 2. Note that the output of Algorithm 2 exhibits the same ansatz space dimension as the one of Algorithm 1. Nonetheless, this approach should be preferred because of two reasons. First, the memory complexity of the TT tensor $\tilde{\boldsymbol{u}}_a$ depends only linearly on the dimension $\tilde{d}_{\mathrm{a}}$ while the complexity of the TT operator $\boldsymbol{W}$ depends quadratically on the dimension. This reduces the computational cost of applying Algorithm 1 in line 2 of Algorithm 2. Second, multiplication can be performed efficiently in the TT format and it is easy to balance accuracy and computational cost. This can be done by performing projections to lower dimensional discrete spaces or by rounding via a truncated SVD as in line 5 of Algorithm 2. Scaling h works well to reduce the ansatz space dimension and the rank of exponentials but may not work for other holonomic functions. However, we expect that similar tricks can be applied in these cases. For $\sin$ and $\cos$ for example a simple approach could be to reduce the frequency by approximating $u(h(sy))$ instead of $u(h(y))$ and to scale the basis functions afterwards.

Algorithm 2: Scaled ExpTT

Input: TT representation of the exponent h; approximation dimensions d_a; scaling number s; approximation dimensions for scaled problem $\tilde{d}_a$; initial condition y_0; stopping tolerance ε; rescaling tolerance ε_s; maximum number of iterations N_{ITER}

Output: TT approximation $\boldsymbol{u}_a$ of $\exp h$

Set $\tilde{h} \leftarrow s^{-1}h$;

Compute $\tilde{\boldsymbol{u}}_a, \mathrm{res} \leftarrow \mathtt{ExpTT}(\tilde{\boldsymbol{h}}, \tilde{d}_a, y_0, \varepsilon, N_{\mathrm{ITER}})$;

Set $\boldsymbol{u}_a \leftarrow \tilde{\boldsymbol{u}}_a$;

for $j = 1, \dots, s$ **do**

  $\boldsymbol{u}_a \leftarrow \boldsymbol{u}_a \cdot \tilde{\boldsymbol{u}}_a$;

  project $\boldsymbol{u}_a$ onto $\mathcal{V}_{d_a}$ and round to tolerance ε_s;

return $\boldsymbol{u}_a$

4. Numerical experiments

In this section we examine the numerical performance of the proposed Galerkin method to approximate the exponential of a function in TT format. To assess its practical potential, we investigate two benchmark problems common in Uncertainty Quantification.

First, we consider the reconstruction of a lognormal diffusion coefficient as it appears frequently when modeling the porosity in the prototypical Darcy equation. This problem exhibits several complications analytically [42] and numerically [43] and often is tackled with sampling techniques, in particular (multilevel) Monte Carlo methods [44, 45]. Functional (polynomial chaos) approaches were e.g. examined in [15] with an adaptive stochastic Galerkin FEM in TT format and in [27] with a randomized least squares tensor regression. Stochastic collocation was e.g. used in [46, 47].

Second, we consider the recovery of the likelihood in the context of Bayesian inverse problems the theory of which can e.g. be found in [5]. Usually, again sampling methods are used for this often high-dimensional problem, the most popular of which certainly is the Markov chain Monte Carlo method. Nevertheless, recently some developments took place which showed that functional approximations of (posterior) densities are feasible and may prove beneficial in terms of convergence rates [11, 48], see also [25, 49] for different low-rank techniques.

These experiments are similar to those performed in [15–18, 27] for the lognormal diffusion coefficient and to those in [11, 48] for the likelihood reconstruction. We compare the approximation accuracy and computational time of our method to the results of other techniques from the literature[1], namely [16–18].

As weight function ρ we choose the density of the standard Gaussian distribution. As a basis for the trial and test spaces as well as for the parametrization of the exponent function h, we employ normalized tensorized (probabilistic) Hermite polynomials. One reason for this is that the triple product tensor κ can be computed analytically [15, 50, 51], which increases the overall computational performance. Moreover, differentials of polynomials are explicitly known and thus cheap to compute and, in the case of Hermite polynomials, Hermite polynomials again.

Proposition 4.1. *For the normalized multivariate Hermite polynomials there holds*

$$\partial_m P_\nu = \sqrt{\nu_m} P_{\nu - e_m}, \qquad \text{for all } m = 1, \dots, M,$$

[1] It has to be noted that the runtime of the previously reported experiments cannot be compared directly to what we observe with our (unoptimized) implementation. We nevertheless think that this provides a useful indication of the required computational effort.

where e_m denotes the canonical unit vector in $\mathbb{R}^M$. Moreover, the univariate differentiation operator $\boldsymbol{D}$ is given analytically by $\boldsymbol{D}_{ij} = \sqrt{j}\delta_{i,j-1}$.

Proof. The normalized (probabilistic) Hermite polynomials of degree k are given for any $j = 1, \dots, M$ by $P_k^j = H_k/\sqrt{k!}$. Since the Hermite polynomials constitute an Appell sequence, i.e. $H_k' = kH_{k-1}$ for all $k \in \mathbb{N}$, it holds $(P_k^j)' = \sqrt{k}P_{k-1}^j$. Hence, $(P_k^j, (P_\ell^j)')_\rho = \sqrt{\ell}\delta_{k,\ell-1}$. The multiplicative structure of the tensorized Hermite polynomials completes the claim. □

We note that due to orthogonality of the polynomial basis, the triple product expansion is finite, i.e. there holds $\tau_{ijk} = 0$ for any $k > i + j$. This guarantees that we can choose ansatz and test spaces $\mathcal{V}_\mathrm{a}$ and $\mathcal{V}_\mathrm{t}$ such that the condition $B(\mathcal{V}_\mathrm{t}) \subseteq \mathcal{V}_\mathrm{t}$ of Theorem 2.4 is satisfied.

The finite element discretization is based on the open source package `FEniCS` [52] and all finite element computations use uniform triangulations of the unit square $D = [0, 1]^2$. The fully discretized problem $\boldsymbol{W}\boldsymbol{u} = \boldsymbol{b}$ is solved in the TT format using Algorithm 2, which relies on the TT representation and the ALS algorithm implemented in the open source tensor library `xerus` [53].

Our approach and the variational Monte Carlo (VMC) method [27] (a tensor regression technique employed several times in this section) rely on the choice of an initial guess. Since this is chosen randomly, the obtained approximation as well as the CPU time for the computation of both methods may vary slightly for repeated runs of the same experiment. Nevertheless, the deviations are minuscule and we hence refrain from a statistical assessment of the results.

