A Finite Element Method for Elliptic Problems with Stochastic Input Data

Helmut Harbrecht

no. 408
A FINITE ELEMENT METHOD FOR ELLIPTIC PROBLEMS WITH STOCHASTIC INPUT DATA

HELMUT HARPRECHT

Abstract. We compute the expectation and the two-point correlation of the solution to elliptic boundary value problems with stochastic input data. Besides stochastic loadings, via perturbation theory, our approach covers also elliptic problems on stochastic domains or with stochastic coefficients. The solution’s two-point correlation satisfies a hypo-elliptic boundary value problem on the tensor product domain. For its numerical solution we apply a sparse tensor product approximation by multilevel frames. This way standard finite element techniques can be used. Numerical examples illustrate feasibility and scope of the method.

1. Introduction

In recent years it became more and more important to model and simulate boundary value problems with stochastic input parameters. If a statistical description of the input data is available, one can mathematically describe data and solutions as random fields and aim at the computation of corresponding deterministic statistics of the unknown random solution. Applications are, besides traditional engineering, for example biomedical or biomechanical processes. To simulate biomechanical processes one has, on the one hand, uncertain domains arising from e.g. tomographic data. On the other hand, one often has only estimates on the material parameters.

In the present paper we consider elliptic boundary value problems with stochastic load. Nevertheless, assuming small stochastic perturbations around a nominal input parameter, we can linearize in order to approximate the solution’s nonlinear dependence on this parameter. That way, we are also able to treat boundary value problems on stochastic domains or with stochastic diffusion matrix.

From statistical information of the random input data we like to compute statistics of engineering interest for the random solution of the boundary value problem. In particular, we like to compute the expectation

\[ E_u(x) = \int_\Omega u(x, \omega) dP(\omega), \quad x \in D, \]

and the two-point correlation

\[ \text{Cor}_u(x, y) = \int_\Omega u(x, \omega) u(y, \omega) dP(\omega), \quad x, y \in D. \]
From these quantities the variance is derived by \( \text{Var}_u(x) = \text{Cor}_u(x, x) - E_u^2(x) \). Our goal of computation is thus as follows: for given mean and two-point correlation of the stochastic input parameter, compute, to leading order, the mean and the two-point correlation of the random solution of the boundary value problem.

From [29, 30] it is known that the \( p \)-th stochastic moment satisfies a hypo-elliptic boundary value problem on the \( p \)-fold tensor product domain \( D \times D \times \cdots \times D \). Therefore, the two-point correlation is given by a partial differential equation on the tensor product domain \( D \times D \). We employ a sparse tensor product approximation to solve this high dimensional problem efficiently. Sparse tensor product spaces have been successfully applied in the deterministic computation of the solution’s statistics for elliptic partial differential equations with stochastic data, see for example [20, 26, 29, 30]. However, in the present paper we are going to use a multilevel frame as proposed in [21] instead of wavelet or multilevel bases to approximate the functions from the sparse tensor product space. Thus we can use standard multigrid hierarchies and traditional finite elements. The frame construction is based on the BPX-preconditioner (see e.g. [5, 11, 17, 18, 25]) and related generating systems (see e.g. [15, 16]).

The starting point of the frame construction is a multilevel hierarchy of nested finite element spaces

\[
V_0 \subset V_1 \subset V_2 \subset \cdots \subset H^1(D).
\]

On polygonal domains such a sequence is obtained by the standard procedure of uniformly refining a coarse level triangulation. In case of curved domains we can employ parametric finite elements to realize our goal. The construction of parametric finite elements is based on a decomposition of the given domain into (curved) simplicial patches and suitable diffeomorphisms between the reference simplex and these patches. Uniform refinement of the reference simplex will lead then to a mesh of the domain. On each patch finite elements are defined via parameterization, gluing across the patch boundaries. The push-forward and pull-back operators, required in finite element based methods, can be derived from the underlying diffeomorphisms.

Up to now we did not tackle inhomogeneous Dirichlet data in the multilevel frame approach from [21]. We present here an construction to extent given Dirichlet data into the domain. Our method retains the structural and computational advantages of multilevel frames while preserving the efficiency. Thus, we obtain an algorithm which computes the two-point correlation with a complexity that stays essentially proportional to the number of unknowns \( N \) required for discretizing the domain \( D \). Here and throughout the paper, “essentially” means up to powers of \( \log N \) resp., in the context of convergence rates, up to powers of \( |\log h| \). Our algorithm involves
only prolongations, restrictions, and finite element mass and stiffness matrices. These ingredients are provided by standard finite element tools.

The outline of the paper is as follows. Section 2 is concerned with elliptic boundary value problems with stochastic input data. Particularly we show how to treat stochastic domains or differential operators with stochastic coefficients. In Section 3 we introduce the finite element spaces and compute the solution’s expectation. Section 4 is dedicated to the efficient second moment computation by multilevel frames. Finally, in Section 5 we present numerical results.

Throughout the paper, in order to avoid the repeated use of generic but unspecified constants, by $C \lesssim D$ we mean that $C$ can be bounded by a multiple of $D$, independently of parameters which $C$ and $D$ may depend on. Obviously, $C \gtrsim D$ is defined as $D \lesssim C$, and $C \sim D$ as $C \lesssim D$ and $C \gtrsim D$.

2. Boundary value problems with uncertain parameters

2.1. Boundary value problems with stochastic load. Let $(\Omega, \Sigma, P)$ be a probability space and $D \subset \mathbb{R}^n$ be a domain with Lipschitz boundary $\partial D$. Let $A \in L^\infty(D, \mathbb{R}^{n \times n})$ be symmetric and positive definite, that is

$$\alpha \|\xi\|_2^2 \leq \xi^T A(x) \xi \leq \beta \|\xi\|_2^2, \quad x \in D, \quad \xi \in \mathbb{R}^n.$$

We first focus on the following Dirichlet problem with stochastic load

$$- \text{div} \left[ A(x) \nabla u(x, \omega) \right] = f(x, \omega), \quad x \in D, \quad \omega \in \Omega,$$

$$u(x, \omega) = g(x, \omega), \quad x \in \partial D, \quad \omega \in \Omega.$$

We shall assume that the statistics of $f$ and $g$ is known. This means, the expectations of $f$ and $g$, that is

$$E_f(x) = \int_\Omega f(x, \omega) dP(\omega), \quad x \in D,$$

$$E_g(x) = \int_\Omega g(x, \omega) dP(\omega), \quad y \in \partial D,$$

as well as their two-point correlation, that is

$$\text{Cor}_f(x, y) = \int_\Omega f(x, \omega) f(y, \omega) dP(\omega), \quad x, y \in D,$$

$$\text{Cor}_g(x, y) = \int_\Omega g(x, \omega) g(y, \omega) dP(\omega), \quad x, y \in \partial D,$$

are given. If $f$ and $g$ are correlated we need in addition this correlation

$$\text{Cor}_{fg}(x, y) = \int_\Omega f(x, \omega) g(y, \omega) dP(\omega), \quad x \in D, \quad y \in \partial D.$$
Notice that we have
\[ \operatorname{Cor}_{g,f}(x, y) = \int_{\Omega} g(x, \omega) f(y, \omega) dP(\omega) = \operatorname{Cor}_{f,g}(y, x), \quad x \in \partial D, \; y \in D. \]