Computation of the error. The relative discrete residual and (up to the data oscillation) the equivalent relative energy error is denoted by $\mathrm{res}(\boldsymbol{u}_a) := \|\boldsymbol{B}\boldsymbol{u}_a - \boldsymbol{f}\|_2\|\boldsymbol{f}\|_2^{-1}$. The residual is computed according to Algorithm 1 based on the solution of (2.7). If the exponent is scaled, i.e. if we apply Algorithm 2, the relative residual of the scaled version of (2.7) is considered. For comparison, we additionally compute the absolute and relative L^2-errors via a Monte Carlo estimation. For this, a set of N_MC independent samples $y^{(i)} \sim \mathcal{N}(0, I)$ is drawn. The approximate solution $u_a \in \mathcal{V}_a$ obtained by our algorithm evaluated in the samples $y^{(i)}$ is compared to the corresponding (deterministic) sampled solution $u(y^{(i)})$. The absolute and relative mean squared errors are approximated by a Monte Carlo quadrature for each $\hat{v} \in \mathcal{V}_a$,

$$\mathcal{E}_u(\hat{v}) = \frac{1}{N_\mathrm{MC}} \sum_{i=1}^{N_\mathrm{MC}} \|u(y^{(i)}) - \hat{v}(y^{(i)})\|_* \quad \text{and} \quad \varepsilon_u(\hat{v}) = \frac{1}{N_\mathrm{MC}} \sum_{i=1}^{N_\mathrm{MC}} \frac{\|u(y^{(i)}) - \hat{v}(y^{(i)})\|_*}{\|u(y^{(i)})\|_*}, \tag{4.1}$$

where $\| \bullet \|_*$ is either the absolute value if $u(y) \in \mathbb{R}$ or $\| \bullet \|_{L^2(D)}$ if $u(y) \in H_0^1(D)$. For the latter case, we additionally introduce the relative L^∞-error

$$\varepsilon_u^\infty(\hat{v}) = \frac{1}{N_\mathrm{MC}} \sum_{i=1}^{N_\mathrm{MC}} \frac{\|u(y^{(i)}) - \hat{v}(y^{(i)})\|_{L^\infty(D)}}{\|u(y^{(i)})\|_{L^\infty(D)}}, \tag{4.2}$$

to allow a comparison to results of previous works. The choice $N_\mathrm{MC} = 10^3$ proved to be sufficient to obtain reliable estimates in our experiments.

The random model problem. The experiments we investigate concern the stationary random diffusion problem as described in [54–56] on the unit square $D = [0, 1]^2$. Concretely, for almost all $y \in \mathbb{R}^M$ we consider the random elliptic problem

$$\begin{aligned} -\operatorname{div}(\kappa(x, y)\, \nabla w(x, y)) &= f(x), \qquad \text{in } D, \\ w(x, y) &= 0, \qquad \text{on } \partial D. \end{aligned} \tag{4.3}$$

For the sake of a clear presentation, the source term $f \in L^2(D)$ and the boundary conditions are assumed to be deterministic. The diffusion coefficient $\kappa\colon D \times \mathbb{R}^M \to \mathbb{R}$ is typically considered

lognormal and isotropic, i.e. $\log\kappa$ is an isotropic Gaussian random field [16]. Pointwise solvability of (4.3) for almost all $y\in\mathbb{R}^M$ is guaranteed by a Lax–Milgram argument in [42, 56]. Well-posedness of the variational parametric problem is way more intricate and requires a larger solution space. We refer to [56] for a detailed discussion. Following the lines of e.g. [3], we assume a truncated Karhunen-Loève expansion of the affine exponent $\gamma=\log\kappa$ of the form

$$\gamma(x,y)=\sum_{m=1}^{M}\gamma_m(x)y_m \qquad \text{for all } x\in D \text{ and almost all } y\in\mathbb{R}^M. \tag{4.4}$$

The expansion coefficient functions γ_m enumerate all planar Fourier sine modes in increasing total order and are given by

$$\gamma_m(x)=\frac{9}{10\zeta(\sigma)}m^{-\sigma}\cos\big(2\pi\beta_1(m)x_1\big)\,\cos\big(2\pi\beta_2(m)x_2\big), \tag{4.5}$$

where ζ is the Riemann zeta function and for $k(m)=\lfloor-\frac{1}{2}+\sqrt{\frac{1}{4}+2m}\rfloor$,

$$\beta_1(m)=m-k(m)\frac{k(m)+1}{2} \qquad \text{and} \qquad \beta_2(m)=k(m)-\beta_1(m).$$

For our experiments we set a slow decay rate of $\sigma=2$. For the deterministic discretization we choose either lowest order discontinuous Lagrange elements or continuous Lagrange elements. However, other finite elements can be used with only slight adaptations as well.

Bayesian log-likelihoods. This section gives a short review of the Bayesian approach to inverse problems. Its aim is to illustrate how our method can be used in this setting. A comprehensive description on the Bayesian perspective on inverse problems can e.g. be found in [5, 11, 57].

For an uncertain input $y\in\mathbb{R}^M$ consider the forward map

$$\hat{G}\colon\mathbb{R}^M\to H_0^1(D),\qquad y\mapsto w(y),$$

where the model output $w(y)\in H_0^1(D)$ is chosen as the solution of (4.3). The inverse problem can then be formulated as

$$\text{For any given } \hat{w}\in H_0^1(D),\ \text{find } y\in\mathbb{R}^M,\ \text{such that } \hat{G}(y)=\hat{w}. \tag{4.6}$$

In practical applications, it is not possible to directly observe $\hat{w}\in H_0^1(D)$. Hence, we assume that the measurement process of $\hat{w}$ is given by the bounded linear *observation* operator $\mathcal{O}\colon H_0^1(D)\to\mathbb{R}^J$ for some $J\in\mathbb{N}$. These observations are usually either obtained directly from sensors or after a postprocessing step. In our case, the observation operator describes the representation of a function in $H_0^1(D)$ by a finite element discretization with J degrees of freedom. When Courant FE are used, the degrees of freedom are equivalent to point observations in the domain related to the used mesh. In most applications, exact (deterministic) solutions to (4.6) do not exist or are not unique, which implies that the inverse problem is ill-posed. A remedy is to introduce some kind of regularization to (4.6). The most commonly chosen probabilistic approach introduces a random additive centered Gaussian measurement noise $\eta\sim\mathcal{N}(0,\Gamma)$ with covariance $\Gamma\in\mathbb{R}^{J\times J}$. With this, noisy observations are defined by

$$\delta=(\mathcal{O}\circ\hat{G})(y)+\eta=:G(y)+\eta. \tag{4.7}$$

Under some mild assumptions on G, one can show a continuous version of the Bayes formula. This yields the existence of a unique Radon–Nikodym derivative of the posterior measure π_δ of the conditional random variable $y|\delta$ with respect to the prior measure π_0 of y. We refer to [5, 57] and [58] for an

analysis in the context of parametric PDEs. Assuming the Gaussian noise η is independent of y this writes as

$$\frac{\mathrm{d}\pi_\delta}{\mathrm{d}\pi_0}(y) = Z^{-1}L(y;\delta)\,, \qquad Z := \mathbb{E}_{\pi_0}[L(y;\delta)], \tag{4.8}$$

where the likelihood $L(y;\delta) := \exp \ell(y;\delta)$ is given by the negative Bayesian potential (log-likelihood)

$$\ell(y;\delta) = -\frac{1}{2}\|\delta - G(y)\|_\Gamma^2 := -\frac{1}{2}(\delta - G(y)) \cdot \Gamma^{-1}(\delta - G(y)), \tag{4.9}$$

and Z is a normalization constant referred to as *evidence*.

A surrogate for the forward map $G(y)$ can be computed in the TT fromat as presented in [15, 27, 33, 48]. From this it is easy to derive a representation of the log-likelihood ℓ in TT format by simple algebraic operations. Our approach now provides the means to close the quite challenging, remaining gap to compute the TT representation of the likelihood L.

Initial condition for vector valued functions. The construction of the operator $\boldsymbol{B}$ and right-hand side $\boldsymbol{f}$ in Section 3 is done with real-valued functions in mind. However, this is not strictly required by our method. As an example, for some $y \in \mathbb{R}^M$ consider a discretization of the exponent of the log-normal diffusion coefficient $\gamma(y)$ in a finite element space of dimension J with the expansion

$$\gamma(x,y) \approx \sum_{j=1}^{J} \sum_{\mu \in [d_h]^M} \boldsymbol{\gamma}[j,\mu]\varphi_j(x)P_\mu(y) \qquad \text{for } x \in D \text{ and } y \in \mathbb{R}^M,$$

where $\{\varphi_j\}_{j=1}^J$ is a basis of the finite element space. The TT representation of $\boldsymbol{\gamma}$ then reads

$$\boldsymbol{\gamma}[j,\mu] = \sum_{k=1}^{r} \boldsymbol{\gamma}_0[j,k_1] \prod_{m=1}^{M} \boldsymbol{\gamma}_m[k_m,\mu_m,k_{m+1}].$$

In this setting it is not straight-forward to perform the construction of the operator $\boldsymbol{B}$ as described in (3.5)–(3.6). This is because the basis functions $\{\varphi_j\}_{j=1}^J$ for the deterministic mode depend on more than a single variable and have to be smooth enough for the operator (3.5) to be well-defined. As an alternative, we choose a set $\{x^{(j)}\}_{j=1}^J$ of interpolation points for the FE space and build the operator $\boldsymbol{B}$ and right-hand side $\boldsymbol{f}$ pointwise for each finite element node $x^{(j)}$. Here the interpolation points have to be chosen in such a way that a FE function can be recovered uniquely from its values at these points. Since we use Lagrange FEM we use the Lagrange points together with the standard interpolation. The resulting equations can be combined in a single system, which results in a slightly different operator $\boldsymbol{W}$ and right-hand side $\boldsymbol{b}$ but has no effect on the ranks.

4.1. **Approximation of the Darcy diffusion coefficient.** In this section we investigate the approximation of the log-normal diffusion coefficient κ of (4.3), which for instance can be used in a stochastic Galerkin scheme. For the experiments conducted in this section we choose to discretize the diffusion field $\kappa \in L^2(\mathbb{R}^M, \rho; L^\infty(D))$ with conforming first order Lagrange finite elements for varying degrees of freedom (DoF) and stochastic dimensions M. We observe that the choice of polynomial order for the spatial component has no influence on the approximation quality as a comparison with order zero discontinuous and higher order continuous Lagrange elements yields similar results. The exponent $\gamma = \log \kappa$ is approximated in the same finite element space as κ. Since γ is an affine function in the stochastic variables y (cf. by (4.4)), we set $d_h = 2$. To obtain an approximation of the exponent γ in TT format we employ the variational Monte Carlo (VMC) method [27]. This method recovers the tensor train representation $\boldsymbol{f}$ of a function f from a given set of samples $\{(y_i, f(y_i)\}_{i=1}^{N_{\mathrm{VMC}}}$ by minimizing the

least-squares loss

$$\underset{\boldsymbol{f}}{\text{minimize}} \sum_{i=1}^{N_{\mathrm{VMC}}} \|f(y_i) - \boldsymbol{f} \cdot \boldsymbol{P}(y_i)\|.$$

Here $\boldsymbol{f} \cdot \boldsymbol{P}(y_i)$ denotes the Frobenius-inner product of the tensors $\boldsymbol{f}$ and $\boldsymbol{P}(y_i)$ and $\boldsymbol{P}(y_i)$ is defined as $\boldsymbol{P}(y_i)[\mu] := P_\mu(y_i)$. Besides the TT cross algorithm [18, 49], the VMC method is another versatile way to obtain TT representations of functions in a non-intrusive sample based manner. And, in contrast to the TT cross algorithm, convergence bounds are known in probability (cf. [27, 59]). Here, it is in principle possible to find an exact representation of the affine exponent γ. Nevertheless, we choose an approximation via VMC for two reasons. First, the non-intrusive character of VMC allows for easy adaptation to other more complicated problems, which is why we expect this to be commonly done, even if it is feasible to obtain exact representations in specific cases. Second, there might not exist an exact TT representation for other applications or it might be very intricate to derive. The choice of an inexact representation of γ thus demonstrates the practical relevance of our method due to a broad applicability. Additionally, since Theorem 2.4 holds for any approximation of κ, this is a good opportunity to confirm our theoretical results.