It has been shown in [29] that the statistics of \( u \) is determined by deterministic equations. Namely, the expectation \( \mathbb{E}_u \in H^1(D) \) of \( u \) satisfies
\[ -\nabla [A(x) \nabla \mathbb{E}_u(x)] = E_f(x), \quad x \in D, \\ \mathbb{E}_u(x) = E_y(x), \quad x \in \partial D, \tag{2.2} \]
while the two-point correlation \( \operatorname{Cor}_u \in H^{1,1}(D \times D) := H^1(D) \otimes H^1(D) \) is given by the following tensor product boundary value problem
\[ (\nabla_x \otimes \nabla_y) [(A(x) \otimes A(y))(\nabla_x \otimes \nabla_y) \operatorname{Cor}_u(x, y)] = \operatorname{Cor}_{f,y}(x, y), \quad x, y \in D, \\ -\nabla_x [A(x) \nabla_x \operatorname{Cor}_u(x, y)] = \operatorname{Cor}_{f,g}(x, y), \quad x \in D, \; y \in \partial D, \\ -\nabla_y [A(y) \nabla_y \operatorname{Cor}_u(x, y)] = \operatorname{Cor}_{g,f}(y, x), \quad x \in \partial D, \; y \in D, \quad x, y \in \partial D. \tag{2.3} \]

We mention that the variance \( \operatorname{Var}_u \in H^1(D) \) of \( u \) can then be derived from
\[ \operatorname{Var}_u(x) = \mathbb{E}_{u^2}(x) - \mathbb{E}_u^2(x) = \operatorname{Cor}_u(x, x) - \mathbb{E}_u^2(x), \quad x \in D. \]

2.2. PDEs on stochastic domains. We will now consider the domain as the uncertain input parameter of the boundary value problem, i.e.,
\[ -\nabla [A(x) \nabla u(x, \omega)] = f(x), \quad x \in D(\omega), \quad \omega \in \Omega, \\ u(x, \omega) = g(x), \quad x \in \partial D(\omega), \quad \omega \in \Omega. \tag{2.4} \]
Herein, the stochastic domain is modeled as follows. Let \( \overline{D} \) denote the sufficiently smooth nominal domain and consider stochastic boundary variations, for example in direction of the outward normal
\[ U(x, \omega) = \varepsilon \kappa(x, \omega) n(x) : \partial \overline{D} \to \mathbb{R}^n, \]
with \( \|\kappa(\cdot, \omega)\|_{C^{3,\alpha}(\partial \overline{D})} \leq 1 \) almost surely. Then, the stochastic domain is described via the method of mappings
\[ \partial D(\omega) = \{x + \varepsilon \kappa(x, \omega) n(x) : x \in \partial D\}. \]
For small parameters \( \varepsilon > 0 \) one can linearize the problem (2.4). By using first and second order local shape derivatives we derive the second order shape Taylor expansion
\[ u(x, \omega) = \overline{u}(x) + \varepsilon du(x)[\kappa(\omega)] + \frac{\varepsilon^2}{2} d^2 u(x)[\kappa(\omega), \kappa(\omega)] + O(\varepsilon^3), \quad x \in K \subseteq \overline{D}, \quad \text{P-a.e.} \; \omega \in \Omega, \tag{2.5} \]
A FE METHOD FOR ELLIPTIC PROBLEMS WITH STOCHASTIC INPUT DATA

see [13, 14, 22, 28]. Herein, \( \overline{\pi} \in H^1(\overline{D}) \) is defined by

\[
- \text{div} \left[ A(x) \nabla \overline{\pi}(x) \right] = f(x), \quad x \in \overline{D},
\]

\[
\overline{\pi}(x) = g(x), \quad x \in \partial \overline{D},
\]

and the local shape derivative \( du = du[\kappa(\omega)] \in H^1(\overline{D}) \) reads as

\[
- \text{div} \left[ A(x) \nabla du(x, \omega) \right] = 0, \quad x \in \overline{D},
\]

\[
du(x, \omega) = \kappa(x, \omega) \frac{\partial \overline{\pi}}{\partial n}(x), \quad x \in \partial \overline{D}.
\]

With the help of (2.5), the stochastic domain is transformed to a stochastic right hand side since the random boundary perturbation appears only in the Dirichlet data of the local shape derivatives.

Without loss of generality we assume that \( E_{\kappa} = 0 \) (otherwise we redefine \( \overline{D} \) correspondingly) and arrive in view of (2.5) at (cf. [20])

\[
E_u(x) = \overline{\pi}(x) + O(\varepsilon^2), \quad \text{Cov}_{u}(x,y) = \varepsilon^2 \text{Cor}_{du}(x,y) + O(\varepsilon^3), \quad x, y \in K \subset \overline{D}.
\]

Herein, \( \text{Cor}_{du} \in H^{1,1}(\overline{D} \times \overline{D}) \) satisfies the boundary value problem

\[
(\text{div}_x \otimes \text{div}_y) \left[ (A(x) \otimes A(y)) (\nabla_x \otimes \nabla_y) \text{Cor}_{du}(x,y) \right] = 0, \quad x, y \in \overline{D},
\]

\[
\text{div}_x \left[ A(x) \nabla_x \text{Cor}_{du}(x,y) \right] = 0, \quad x \in \overline{D}, y \in \partial \overline{D},
\]

\[
\text{div}_y \left[ A(y) \nabla_y \text{Cor}_{du}(x,y) \right] = 0, \quad x \in \partial \overline{D}, y \in \overline{D},
\]

\[
\text{Cor}_{du}(x,y) = \text{Cor}_{\kappa}(x,y) \left[ \frac{\partial \overline{\pi}}{\partial n}(x) \otimes \frac{\partial \overline{\pi}}{\partial n}(y) \right], \quad x, y \in \partial \overline{D}.
\]

2.3. PDEs with stochastic coefficients. Finally we should be concerned with stochastic coefficients. To that end, let the stochastic matrix \( A(\cdot, \omega) \in L^\infty(D, \mathbb{R}^{n \times n}) \) be symmetric and positive definite, satisfying almost surely

\[
\underline{\alpha} \| \xi \|^2_{L^2} \leq \xi^T A(x, \omega) \xi \leq \overline{\alpha} \| \xi \|^2_{L^2}, \quad x \in D, \ \xi \in \mathbb{R}^n.
\]

Then, we are interested in the solution of the boundary value problem

\[
- \text{div} \left[ A(x, \omega) \nabla u(x, \omega) \right] = f(x), \quad x \in D, \quad \omega \in \Omega,
\]

\[
u(x, \omega) = g(x), \quad x \in \partial D, \quad \omega \in \Omega.
\]

We proceed likewise to the situation of a stochastic domain and use sensitivity analysis to linearize this problem.

Lemma 1. Consider the uniformly elliptic diffusion matrix \( A \in L^\infty(D, \mathbb{R}^{n \times n}) \) and \( u \in H^1(D) \) being the solution of

\[
- \text{div}[A \nabla u] = f \text{ in } D, \quad u = g \text{ on } \partial D.
\]
Then, the mapping
\[ F : L^\infty(D, \mathbb{R}^{n \times n}) \to \{ v \in H^1(D) : v|_{\partial D} = g \}, \quad A \mapsto F(A) := u \]
is Fréchet differentiable where the derivative \( du = du[dA] \in H^1_0(D) \) in direction \( dA \in L^\infty(D, \mathbb{R}^{n \times n}) \) is given by
\[
(2.9) \quad - \text{div}[A \nabla du] = \text{div}[dA \nabla u] \text{ in } D, \quad du = 0 \text{ on } \partial D.
\]

**Proof.** Let \( A \in L^\infty(D, \mathbb{R}^{n \times n}) \) be a uniformly elliptic matrix and \( dA \in L^\infty(D, \mathbb{R}^{n \times n}) \). For \( \varepsilon > 0 \) sufficiently small consider the solutions \( u \) and \( u_\varepsilon \) of the boundary value problems
\[
- \text{div}[A \nabla u] = f \text{ in } D, \quad u = g \text{ on } \partial D,
\]
\[
- \text{div}[(A + \varepsilon dA) \nabla u_\varepsilon] = f \text{ in } D, \quad u_\varepsilon = g \text{ on } \partial D.
\]
Subtracting both equations and observing that \( u_\varepsilon - u \in H^1_0(D) \), the weak formulation implies for all \( v \in H^1_0(D) \) that
\[
0 = \int_D A(\nabla u_\varepsilon - \nabla u) \nabla v + \varepsilon dA \nabla u_\varepsilon \nabla v dx.
\]
Thus, it holds for all \( v \in H^1_0(D) \) that
\[
\int_D \frac{A \nabla u_\varepsilon - \nabla u}{\varepsilon} \nabla v dx = - \int_D dA \nabla u_\varepsilon \nabla v dx.
\]
Due to
\[
du[dA] = \lim_{\varepsilon \to 0} \frac{u_\varepsilon - u}{\varepsilon}
\]
and \( \|u_\varepsilon - u\|_{H^1_0(D)} \to 0 \) as \( \varepsilon \to 0 \), we conclude that \( du \) is the Gâteaux derivative. Finally, Fréchet differentiability follows from
\[
\|u_\varepsilon - u - \varepsilon du\|_{H^1_0(D)} = \sup_{v \in H^1_0(D)} \frac{1}{\|v\|_{H^1_0(D)}} \int_D A \nabla (u_\varepsilon - u - \varepsilon du) \nabla v dx
\]
\[
= \sup_{v \in H^1_0(D)} \frac{1}{\|v\|_{H^1_0(D)}} \left\{ \int_D [(A + \varepsilon dA) \nabla u_\varepsilon - A \nabla u] \nabla v dx \right\}
\]
\[
= \varepsilon \sup_{v \in H^1_0(D)} \frac{1}{\|v\|_{H^1_0(D)}} \int_D dA \nabla (u_\varepsilon - u) \nabla v dx
\]
\[
\leq \varepsilon \|dA\|_{L^\infty(D, \mathbb{R}^{n \times n})} \|u_\varepsilon - u\|_{H^1_0(D)}.
\]
\( \Box \)
Thus, assuming that
\[ A(x, \omega) = \overline{A}(x) + \varepsilon dA(x, \omega) \]
and \( ||dA(x, \omega)||_{L^2} \leq 1 \) on \( D \) almost surely, we can expand \( u(\omega) \) into a stochastic Taylor expansion
\[ u(\omega) = \overline{u} + \varepsilon du[dA(\omega)] + \frac{\varepsilon^2}{2} d^2u[dA(\omega), dA(\omega)] + O(\varepsilon^3), \quad \text{P-a.e. } \omega \in \Omega, \]
and apply the techniques also used in [20].

**Theorem 2.** Assume \( E_{dA} = 0 \). Then, for \( \varepsilon > 0 \) sufficiently small, it holds
\[ E_u(x) = \overline{u}(x) + O(\varepsilon^2), \quad \text{Cov}_u(x, y) = \varepsilon^2 \text{Cor}_{du}(x, y) + O(\varepsilon^3), \quad x, y \in D. \]
Herein, \( \overline{u} \in H^1(D) \) and \( \text{Cor}_{du} \in H^{1,1}(D \times D) := H^1_0(D) \otimes H^1_0(D) \) satisfy the boundary value problems
\[ -\text{div} \left[ \overline{A}(x) \nabla \overline{u}(x) \right] = f(x), \quad x \in D, \]
\[ \overline{u}(x) = g(x), \quad x \in \partial D, \]
and
\[ \left( \text{div}_x \otimes \text{div}_y \right) \left[ (\overline{A}(x) \otimes \overline{A}(y)) (\nabla_x \otimes \nabla_y) \text{Cor}_{du}(x, y) \right] \]
\[ = \left( \text{div}_x \otimes \text{div}_y \right) \left[ \text{Cor}_{dA}(x, y) (\nabla_x \overline{u}(x) \otimes \nabla_y \overline{u}(y)) \right], \quad x, y \in D, \]
\[ \text{div}_x \left[ \overline{A}(x) \nabla_x \text{Cor}_{du}(x, y) \right] = 0, \quad x \in D, \quad y \in \partial D, \]
\[ \text{div}_y \left[ \overline{A}(y) \nabla_y \text{Cor}_{du}(x, y) \right] = 0, \quad x \in \partial D, \quad y \in D, \]
\[ \text{Cor}_{du}(x, y) = 0, \quad x, y \in \partial D. \]

**Proof.** For sake of simplicity we abbreviate
\[ du(\omega) := du[dA(\omega)], \quad d^2u(\omega) := d^2u[dA(\omega), dA(\omega)]. \]
We apply the stochastic Taylor expansion (2.10) and arrive at
\[ E \left( u(\omega) \right) = E \left( \overline{u} + \varepsilon du(\omega) + O(\varepsilon^2) \right) = \overline{u} + \varepsilon E \left( du(\omega) \right) + O(\varepsilon^2). \]
Due to (2.9) the function \( E_{du} = E \left( du(\omega) \right) \in H^1_0(D) \) satisfies
\[ -\text{div}[A \nabla E_{du}] = \text{div}[E_{dA} \nabla u] \text{ in } D, \quad E_{du} = 0 \text{ on } \partial D, \]
i.e., \( E_{du} = 0 \) since \( E_{dA} = 0 \).
Next, we expand both terms on the right hand side of
\[ \text{Cov}_u(x, y) = E \left( u(x, \omega) \cdot u(y, \omega) \right) - E \left( u(x, \omega) \right) \cdot E \left( u(y, \omega) \right). \]
On the one hand, we get (recall that $E_{du} = 0$)

$$E(u(x, \omega) \cdot u(y, \omega)) = E\left(\left[\pi(x) + \varepsilon du(x, \omega) + \frac{\varepsilon^2}{2} d^2 u(x, \omega) + O(\varepsilon^3)\right] \cdot \left[\pi(y) + \varepsilon du(y, \omega) + \frac{\varepsilon^2}{2} d^2 u(y, \omega) + O(\varepsilon^3)\right]\right)$$

$$= \pi(x) \cdot \pi(y) + \varepsilon^2 E(du(x, \omega) \cdot du(y, \omega))$$

$$+ \frac{\varepsilon^2}{2} \pi(x) E(d^2 u(x, \omega)) + \frac{\varepsilon^2}{2} \pi(y) E(d^2 u(y, \omega)) + O(\varepsilon^3).$$