The VMC method only requires evaluations of γ in realizations $\{y^{(i)}\}_{i=1}^{N_{\mathrm{VMC}}}$ to find a low-rank approximation of a function in TT format. We increase N_{VMC} as M gets larger to obtain approximations γ_{VMC} of γ with relative error $\varepsilon_\gamma(\gamma_{\mathrm{VMC}}) \leq 10^{-8}$ for all M depicted in Tables 1 and 2. The approximations of κ are computed for uniform polynomial degree $d_a = 10$ for each stochastic component via Algorithm 2 with scaling number $s = 32$. We choose continuous first-order Lagrange elements with various mesh sizes, Lagrange interpolation points $x_0^{(1)}, \dots, x_0^{(J)}$ and $y_0 = 0 \in \mathbb{R}^M$ as initial points. As the stopping tolerance for all experiments we set $\varepsilon = 10^{-8}$ and round the rescaling of the approximation to $\varepsilon_s = 10^{-7}$ in each iteration (cf. Algorithm 2).

M	$\varepsilon_\gamma(\gamma_{\mathrm{VMC}})$	$\varepsilon_\kappa^\infty(\kappa_{\mathrm{VMC}})$	$\mathrm{res}(\tilde{\kappa}_a)$	$\varepsilon_\kappa(\kappa_a)$	$\varepsilon_\kappa^\infty(\kappa_a)$	time [s]
5	$3.33 \cdot 10^{-9}$	$1.32 \cdot 10^{-3}$	$9.82 \cdot 10^{-5}$	$4.10 \cdot 10^{-5}$	$6.94 \cdot 10^{-5}$	110.05
10	$5.82 \cdot 10^{-9}$	$2.77 \cdot 10^{-2}$	$5.59 \cdot 10^{-5}$	$1.33 \cdot 10^{-5}$	$2.87 \cdot 10^{-5}$	201.43
15	$1.41 \cdot 10^{-9}$	$1.95 \cdot 10^{-2}$	$3.10 \cdot 10^{-5}$	$7.40 \cdot 10^{-6}$	$1.99 \cdot 10^{-5}$	590.92
20	$2.68 \cdot 10^{-9}$	$1.77 \cdot 10^{-2}$	$1.33 \cdot 10^{-5}$	$6.79 \cdot 10^{-6}$	$1.79 \cdot 10^{-5}$	1865.92

TABLE 1. Relative appoximation errors and computation time for the approximation of the log-normal diffusion coefficient κ for different numbers of stochastic parameters M. The computation is done on a uniform triangulation of D with 5000 triangles (2601 FE DoFs) and uses stochastic polynomials of degree 10 or less for each mode. Here, κ_{VMC} is an approximation of κ obtained via direct VMC and κ_a is the output of Algorithm 2. $\tilde{\kappa}_a$ is the scaled approximation computed in line 2 of Algorithm 2.

Table 1 shows errors of the approximations γ_{VMC} and κ_{VMC} obtained via the VMC method and of the output κ_a of Algorithm 2 for different expansion lengths M. Algorithm 1 converges in less than 10 iterations to the prescribed tolerance of $\varepsilon = 10^{-8}$.

When using the generic VMC approach to directly reconstruct an approximation κ_{VMC} of κ from samples with the same stochastic dimensions and $N_{\mathrm{VMC}} = 10^4$, the relative error of $\varepsilon_\kappa^\infty(\kappa_{\mathrm{VMC}})$ seems to stagnate independent of M at about 10^{-2}, which exceeds the error of our method by three orders of magnitude.

Even though the exponent γ does not satisfy the conditions of Corollary 2.6, the relative discrete residual $\mathrm{res}(\tilde{\kappa}_a)$ is of the same magnitude as $\varepsilon_{\tilde{\kappa}}(\tilde{\kappa}_a)$ and $\varepsilon_{\tilde{\kappa}}^\infty(\tilde{\kappa}_a)$ independent of the number of modes M and the degrees of freedom of the FE space in our experiments. Since we do not observe a significant change in the L^2 and L^∞ errors upon rescaling, we omit those values in Tables 1 and 2.

The error $\varepsilon_\kappa^\infty(\kappa_a)$ is comparable to the approximation results of [18] and about one order of magnitude smaller then the ones reported in [16, 17], which suggests that our method compares favourably to the state of the art algorithms. Table 2 shows errors and computation times for the reconstruction of κ for a fixed number of modes $M = 20$ and an increasing number of FE degrees of freedom. The dimension of the finite element space does not seem to have any influence on either the relative approximation errors $\varepsilon_\kappa(\kappa_a)$ and $\varepsilon_\kappa^\infty(\kappa_a)$ or the discrete residual $\mathrm{res}(\tilde{\kappa}_a)$.

DoF	$\varepsilon_\gamma(\gamma_{\mathrm{VMC}})$	$\varepsilon_\kappa^\infty(\kappa_{\mathrm{VMC}})$	$\mathrm{res}(\tilde{\kappa}_a)$	$\varepsilon_\kappa(\kappa_a)$	$\varepsilon_\kappa^\infty(\kappa_a)$	time [s]
441	$1.83 \cdot 10^{-9}$	$1.93 \cdot 10^{-2}$	$6.41 \cdot 10^{-6}$	$2.88 \cdot 10^{-6}$	$9.49 \cdot 10^{-6}$	1412.94
2601	$2.68 \cdot 10^{-9}$	$1.77 \cdot 10^{-2}$	$1.33 \cdot 10^{-5}$	$6.79 \cdot 10^{-6}$	$1.79 \cdot 10^{-5}$	1865.92
6561	$3.71 \cdot 10^{-10}$	$2.15 \cdot 10^{-2}$	$2.12 \cdot 10^{-5}$	$8.99 \cdot 10^{-6}$	$2.22 \cdot 10^{-5}$	4043.40
10201	$1.22 \cdot 10^{-9}$	$2.49 \cdot 10^{-2}$	$3.00 \cdot 10^{-5}$	$3.60 \cdot 10^{-5}$	$9.89 \cdot 10^{-5}$	12788.44

TABLE 2. Relative appoximation errors and computation time for the approximation of the log-normal diffusion coefficient κ for different numbers of FE degrees of freedom. The computation is done on uniform triangulations of D with $M = 20$ parameters and uses stochastic polynomials of degree smaller or equal than 10 in each mode. Here, κ_{VMC} is an approximation of κ obtained via direct VMC and κ_a is the output of Algorithm 2. $\tilde{\kappa}_a$ is the scaled approximation computed in line 2 of Algorithm 2.