On the other hand we find

$$E(u(x, \omega)) \cdot E(u(x, \omega))$$

$$= \left(\pi(x) + \frac{\varepsilon^2}{2} E(d^2 u(x, \omega)) + O(\varepsilon^3)\right) \cdot \left(\pi(y) + \frac{\varepsilon^2}{2} E(d^2 u(y, \omega)) + O(\varepsilon^3)\right)$$

$$= \pi(x) \pi(y) + \frac{\varepsilon^2}{2} \pi(x) E(d^2 u(y, \omega)) + \frac{\varepsilon^2}{2} \pi(y) E(d^2 u(x, \omega)) + O(\varepsilon^3).$$

Subtracting both equations in accordance with (2.13) yields

$$\text{Cov}_u(x, y) = \varepsilon^2 E(du(x, \omega) \cdot du(y, \omega)) + O(\varepsilon^3) = \varepsilon^2 \text{Cor}_{du}(x, y) + O(\varepsilon^3).$$

Therein, (2.9) implies that $\text{Cor}_{du} \in H^{1,1}_0(D \times D)$ is given by (2.12). \hfill \Box

### 3. Finite element discretization

#### 3.1. Multiresolution analyses. Let $D \subset \mathbb{R}^n$ be a polygonal and bounded domain with coarse grid triangulation $T_0 = \{\tau_{0,k}\}$. By uniform refinement of each simplex on level $j - 1$ into $2^n$ simplices on level $j$, we obtain recursively the triangulations $T_j = \{\tau_{j,k}\}$ for all $j > 0$. On the triangulation $T_j$ we define standard piecewise continuous Lagrangian finite elements $\Phi_j = \{\varphi_{j,k} : k \in \Delta_j\}$ and get a nested sequence of finite dimensional trial spaces

$$V_0 \subset V_1 \subset \cdots \subset V_j \cdots \subset H^1(\Omega),$$

where

$$V_j = \text{span}\{\varphi_{j,k} : k \in \Delta_j\} = \{u \in C(\Omega) : u|_\tau \in \Pi_d \text{ for all } \tau \in T_j\}$$

and $\dim V_j \sim 2^{jn}$. For our approach it is convenient to normalize the Lagrangian finite elements with respect to the energy space $H^1(D)$, i.e.,

$$\|\varphi_{j,k}\|_{H^1(D)} \sim 1.$$
The trial spaces $V_j$ satisfy the following Jackson and Bernstein type estimates for all $s \leq t < 3/2$, $t \leq q \leq d + 1$

\begin{equation}
\inf_{v_j \in V_j} \| u - v_j \|_{H^t(\Omega)} \lesssim h_j^{3-t} \| u \|_{H^t(\Omega)}, \quad u \in H^q(\Omega),
\end{equation}

and

\begin{equation}
\| v_j \|_{H^t(\Omega)} \lesssim h_j^{s-t} \| v_j \|_{H^s(\Omega)}, \quad v_j \in V_j,
\end{equation}

uniformly in $j$, where we set $h_j := 2^{-j}$. Notice that the parameter $h_j \sim \max_k \{ \text{diam } \tau_{j,k} \}$ refers to the mesh size associated with the subspace $V_j$ on $\Omega$.

3.2. Curved domains and parametric finite elements. In case of non-polygonal domains with piecewise smooth boundary we can use parametric finite elements to realize the multiresolution analysis (3.1). We assume that the domain $D$ is given as a collection of simplicial smooth patches. More precisely, let $\triangle$ denote the reference simplex in $\mathbb{R}^n$. The domain $D$ is partitioned into a finite number of patches

\begin{equation}
D = \bigcup_{k=1}^M \tau_{0,k}, \quad \tau_{0,k} = \kappa_k(\triangle), \quad k = 1, 2, \ldots, M,
\end{equation}

where each $\kappa_k : \triangle \to \tau_{0,k}$ defines a diffeomorphism of $\triangle$ onto $\tau_{0,k}$. The intersection $\tau_{0,k} \cap \tau_{0,k'}$, $k \neq k'$, of the patches $\tau_{0,k}$ and $\tau_{0,k'}$ is supposed to be either $\emptyset$, or a lower dimensional face.

A mesh of level $j$ on $D$ is induced by regular subdivisions of depth $j$ of the reference simplex into $2^{jn}$ triangles. This generates the $2^{jn}M$ elements $\{ \tau_{j,k} \}$. In order to ensure that the collection $\{ \tau_{j,k} \}$ of elements on the level $j$ forms a regular mesh on $D$, the parametric representation is subjected to the following matching condition: A bijective, affine mapping $\Xi : \triangle \to \triangle$ exists such that for all $x = \kappa_i(s)$ on a common interfaces of $\tau_{0,i}$ and $\tau_{0,i'}$ it holds that

$$
\kappa_i(s) = (\kappa_i' \circ \Xi)(s).
$$

In other words, the diffeomorphisms $\kappa_i$ and $\kappa_i'$ coincide at common interface except for orientation. An example of such a triangulation is found in the left plot of Figure 2.

Finally, we can define the ansatz functions via parametrization, lifting Lagrangian finite elements from $\triangle$ to the domain $D$ by using the mappings $\kappa_i$ and gluing across patch boundaries. This yields a sequence of nested finite element spaces (3.1) that satisfy the same Jackson and Bernstein type estimates (3.3) and (3.4) as the standard finite element spaces from the previous subsection.
3.3. Mean field equation. According to Section 2, the mean field equation takes in any case the form

\[ -\text{div} \left[A(x)(\nabla u(x))\right] = f(x), \quad x \in D, \]

\[ u(x) = g(x), \quad x \in \partial D, \]

(3.6)
cf. (2.2), (2.6), and (2.11). To resolve the non-homogeneous Dirichlet data, we need to distinguish between the basis functions \( \{ \varphi_{j,k} : k \in \Delta_j^D \} \) which are supported in the interior of the domain, i.e., \( \varphi_{j,k}|_{\partial D} \equiv 0 \), and boundary functions \( \{ \varphi_{j,k} : k \in \Delta_j^{\partial D} \} \) with \( \varphi_{j,k}|_{\partial D} \neq 0 \). Notice that \( \Delta_j = \Delta_j^D \cup \Delta_j^{\partial D} \) and \( \Delta_j^D \cap \Delta_j^{\partial D} = \emptyset \).

Define the stiffness matrices

\[ A_{j,j'}^{\Theta,\Xi} := \left[ (A\nabla \varphi_{j',l}, \nabla \varphi_{j,k})_{L^2(D)} \right]_{k \in \Delta_j^\Theta, l \in \Delta_{j'}^\Xi}, \quad \Theta, \Xi \in \{ D, \partial D \}, \]

(3.7)

the mass matrix with respect to the boundary

\[ G_{j,j'}^{\partial D,\partial D} := \left[ (\varphi_{j',l}, \varphi_{j,k})_{L^2(\partial D)} \right]_{k \in \Delta_j^{\partial D}, l \in \Delta_{j'}^{\partial D}}, \]

(3.8)

and the data vectors

\[ \mathbf{g}_j^{\partial D} = \left[ (g, \varphi_{j,k})_{L^2(\partial D)} \right]_{k \in \Delta_j^{\partial D}}, \quad \mathbf{f}_j^\Theta = \left[ (f, \varphi_{j,k})_{L^2(D)} \right]_{k \in \Delta_j^\Theta}, \quad \Theta \in \{ D, \partial D \}. \]

Of course, in the present context of traditional finite elements only matrices with \( j = j' \) appear. But later on we will need them also in the situation \( j \neq j' \). To compute an approximate solution

\[ u_j = \sum_{k \in \Delta_j} u_{j,k} \varphi_{j,k} = \sum_{k \in \Delta_j^{\partial D}} u_{j,k} \varphi_{j,k} + \sum_{k \in \Delta_j^D} u_{j,k} \varphi_{j,k} = u_j^D + u_j^{\partial D} \]

of (3.6) we determine first the boundary part \( u_j^{\partial D} \in H^1(\overline{D}) \) such that

\[ G_{j,j'}^{\partial D,\partial D} u_j^{\partial D} = \mathbf{g}_j^{\partial D}. \]

(3.9)

Thus, \( u_j^{\partial D}|_{\partial D} \) is the \( L^2 \)-orthogonal projection of the Dirichlet datum \( g \) onto the discrete trace space \( V_j|_{\partial D} \). Instead, often \( u_j^{\partial D} \) is determined by simply interpolating the Dirichlet data \( g \). However, we considered here (3.9) in order to motivate our proceeding later on in the context of sparse multilevel frames.

Having \( u_j^{\partial D} \) at hand we can compute the domain part \( u_j^D \in H^1_0(D) \) from

\[ A_{j,j'}^{D,\partial D} u_j^D = \mathbf{f}_j^D - A_{j,j'}^{D,\partial D} u_j^{\partial D}. \]

(3.10)

We employ the cg-method to iteratively solve the linear systems of equations (3.9) and (3.10). Using a nested iteration, combined with the BPX-preconditioner [5, 11, 25] in case of (3.10), results in linear over-all complexity [19]. In particular, we conclude the following standard result.
Theorem 3. The approximate solution \( u_j = u_j^D + u_j^D \) from (3.9) and (3.10) is computed in linear complexity. The order of convergence is
\[
\|u - u_j\|_{H^{1-d}(D)} \lesssim h_j^{2d} \|u\|_{H^{d+1}(D)}
\]
provided that the given data are sufficiently smooth.

Proof. We only have to prove the error estimate. To solve the problem
\[
J(u) = \frac{1}{2} \int_D A \nabla u \nabla ud\mathbf{x} - \int_D f ud\mathbf{x} \to \inf_{u \in H^1(D)} u \text{ subject to } u = g \text{ on } \Gamma
\]
we introduce the Lagrange multiplier \( \lambda \in H^{-1/2}(\partial D) \). By discretizing \( u \in H^1(D) \) in \( V_j \) and \( \lambda \) in the discrete trace space \( V_j|_{\partial D} \), that is
\[
\lambda_j = \sum_{k \in \Delta j} \lambda_{j,k} \phi_{j,k}|_{\partial D},
\]
we arrive at the linear system of equations
\[
(3.11) \begin{bmatrix}
A_{j,j}^{\partial D,\partial D} & A_{j,j}^{\partial D,D} & G_{j,j}^{\partial D,\partial D} \\
A_{j,j}^{\partial D,D} & A_{j,j}^{D,D} & 0 \\
G_{j,j}^{\partial D,\partial D} & 0 & 0
\end{bmatrix}
\begin{bmatrix}
u_j^D \\
u_j^D \\
\lambda_j^D \\
\mu_j^D
\end{bmatrix}
= \begin{bmatrix}
f_j^D \\
f_j^D \\
G_j^\partial D \\
G_j^\partial D
\end{bmatrix}.
\]
Herein, the unknowns with respect to \( u_j \) are uniquely determined by (3.9), (3.10). Since the Lagrange multiplier is uniquely determined by \( \lambda = -\partial u/\partial n \) (see [1, 4, 32]) it holds \( \lambda \in H^{d-1/2}(\partial D) \) provided that \( u \in H^{d+1}(D) \). In view of (3.3) the standard estimate in the associated energy space \( H^1(D) \times H^{-1/2}(\partial D) \) gives
\[
\|u - u_j\|_{H^1(D)} + \|\lambda - \lambda_j\|_{H^{-1/2}(\partial D)} \lesssim \inf_{v_j \in V_j} \|u - v_j\|_{H^1(D)} + \inf_{\mu_j \in V_j|_{\partial D}} \|\lambda - \mu_j\|_{H^{-1/2}(\partial D)} \lesssim h^d \|u\|_{H^d(\Omega)}.
\]
Applying the Aubin-Nitsche trick doubles the order of convergence and gives the desired estimate on \( u_j \).

\[\square\]

4. Fast second moment computation

4.1. Sparse tensor product spaces. We are now going to discretize hypo-elliptic boundary value problems on the tensor product domain \( D \times D \) as they are satisfied
by the correlation, i.e., equations of the form
\begin{equation}
      (\text{div}_x \otimes \text{div}_y) \left[ (A(x) \otimes A(y)) (\nabla_x \otimes \nabla_y) u(x, y) \right] = f(x, y), \quad x, y \in D,
\end{equation}
\begin{align*}
    - \text{div}_x [A(x) \nabla_x u(x, y)] &= h(x, y), \quad x \in D, \quad y \in \partial D, \\
    - \text{div}_y [A(y) \nabla_y u(x, y)] &= h(y, x), \quad x \in \partial D, \quad y \in D, \\
    u(x, y) &= g(x, y), \quad x, y \in \partial D.
\end{align*}

Instead of the full tensor product space
\[ V_J \otimes V_J = \sum_{j, j' \leq J} V_j \otimes V_{j'} \subset H^{1,1}(D \times D), \]
we shall consider the \textit{sparse} tensor product space
\[ \hat{V}_J = \sum_{j + j' \leq J} V_j \otimes V_{j'} \subset H^{1,1}(D \times D). \]

Abbreviating \( N_J := \dim V_J \) there holds \( \hat{N}_J := \dim \hat{V}_J \sim N_J \log N_J \), cf. [8], which is substantially smaller than the dimension \( N_J^2 \) of the full tensor product space \( V_J \otimes V_J \).