The computation time of our algorithm increases drastically as the number of FE DoFs get larger. However, we suspect that this behavior originates from the discretization of the FE space. As discussed in Section 3.2, the ranks of the operator $\boldsymbol{W}$ and right-hand side $\boldsymbol{b}$ depend quadratically on the ranks of the exponent γ_{VMC}. In our case the ranks are bounded by around $r = 20$ for the first component tensor and the ranks decrease with the distance to the first component. We also observe that the maximal ranks of γ_{VMC} increase as M gets larger. To improve storage capacity of the ALS algorithm, we *round* $\boldsymbol{W}$ to a precision of 10^{-12} by applying a truncated SVD to each component of the TT operator, which significantly reduces the ranks. However, this process is computationally expensive as the finite element component of $\boldsymbol{W}$ consists of a high-dimensional tensor whose sparsity is lost upon rounding. This in turn increases storage capacity and computation time of the truncated SVD. A different choice of spatial discretization by e.g. a reduced basis approach [60] could decrease the dimension of the deterministic approximation space and thus possibly reduce the computation time significantly. That the computation time increases with the cardinality of the spatial discretization can also be observed in the next section on log-likelihood reconstruction. A verification of this and possible improvements are subject to future work.

4.2. **Approximation of posterior densities.** In the following experiment we examine the approximation quality of our approach for the Bayesian likelihood (4.9). The forward map $\hat{G}(y) = w(y) \in H_0^1(D)$ is determined by the solution of the stationary diffusion problem (4.3) with log-normal random permeability $\kappa \in L^2(\mathbb{R}^M, \rho; L^\infty(D))$ which is specified by the affine exponent (4.4)–(4.5). The parameter to observation map G is the FE solution of $\hat{G}$ discretized with a lowest order conforming Lagrange finite element method with $J = 2601$ degrees of freedom in the physical space and a maximal polynomial chaos degree of 2 for all stochastic modes. The observation

$$\delta = G(y^*) + \eta$$

is a perturbed realization of G for some random sample $y^* \sim \mathcal{N}(0, I)$ where the perturbation noise η is chosen with covariance $\sigma^2 I$ for $\sigma = 10^{-3}$.

The absolute approximation errors for different quantities are depicted in Table 3. Relative errors are shown in Table 4. The VMC approximation of the forward map is denoted by G_{VMC} and with ℓ_{VMC} we denote the TT representation of the log-likelihood that is computed algebraically from G_{VMC}. The

M	$\mathcal{E}_G(G_{\mathrm{VMC}})$	$\mathcal{E}_\ell(\ell_{\mathrm{VMC}})$	$\mathrm{res}(L_a)$	$\mathcal{E}_L(L_a)$	$\mathcal{E}_{\hat{L}}(L_a)$
5	$2.80 \cdot 10^{-4}$	$3.24 \cdot 10^{-4}$	$2.63 \cdot 10^{-5}$	$2.57 \cdot 10^{-4}$	$2.79 \cdot 10^{-6}$
10	$2.82 \cdot 10^{-4}$	$4.30 \cdot 10^{-4}$	$8.79 \cdot 10^{-6}$	$2.69 \cdot 10^{-4}$	$6.75 \cdot 10^{-6}$
20	$3.04 \cdot 10^{-4}$	$4.83 \cdot 10^{-4}$	$1.12 \cdot 10^{-5}$	$3.46 \cdot 10^{-4}$	$6.31 \cdot 10^{-6}$
30	$3.40 \cdot 10^{-4}$	$3.53 \cdot 10^{-4}$	$8.66 \cdot 10^{-6}$	$3.34 \cdot 10^{-4}$	$8.57 \cdot 10^{-7}$
40	$3.49 \cdot 10^{-4}$	$3.30 \cdot 10^{-4}$	$1.62 \cdot 10^{-5}$	$8.04 \cdot 10^{-4}$	$3.34 \cdot 10^{-5}$

TABLE 3. Absolute appoximation errors for the approximation of the forward model G, the log-likelihood ℓ and the likelihood L for different expansion dimensions M. The forward model is discretized on a uniform triangulation with 5000 simplices. Here, $\hat{L} = \exp \ell_{\mathrm{VMC}}$ is used as a reference for the error of our method.

likelihood approximations of our method is labeled by L_a where the stochastic discretization space for each mode is restricted to polynomials of maximal degree 3. As stopping tolerance for our method, we set $\varepsilon = 10^{-8}$. Due to the relatively small function values of the log-likelihood, it suffices to set the scaling to $s = 1$. This implies $\tilde{d}_a = d_a = 4$ and renders the choice of ε_s irrelevant. As initial point for the method we choose $y_0 = 0 \in \mathbb{R}^M$. To determine if the approximation accuracy is limited by our method or by the reconstruction of the forward model G, we additionally compute the error between the likelihood approximation L_a and samples $\hat{L}(y) = \exp \ell_{\mathrm{VMC}}(y)$ for $y \sim \mathcal{N}(0, I)$. Finally, we compare the approximation obtained by our method to a merely sample based VMC tensor reconstruction L_{VMC} where the stochastic discretization space for each mode is restricted to polynomials of maximal degree 3 as well. We observe that $N_{\mathrm{VMC}} = 10^3$ samples seem sufficient for the reconstructions and an increase of N_{VMC} yields no significant improvements.

M	$\varepsilon_G(G_{\mathrm{VMC}})$	$\varepsilon_\ell(\ell_{\mathrm{VMC}})$	$\varepsilon_L(L_{\mathrm{VMC}})$	$\varepsilon_L(L_a)$	$\varepsilon_{\hat{L}}(L_a)$	time [s]
5	$1.86 \cdot 10^{-2}$	85.7	$1.19 \cdot 10^{-4}$	$2.57 \cdot 10^{-4}$	$2.79 \cdot 10^{-6}$	0.09
10	$1.87 \cdot 10^{-2}$	86.3	$1.13 \cdot 10^{-4}$	$2.69 \cdot 10^{-4}$	$6.75 \cdot 10^{-6}$	0.78
20	$2.02 \cdot 10^{-2}$	84.7	$9.66 \cdot 10^{-5}$	$3.46 \cdot 10^{-4}$	$6.31 \cdot 10^{-6}$	0.29
30	$2.26 \cdot 10^{-2}$	87.0	$9.36 \cdot 10^{-5}$	$3.34 \cdot 10^{-4}$	$8.57 \cdot 10^{-7}$	0.77
40	$2.32 \cdot 10^{-2}$	87.2	$7.57 \cdot 10^{-5}$	$8.04 \cdot 10^{-4}$	$3.34 \cdot 10^{-5}$	1.67

TABLE 4. Relative approximation errors for the approximation of the forward model G, the log-likelihood ℓ and the likelihood L for different expansion dimensions M. The forward model is discretized on a uniform triangulation with 5000 simplices. Here, $\hat{L} = \exp \ell_{\mathrm{VMC}}$ is used as a reference for the error of our method. The last column is the measured time our algorithm requires to compute the likelihood L_a.

In Table 3 it can be seen that the approximation of the forward model G and the approximation of the log-likelihood ℓ seem to stagnate at an error of 10^{-4} independent of the number of modes. Note that the error of the latter directly depends on the error of the former. The absolute approximation quality of L_a has the same order of magnitude as the one of G_{VMC}, which is expected. However, the residual $\mathrm{res}(L_a)$ of the problem and thus the error in the energy norm is smaller and suggests that a significantly better approximation is obtained. This is verified by the last column of Table 3 where it can be seen that the main contribution of the approximation error originates from the error of the approximation of the forward map G. Here we assume the log-likelihood ℓ_{VMC} to be exact and compute the error with respect to $\hat{L} = \exp \ell_{\mathrm{VMC}}$ instead of $L = \exp \ell$.

It can be seen that the error of G_{VMC} in Table 4 is of the same order of magnitude as that of κ_{VMC} in Table 1. This indicates that VMC is not capable of recovering the exponential of a function with sufficient accuracy. This entails a large approximation error for ℓ_{VMC}. In conjunction with the fact that $\mathbb{E}[|\ell|] \ll 1$ this explains the large relative error (> 80) of the log-likelihood approximation. The relative

error of the sample based tensor regression L_{VMC} is only slightly smaller than the approximation error of our method. However, it is worth mentioning that the experiments suggest that our method only takes about one tenth of the computation time. The last column of Table 4 shows the measured time of our algorithm to compute the approximation L_a from ℓ_{VMC}. Similar to the approximation errors, the runtime seems to be rather independent from the number of expansion dimensions M or increases only slightly as M increases. This is a very different behaviour when compared to the results of Section 4.1. However, as mentioned before, the computation time seems to be correlated to (due to our "naive" implementation) the dimension of the deterministic component, which explains the short running times of our algorithm for the approximation of the real-valued likelihood L.

5. Conclusion

We derive a novel numerical approach to compute a low-rank approximation of the exponential of a multivariate function. We assume that the exponent is given with respect to an orthonormal basis of (tensor product Hermite) polynomials and that the coefficient tensor of the expansion is in the tensor train (TT) format. It is the central notion to consider the exponential as the solution of a system of ordinary differential equations. This allows to approximate the function via a Galerkin projection method. The Laplace-like structure of the resulting operator and right-hand side make an efficient representation in the TT format possible, which renders the problem amenable to the ALS. We establish that the residual minimized by the ALS is equivalent to a certain energy norm up to a data oscillation term. This not only implies that the ALS minimizes the distance to the exact solution in the energy norm but also that the resulting residual provides an error estimator for the solution, which in principle could be used for an adaptive refinement algorithm.

The algorithm is tested for the reconstruction of the log-normal diffusion coefficient of a random elliptic PDE and a Bayesian likelihood, where the forward map is given by a polynomial chaos surrogate in TT format. We compare our results to established methods and to a black-box sample based reconstruction algorithm. We observe that the performance of our approach is state-of-the-art with respect to the approximation accuracy, computation time and storage capacity for up to $M = 40$ stochastic dimensions and large polynomial degrees up to $p = 10$. Almost all computations are carried out on a common desktop computer[2] with the exception of the experiments with large FE dimensions (last two rows in Table 2), for which slightly more memory was required to assemble the operator $\boldsymbol{W}$. However, it should be noted that the computation of an approximation of the log-normal diffusion field is quite expensive computationally. In particular, our choice of spatial discretization in conjunction with our unoptimized implementation leads to long computations as the spatial dimension or the number of stochastic modes increase. Nevertheless, we are confident that this can be alleviated by a more appropriate choice of discretization of the physical space and an optimized implementation for the assembly of the operator, which will be subject to future work.

It should be emphasized that in principle the scope of our method reaches far beyond what is discussed and illustrated in this work. The proposed method is applicable to a wide range of holonomic-like functions such as algebraic functions, sine and cosine, the error function, Bessel functions and hypergeometric functions. Moreover, since the sought function may satisfy multiple differential equations it is possible to choose one for which the induced energy norm is best suited for the problem at hand. This may also be obtained by considering other orthogonal bases, e.g. from the Askey scheme. The dependence of the energy norm on the dynamical system clearly highlights a limitation of our current theory and it would be interesting to investigate for which classes of dynamical systems an equivalence of the energy norm to a more convenient norm like the L^2 or H^1 norm can be established.

[2] $2.1\,\mathrm{GHz}$ Intel Core i3 processor and $16\,\mathrm{GB}$ of memory.

The exponential field approximation discussed here could be used to develop fully adaptive approximation schemes for the solutions of parametric PDEs similar to [15] with the crucial advantage that only pointwise evaluations of the solution of the considered PDE are required. In contrast to the involved intrusive stochastic Galerkin methods of [15, 33], a black-box adaptive non-intrusive method could be devised which still yields the Galerkin solution with high probability.

On a more practical side, it is possible to apply our algorithm to obtain a functional representation of a Bayesian posterior density. This allows, among other things, a very efficient computation of statistical quantities such as mean, variance, higher order moments and marginals (cf. [11]) or fast generation of independent posterior samples (cf. [10]). This is important in many reconstruction tasks such as [7, 24].

REFERENCES

[1] O. Le Maitre and O. M. Knio. *Spectral Methods for Uncertainty Quantification: With Applications to Computational Fluid Dynamics (Scientific Computation)*. English. Springer, 04/2010, p. 552.

[2] R. G. Ghanem and P. Spanos. *Stochastic Finite Elements: A Spectral Approach*. English. Springer, 12/1990, p. 214.

[3] M. Eigel, C. J. Gittelson, C. Schwab, and E. Zander. “Adaptive stochastic Galerkin FEM”. In: *Computer Methods in Applied Mechanics and Engineering* 270 (03/2014), pp. 247–269.

[4] M. Eigel, C. Gittelson, C. Schwab, and E. Zander. *A convergent adaptive stochastic Galerkin finite element method with quasi-optimal spatial meshes*. Tech. rep. 2014-01. Switzerland: Seminar for Applied Mathematics, ETH Zürich, 2014.