The following lemma, proven in [26, 30], tells us that the approximation power in the sparse tensor product spaces is nearly as good as in the full tensor product spaces, provided that the given function has some extra regularity in terms of bounded mixed derivatives.

**Lemma 4.** For \( 0 \leq s \leq 3/2 \), \( s \leq t \leq d \) there holds the error estimate
\[
    \inf_{\hat{u}_J \in \hat{V}_J} \|u - \hat{u}_J\|_{H^{s,t}(D \times D)} \lesssim \begin{cases} 
        2^{J(s-t)} \sqrt{J} \|u\|_{H^{s,s}(D \times D)}, & \text{if } t = d + 1, \\
        2^{J(s-t)} \|u\|_{H^{s,t}(D \times D)}, & \text{otherwise},
    \end{cases}
\]
provided that \( u \in H^{s,t}(D \times D) \).

\[ \frac{4}{5} \]

**4.2. Galerkin discretization.** To approximate functions in \( \hat{V}_J \) one traditionally uses hierarchical bases like wavelet or multilevel bases, see for example [6, 7, 8, 15, 33]. Here we will use multilevel frames as proposed in [21], i.e., we will represent functions by the redundant but stable collection
\[ \hat{\Phi}_J := \{ \varphi_{j,k} \otimes \varphi_{j',k'} : k \in \Delta_j, \ k' \in \Delta_{j'}, \ j + j' \leq J \}. \]

It has been shown in [21] that \( \text{card}(\hat{\Phi}_J) \sim \hat{N}_J \sim N_J \log N_J \), i.e., this set has still optimal cardinality. We like to remark that the frame \( \hat{\Phi}_J \) is the restriction to \( \hat{V}_J \) of the two-fold tensor product of the frame that underlies the BPX-preconditioner, see [21] for the details.
To solve (4.1) by the Galerkin method we need the data vectors

$$\hat{f}^{D, D} = \left( f, \varphi^{D}_{j,k} \otimes \varphi^{D}_{j',k'} \right)_{k \in \Delta^{D}_j, k' \in \Delta^{D}_{j' + \cdot}, j \leq J},$$

$$\hat{h}^{D, \partial D} = \left( h, \varphi^{D}_{j,k} \otimes \varphi^{\partial D}_{j',k'} \right)_{k \in \Delta^{D}_j, k' \in \Delta^{\partial D}_{j' + \cdot}, j \leq J},$$

$$\hat{h}^{\partial D, D} = \left( h, \varphi^{\partial D}_{j,k'} \otimes \varphi^{D}_{j',k} \right)_{k \in \Delta^{\partial D}_j, k' \in \Delta^{D}_{j' + \cdot}, j \leq J},$$

$$\hat{g}^{\partial D, \partial D} = \left( g, \varphi^{\partial D}_{j,k} \otimes \varphi^{\partial D}_{j',k} \right)_{k \in \Delta^{\partial D}_j, k' \in \Delta^{\partial D}_{j' + \cdot}, j \leq J}.$$

Notice that there holds

$$\left[ \hat{h}^{D, \partial D} \right]_{(j,k),(j',k')} = \left[ \hat{h}^{\partial D, D} \right]_{(j',k'),(j,k)}, \quad k \in \Delta^{\partial D}_j, \quad k' \in \Delta^{D}_{j'}, \quad j + j' \leq J.$$

Moreover, based on the two-fold tensor products of the finite element matrices (3.7) and (3.8), we introduce the stiffness matrices

$$\hat{A}^{\Theta, \Xi}_j = \left[ A^{D, \Theta}_{j_1, j_2} \otimes A^{\Xi, \Xi}_{j_1', j_2'} \right]_{j_1 + j_2, j_1', j_2', j_1 + j_2', j_1' + j_2' \leq J}, \quad \Theta, \Xi \in \{D, \partial D\},$$

the mass matrix

$$\hat{G}^{\partial D, \partial D}_j = \left[ G^{\partial D, \partial D}_{j_1, j_2} \otimes G^{\partial D, \partial D}_{j_1', j_2'} \right]_{j_1 + j_2, j_1', j_2', j_1 + j_2', j_1' + j_2' \leq J},$$

and the mixed matrices

$$\hat{B}^{\Theta, \partial D}_j = \left[ A^{D, \Theta}_{j_1, j_2} \otimes G^{\partial D, \partial D}_{j_1, j_2'} \right]_{j_1 + j_2, j_1', j_2', j_1 + j_2', j_1' + j_2' \leq J}, \quad \Theta \in \{D, \partial D\},$$

$$\hat{C}^{\partial D, \Theta}_j = \left[ G^{\partial D, \partial D}_{j_1, j_2} \otimes A^{D, \Theta}_{j_1', j_2'} \right]_{j_1 + j_2, j_1', j_2', j_1 + j_2', j_1' + j_2' \leq J}, \quad \Theta \in \{D, \partial D\}.$$

We shall separate the degrees of freedom of $\hat{u}_j \in \hat{V}_j$ in order to compute the solution to (4.1) successively. To this end, let be

$$\hat{u}_j = \sum_{j + j' \leq J} \sum_{k \in \Delta^D_j} \sum_{k' \in \Delta^\Xi^D_j} \hat{u}_{(j,k),(j',k')}(\varphi_{j,k} \otimes \varphi_{j',k'}) = \hat{u}^{D, D}_j + \hat{u}^{D, \partial D}_j + \hat{u}^{\partial D, D}_j + \hat{u}^{\partial D, \partial D}_j,$$

where

$$\hat{u}^{\Theta, \Xi}_j := \sum_{j + j' \leq J} \sum_{k \in \Delta^\Theta^D_j} \sum_{k' \in \Delta^\Xi^D_j} \hat{u}_{(j,k),(j',k')}(\varphi_{j,k} \otimes \varphi_{j',k'}), \quad \Theta, \Xi \in \{D, \partial D\}.$$

Likewise to the mean field equation we resolve first the Dirichlet data. We determine $\hat{u}_j^{\partial D, \partial D}$ such that $\left. \hat{u}_j^{\partial D, \partial D} \right|_{\partial D \times \partial D}$ coincides with the $L^2$-orthogonal projection of the given Dirichlet data onto the discrete trace space $\hat{V}_j|_{\partial D \times \partial D}$, that is (cf. (3.9))

$$\hat{G}^{\partial D, \partial D}_j \hat{u}_j^{\partial D, \partial D} = \hat{g}^{\partial D, \partial D}_j.$$
Next, we need to compute $\hat{u}_j^{\partial D}$ such that $(\hat{u}_j^{\partial D} + \hat{u}_j^{D})|_{D \times \partial D} \in H^1(D) \otimes H^{1/2}(\partial D)$ satisfies the boundary condition on $D \times \partial D$. In complete analogy we determine $\hat{u}_j^{\partial \partial D}$. Both is done by combining (3.9) and (3.10), namely

\[
\hat{B}_j^{\partial D} \hat{u}_j^{\partial D} = \hat{H}_j^{\partial D} - \hat{B}_j^{\partial D} \hat{u}_j^{\partial D},
\]

\[
\hat{C}_j^{\partial \partial D} \hat{u}_j^{\partial \partial D} = \hat{H}_j^{\partial \partial D} - \hat{C}_j^{\partial \partial D} \hat{u}_j^{\partial \partial D}.
\]