[5] A. M. Stuart. “Inverse problems: A Bayesian perspective”. In: *Acta Numerica* 19 (2010), pp. 451–559.

[6] J. Kaipio and E. Somersalo. *Statistical and computational inverse problems*. Vol. 160. Springer Science & Business Media, 2006.

[7] N. Farchmin, M. Hammerschmidt, P.-I. Schneider, M. Wurm, B. Bodermann, M. Bär, and S. Heidenreich. “Efficient Bayesian inversion for shape reconstruction of lithography masks”. In: *Journal of Micro/Nanolithography, MEMS, and MOEMS* 19.02 (05/2020), p. 1.

[8] S. Mo, N. Zabaras, X. Shi, and J. Wu. “Deep autoregressive neural networks for high-dimensional inverse problems in groundwater contaminant source identification”. In: *Water Resources Research* 55.5 (2019), pp. 3856–3881.

[9] T. Lassila, A. Manzoni, A. Quarteroni, and G. Rozza. “A reduced computational and geometrical framework for inverse problems in hemodynamics”. In: *International journal for numerical methods in biomedical engineering* 29.7 (2013), pp. 741–776.

[10] S. Dolgov, K. Anaya-Izquierdo, C. Fox, and R. Scheichl. “Approximation and sampling of multivariate probability distributions in the tensor train decomposition”. In: *Statistics and Computing* 30.3 (11/2019), pp. 603–625.

[11] M. Eigel, R. Gruhlke, and M. Marschall. *Low-rank tensor reconstruction of concentrated densities with application to Bayesian inversion*. 2020. arXiv: 2008.04264 [math.NA].

[12] A. Mugler. “Verallgemeinertes polynomielles Chaos zur Lösung stationärer Diffusionsprobleme mit zufälligen Koeffizienten”. doctoralthesis. BTU Cottbus - Senftenberg, 2013.

[13] L. Grasedyck, D. Kressner, and C. Tobler. “A literature survey of low-rank tensor approximation techniques”. In: *GAMM-Mitteilungen* 36.1 (08/2013), pp. 53–78.

[14] I. V. Oseledets and E. E. Tyrtyshnikov. “Breaking the Curse of Dimensionality, Or How to Use SVD in Many Dimensions”. In: *SIAM Journal on Scientific Computing* 31.5 (01/2009), pp. 3744–3759.

[15] M. Eigel, M. Marschall, M. Pfeffer, and R. Schneider. “Adaptive stochastic Galerkin FEM for lognormal coefficients in hierarchical tensor representations”.
In: *Numerische Mathematik* 145.3 (06/2020), pp. 655–692.

[16] M. Espig, W. Hackbusch, A. Litvinenko, H. G. Matthies, and P. Wähnert.
“Efficient low-rank approximation of the stochastic Galerkin matrix in tensor formats”.
In: *Computers & Mathematics with Applications* 67.4 (2014), pp. 818–829.

[17] S. Dolgov, B. N. Khoromskij, A. Litvinenko, and H. G. Matthies.
“Polynomial Chaos Expansion of Random Coefficients and the Solution of Stochastic Partial Differential Equations in the Tensor Train Format”.
In: *SIAM/ASA Journal on Uncertainty Quantification* 3.1 (01/2015), pp. 1109–1135.

[18] S. Dolgov and R. Scheichl.
“A Hybrid Alternating Least Squares–TT-Cross Algorithm for Parametric PDEs”.
In: *SIAM/ASA Journal on Uncertainty Quantification* 7.1 (01/2019), pp. 260–291.

[19] I. Oseledets. “DMRG Approach to Fast Linear Algebra in the TT-Format”. en.
In: *Computational Methods in Applied Mathematics* 11.3 (2011).

[20] S. Holtz, T. Rohwedder, and R. Schneider.
“The Alternating Linear Scheme for Tensor Optimization in the Tensor Train Format”. en.
In: *SIAM Journal on Scientific Computing* 34.2 (01/2012), A683–A713.

[21] S. Holtz, T. Rohwedder, and R. Schneider. “On manifolds of tensors of fixed TT-rank”.
In: *Numerische Mathematik* 120.4 (09/2011), pp. 701–731.

[22] Y. Marzouk, T. Moselhy, M. Parno, and A. Spantini.
“Sampling via Measure Transport: An Introduction”. In: *Handbook of Uncertainty Quantification*. Springer International Publishing, 2016, pp. 1–41.

[23] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan.
“Normalizing Flows for Probabilistic Modeling and Inference”.
In: *Journal of Machine Learning Research* 22.57 (2021), pp. 1–64.

[24] A. Andrle, N. Farchmin, P. Hagemann, S. Heidenreich, V. Soltwisch, and G. Steidl.
Invertible Neural Networks versus MCMC for Posterior Reconstruction in Grazing Incidence X-Ray Fluorescence. 2021. arXiv: 2102.03189 [cs.LG].

[25] P. B. Rohrbach, S. Dolgov, L. Grasedyck, and R. Scheichl.
“Rank bounds for approximating gaussian densities in the tensor-train format”.
In: *arXiv preprint arXiv:2001.08187* (2020).

[26] P. Hagemann and S. Neumayer. *Stabilizing Invertible Neural Networks Using Mixture Models*. 2020. arXiv: 2009.02994 [cs.LG].

[27] M. Eigel, R. Schneider, P. Trunschke, and S. Wolf. “Variational Monte Carlo—bridging concepts of machine learning and high-dimensional partial differential equations”.
In: *Advances in Computational Mathematics* 45.5-6 (10/2019), pp. 2503–2532.

[28] J. Ballani and L. Grasedyck. “Tree adaptive approximation in the hierarchical tensor format”.
In: *SIAM journal on scientific computing* 36.4 (2014), A1415–A1431.

[29] I. Oseledets and E. Tyrtyshnikov. “TT-cross approximation for multidimensional arrays”.
In: *Linear Algebra and its Applications* 432.1 (2010), pp. 70–88.

[30] W. Hackbusch. *Tensor Spaces and Numerical Tensor Calculus (Springer Series in Computational Mathematics Book 42)*. English. Springer, 02/2012, p. 524.

[31] B. N. Khoromskij. “Tensor numerical methods for multidimensional PDES: theoretical analysis and initial applications”. In: *ESAIM: Proceedings and Surveys* 48 (01/2015). Ed. by N. Champagnat, T. Lelièvre, and A. Nouy, pp. 1–28.

[32] T. G. Kolda and B. W. Bader. “Tensor Decompositions and Applications”.
In: *SIAM Review* 51.3 (08/2009), pp. 455–500.

[33] M. Eigel, M. Pfeffer, and R. Schneider.
“Adaptive stochastic Galerkin FEM with hierarchical tensor representations”.
In: *Numerische Mathematik* 136.3 (11/2016), pp. 765–803.

[34] M. Bachmayr, R. Schneider, and A. Uschmajew. “Tensor Networks and Hierarchical Tensors for the Solution of High-Dimensional Partial Differential Equations”.
In: *Foundations of Computational Mathematics* 16.6 (04/2016), pp. 1423–1472.

[35] L. Grasedyck. “Hierarchical Singular Value Decomposition of Tensors”.
In: *SIAM Journal on Matrix Analysis and Applications* 31.4 (01/2010), pp. 2029–2054.

[36] M. Loeve. *Probability Theory II (Graduate Texts in Mathematics, 46)*. Springer, 05/1978, p. 432.

[37] W. Hackbusch and R. Schneider. “Tensor Spaces and Hierarchical Tensor Representations”.
In: *Extraction of Quantifiable Information from Complex Systems*.
Springer International Publishing, 2014, pp. 237–261.

[38] V. A. Kazeev and B. N. Khoromskij.
“Low-Rank Explicit QTT Representation of the Laplace Operator and Its Inverse”.
In: *SIAM Journal on Matrix Analysis and Applications* 33.3 (2012), pp. 742–758.
eprint: https://doi.org/10.1137/100820479.

[39] J. Pousin and J. Rappaz. “Consistency, stability, a priori and a posteriori errors for Petrov-Galerkin methods applied to nonlinear problems”.
In: *Numerische Mathematik* 69.2 (12/1994), pp. 213–231.

[40] C. Carstensen, L. Demkowicz, and J. Gopalakrishnan.
“A Posteriori Error Control for DPG Methods”.
In: *SIAM Journal on Numerical Analysis* 52.3 (01/2014), pp. 1335–1353.

[41] D. Braess. *Finite elements*. Third. Theory, fast solvers, and applications in elasticity theory, Translated from the German by Larry L. Schumaker.
Cambridge University Press, Cambridge, 2007, pp. xviii+365.

[42] J. Galvis and M. Sarkis.
“Approximating infinity-dimensional stochastic Darcy’s equations without uniform ellipticity”.
In: *SIAM J. Numer. Anal.* 47.5 (2009), pp. 3624–3651.

[43] A. Mugler and H.-J. Starkloff. “On the convergence of the stochastic Galerkin method for random elliptic partial differential equations”.
In: *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique* 47.5 (2013), pp. 1237–1263.

[44] J. Charrier, R. Scheichl, and A. L. Teckentrup. “Finite element error analysis of elliptic PDEs with random coefficients and its application to multilevel Monte Carlo methods”.
In: *SIAM Journal on Numerical Analysis* 51.1 (2013), pp. 322–352.

[45] M. Eigel, C. Merdon, and J. Neumann. “An Adaptive Multilevel Monte Carlo Method with Stochastic Bounds for Quantities of Interest with Uncertain Data”.
In: *SIAM/ASA Journal on Uncertainty Quantification* 4.1 (01/2016), pp. 1219–1245.

[46] I. Babuška, F. Nobile, and R. Tempone. “A stochastic collocation method for elliptic partial differential equations with random input data”. In: *SIAM review* 52.2 (2010), pp. 317–355.

[47] F. Nobile, L. Tamellini, F. Tesei, and R. Tempone.
“An adaptive sparse grid algorithm for elliptic PDEs with lognormal diffusion coefficient”.
In: *Sparse Grids and Applications-Stuttgart 2014*. Springer, 2016, pp. 191–220.

[48] M. Eigel, M. Marschall, and R. Schneider.
“Sampling-free Bayesian inversion with adaptive hierarchical tensor representations”.
In: *Inverse Problems* 34.3 (02/2018), p. 035010.

[49] S. Dolgov and D. Savostyanov.
“Parallel cross interpolation for high-precision calculation of high-dimensional integrals”.
In: *Computer Physics Communications* 246 (01/2020), p. 106869.

[50] E. Ullmann. “Solution strategies for stochastic finite element discretizations”. PhD thesis. TU Bergakademie Freiberg, 2008.

[51] P. Malliavin. *Stochastic Analysis (Grundlehren der mathematischen Wissenschaften, 313)*. English. Springer, 04/1997, p. 359.

[52] M. Alnæs, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M. E. Rognes, and G. N. Wells. “The FEniCS Project Version 1.5”. en. In: *Archive of Numerical Software* Vol 3 (2015), pp. 9–23.

[53] B. Huber and S. Wolf. *Xerus - A General Purpose Tensor Library*. `https://libxerus.org/`. 2014–2021.

[54] V. H. Hoang and C. Schwab. “N-term Wiener chaos approximation rates for elliptic PDEs with lognormal gaussian random inputs”. In: *Mathematical Models and Methods in Applied Sciences* 24.04 (01/2014), pp. 797–826.

[55] C. J. Gittelson. “Stochastic Galerkin Discretization of the Log-Normal isotropic Diffusion Problem”. In: *Mathematical Models and Methods in Applied Sciences* 20.02 (02/2010), pp. 237–263.

[56] C. Schwab and C. J. Gittelson. “Sparse tensor discretizations of high-dimensional parametric and stochastic PDEs”. In: *Acta Numerica* 20 (2011), pp. 291–467.

[57] M. Dashti and A. M. Stuart. “The Bayesian Approach to Inverse Problems”. In: *Handbook of Uncertainty Quantification*. Springer International Publishing, 2017, pp. 311–428.

[58] C. Schwab and A. M. Stuart. “Sparse deterministic approximation of Bayesian inverse problems”. In: *Inverse Problems* 28.4 (2012), p. 045003.

[59] M. Eigel, P. Trunschke, and R. Schneider. “Convergence bounds for empirical nonlinear least-squares”. In: (2020).

[60] P. Chen, A. Quarteroni, and G. Rozza. “A Weighted Reduced Basis Method for Elliptic Partial Differential Equations with Random Input Data”. In: *SIAM Journal on Numerical Analysis* 51.6 (01/2013), pp. 3163–3185.