Finally, by tensorizing (3.10), we can compute the function $\hat{u}_j^{D,D} \in H^1_D(D \times D)$ inside the tensor product domain $D \times D$:

\[
\hat{A}_j^{D,D} \hat{u}_j^{D,D} = \hat{f}_j^{D,D} - \hat{A}_j^{\partial D,\partial D} \hat{u}_j^{\partial D} - \hat{A}_j^{D,\partial D} \hat{u}_j^{D,\partial D} - \hat{A}_j^{\partial D,D} \hat{u}_j^{\partial D,D}.
\]

**Proposition 5.** The approximate solution $\hat{u}_j \in \hat{V}_j$ to (4.1) satisfies the error estimate

$$\|u - \hat{u}_j\|_{H^1(D 	imes D)} \lesssim 2^{-2J}\|u\|_{H^{1,1}(D \times D)}$$

provided that the given data are sufficiently smooth.

**Proof.** The assertion follows likewise to the proof of Theorem 3 by tensorizing (3.11) and using Lemma 4. \qed

### 4.3. Iterative solution of the linear systems of equations.

Due to the non-uniqueness of the representation of functions in frame coordinates, system matrices corresponding to bijective operators have a large kernel. However, the associated right hand side vectors lie in the image of the system matrix. Thus, Krylov subspace methods converge without further modifications (see, e.g., [10, 15, 16, 23]). This stems from the fact that the Krylov subspace, and thus the residuum, stays orthogonal to the kernel. The symmetry of the identity and the differential operator implies the symmetry of the system matrices in (4.5), (4.6), and (4.7). Hence, we can apply the conjugate gradient method to solve the linear systems of equations (4.5), (4.6), and (4.7).

According to [21] the sparse multilevel frame is stable in $H^s(D \times D)$ for all $3/2 > s > 0$ provided properly scaled ansatz functions. Our scaling (cf. (3.2)) implies $H^{1,1}(D \times D)$-stability which induces the well-conditioning of the stiffness matrix $\hat{A}_j^{D,D}$ in the sense of

$$\min \left\{ \lambda \in \sigma \left( \left( \hat{A}_j^{D,D} \right)^T \hat{A}_j^{D,D} \right) : \lambda > 0 \right\} \sim \max \{ \sigma \left( \left( \hat{A}_j^{D,D} \right)^T \hat{A}_j^{D,D} \right) \} \sim 1.$$

Therefore, the conjugate gradient method will converge with a convergence rate that is independent of the discretization level $J$ (e.g. [10, 15, 23]).

Since the frame just cannot be scaled such that it is stable $L^2$, a diagonal scaling of the matrices $\hat{G}_j^{D,\partial D}$, $\hat{B}_j^{\partial D}$, and $\hat{C}_j^{D,\partial D}$ yields bounded nonzero eigenvalues except for logarithmic factors. Nevertheless, the over-all complexity is not affected since the
number of boundary functions is considerably smaller than the number of interior functions.

4.4. Prolongations and restrictions. Standard finite element tools provide only system matrices $S_{j,j'}$ in the case $j = j'$. However, also matrices with $j \neq j'$ occur in the multilevel frame discretization, cf. (4.2), (4.3), and (4.4). Fortunately, these matrices can be provided by using restrictions and prolongations.

Let $V_0 \subset V_1 \subset \cdots$ be a given sequence of finite element spaces. We denote the restriction of the function

$$f_j = \sum_{k \in \Delta_j} f_{j,k} \varphi_{j,k} = \Phi_j f_j \in V_j$$

to the space $V_\ell$, $\ell < j$, by $I^\ell_j$. The corresponding discrete operator will be denoted by $I^\ell_j$, that is

$$I^\ell_j f_j = \Phi_I f_j \in V_\ell.$$

Conversely, $I^j_\ell$ resp. $I^j_\ell$ denotes the prolongation of $f_\ell = \Phi_\ell f_\ell \in V_\ell$ onto $V_j$. Both the application of the restriction $I^\ell_j$ and the prolongation $I^j_\ell$ to a vector is of complexity $O(2nj) \sim \dim V_j$.

Invoking restriction and prolongation we obviously have

(4.8) $$S_{j,j'} = \begin{cases} I^j_\ell S_{j',j'}, & j \leq j', \\ S_{j,j'} I^j_\ell, & j > j'. \end{cases}$$

Since we deal with local operators, the finite element system matrices $S_{j,j'}$ have only $O(1)$ nonzero coefficients per column and row, independently of the level $j$. Thus, employing (4.8), the matrix-vector multiplication $S_{j,j'} \mathbf{x}$ can be performed in $O(2^{\max(j,j')}) \sim \max\{\dim V_j, \dim V_{j'}\}$ operations, which is order-optimal.

In view of the Galerkin discretization from Subsection 4.2 we also have to deal with a second sequence of finite element spaces, say $W_0 \subset W_1 \subset \cdots$ with $\dim W_j \sim 2^{mj}$. Associated prolongations and restrictions will be denoted by $J^j_\ell$ and $J^\ell_j$ ($j > \ell$).

4.5. Fast two-factor matrix-vector multiplication. We shall provide a fast two-factor matrix-vector multiplications of the following type

(4.9) $$\tilde{y}_j = \tilde{U}_J \tilde{x}_j, \quad \tilde{U}_J = [S_{j_1,j_2} \otimes T_{j_1',j_2'}]_{j_1+j_2,j_1'+j_2' \leq j}, \quad \tilde{x}_j = [x_{j,j'}]_{j+j' \leq j}.$$

The matrices $S_{j,j'}$ and $T_{j,j'}$ are defined with respect to two different multiresolution sequences $\{V_j\}_{j \geq 0}$ and $\{W_j\}_{j \geq 0}$. We mention that the subsequent algorithm is a generalization of the algorithm from [21] where we considered the situation of one ansatz space $V_j = W_j$. Algorithms that employ similar techniques as the presented one have been developed in [2, 3, 21, 27, 29, 30]
We fix first some notation. For a matrix

$$S = \begin{bmatrix} s_1 & s_2 & \cdots & s_q \end{bmatrix} \in \mathbb{R}^{p \times q}, \quad s_i \in \mathbb{R}^p$$

we define \(\text{vec}(S)\) as the column vector

$$\text{vec}(S) = \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_q \end{bmatrix} \in \mathbb{R}^{p \times q}.$$  

Then, for given matrices \(T \in \mathbb{R}^{k \times \ell}, X \in \mathbb{R}^{x \times p}, S \in \mathbb{R}^{q \times p},\) and \(Y \in \mathbb{R}^{k \times q},\) there holds the identity

$$\text{(4.10)} \quad \text{vec}(Y) = (S \otimes T) \text{vec}(X) \iff TXS^T = Y.$$  

Based on the equivalence (4.10) we realize a fast matrix-vector multiplication. To this end, we assume that the vector \(\tilde{X}_J = [\tilde{X}_{j_1,j_2}]_{j_1+j_2 \leq J}\) is blockwise stored in matrix form, i.e. \(\tilde{X}_{j_1,j_2} \in \mathbb{R}^{i_1 \times i_2}\). Then, for the matrix-vector multiplication (4.9) we have to compute products of the form

$$\text{vec}(z) = (S_{j_1,j_1'} \otimes T_{j_2,j_2'}) \text{vec}(\tilde{X}_{j_1',j_2'}).$$  

Using (4.10), this means that

$$z = T_{j_2,j_2'} \tilde{X}_{j_1,j_1'}^T S_{j_1,j_1'}, \quad j_1 + j_2 \leq J, \quad j_1' + j_2' \leq J.$$  

To achieve essentially optimal complexity bounds, the multiplications must be performed in the right order, namely

$$z = \begin{cases} 
T_{j_2,j_2'} (\tilde{X}_{j_1,j_1'}^T S_{j_1,j_1'}), & j_1 + j_2 m \leq j_1' n + j_2 m, \\
(T_{j_2,j_2'} \tilde{X}_{j_1,j_1'}) S_{j_1,j_1'}, & j_1 + j_2 m > j_1' n + j_2 m.
\end{cases}$$

Observing (4.8), in case of \(j_1 n + j_2 m \leq j_1' n + j_2 m\) we compute

$$\text{(4.11)} \quad y := S_{j_1,j_1'} \tilde{X}_{j_1',j_2'} = \begin{cases} 
I_{j_1}^T S_{j_1,j_1'} \tilde{X}_{j_1',j_2'}, & j_1 \leq j_1', \\
S_{j_1,j_1'} I_{j_1}^T \tilde{X}_{j_1',j_2'}, & j_1 > j_1',
\end{cases}$$

$$\text{(4.12)} \quad z := T_{j_2,j_2'} y^T = \begin{cases} 
J_{j_2}^T T_{j_2,j_2'} y^T, & j_2 \leq j_2', \\
T_{j_2,j_2'} J_{j_2}^T y^T, & j_2 > j_2'.
\end{cases}$$

Therein, one needs \(O(2^{\max(j_1,j_1')} + j_2 m)\) operations to get \(y\) by (4.11) and additional \(O(2^{\max(j_2,j_2')} + j_1 n)\) operations to derive \(z\) by (4.12). Thus, abbreviating \(x := j_1 n + j_2 m\) and \(x' = j_1' n + j_2' m\), the complexity to compute \(z\) is bounded by \(O(2^{\max(x,x',j_1 n + j_2 m)})\).

Since \(j_1 n + j_2 m \leq x' - j_2 m + x - j_1 n \leq 2 \max\{x, x'\} - j_1 n - j_2 m'\) implies

$$j_1 n + j_2 m \leq x' - j_2 m + x - j_1 n \leq 2 \max\{x, x'\} - j_1 n - j_2 m'.$$
we conclude \( j_1 n + j'_2 m \leq \max \{ x, x' \} \). Consequently, the computation of \( z \) is of complexity \( O(2^{\max \{ j_1 n + j_2 m, j'_1 n + j'_2 m \}}) \) provided that \( j_1 n + j'_2 m \leq j'_1 n + j_2 m \).

In case of \( j_1 n + j'_2 m > j'_1 n + j_2 m \) we compute

\[
\tag{4.13}
y := T_{j_2 j'_2} \hat{x}^\top_{j_1 j'_2} = \begin{cases} J^T_{j'_2} T_{j_2 j'_2} \hat{x}^\top_{j'_2 j'_2}, & j'_2 \leq j_2, \\
T_{j_2 j'_2} J^T_{j'_2} \hat{x}^\top_{j'_2 j'_2}, & j'_2 > j_2, \end{cases}
\]

\[
\tag{4.14}
z^\top := S_{j_1 j'_1} y^\top = \begin{cases} P_{j'_1} S_{j_1 j'_1} y^\top, & j_1 \leq j'_1, \\
S_{j_1 j'_1} P_{j'_1} y^\top, & j_1 > j'_1. \end{cases}
\]

Using the same argument as above we find that \( z \) is also computed in complexity \( O(2^{\max \{ j_1 n + j_2 m, j'_1 n + j'_2 m \}}) \).

With the above preparations at hand we can formulate the following algorithm which performs the matrix-vector multiplication \( \hat{y}_J = \hat{S}_J \hat{x}_J \):

**Algorithm 6** (Sparse tensor product matrix-vector multiplication).

**input:** finite element matrices \( S_{j,j} \) and \( T_{j,j} \) (\( 0 \leq j \leq J \))

**vector** \( \hat{x}_J = [\hat{x}_{j,j'}]_{0 \leq j + j' \leq J} \)

**output:** blockwise stored vector \( \hat{y} = [\hat{y}_{j,j'}]_{0 \leq j + j' \leq J} \)

\[ \text{for all } 0 \leq j_1 + j_2 \leq J \text{ do begin} \]

\[ \text{initialize } \hat{y}_{j_1,j_2} := 0 \]

\[ \text{for all } 0 \leq j'_1 + j'_2 \leq J \text{ do begin} \]

\[ \text{if } (j_1 + j'_2 \leq j'_1 + j_2) \text{ then} \]

\[ \text{update } \hat{y}_{j_1,j_2} := \hat{y}_{j_1,j_2} + z \text{ with } z \text{ from } (4.11), (4.12) \]

\[ \text{else} \]

\[ \text{update } \hat{y}_{j_1,j_2} := \hat{y}_{j_1,j_2} + z \text{ with } z \text{ from } (4.13), (4.14) \]

\[ \text{end} \]

\[ \text{end} \]

\[ \text{end} \]

**Proposition 7.** Algorithm 6 computes the matrix-vector product (4.9) in essentially linear complexity. Precisely, Algorithm 6 consumes \( O(\max \{ m^3, n^3 \} 2^J \max \{ m, n \}) \) operations.

**Proof.** As we have seen both block matrix-vector products (4.11), (4.12) and (4.13), (4.14) have complexity \( O(2^{\max \{ j_1 n + j_2 m, j'_1 n + j'_2 m \}}) \). By using

\[
2^{\max \{ j_1 n + j_2 m, j'_1 n + j'_2 m \}} \leq 2^{\max \{ j_1 + j_2, j'_1 + j'_2 \} \max \{ m, n \}}
\]

the assertion follows as in [21].
5. Numerical Examples

5.1. An analytical example. Our first test is concerned with the convergence rates of the Galerkin discretization in the sparse tensor product spaces. For test reasons we consider the tensor product boundary value problem

\[
\begin{align*}
(\Delta_x \otimes \Delta_y) u(x, y) &= f_x(x) f_y(y), & x, y &\in D, \\
-\Delta_x u(x, y) &= f_x(x) g_y(y), & x &\in D, y \in \partial D, \\
-\Delta_y u(x, y) &= g_x(x) f_y(y), & x \in \partial D, y &\in D, \\
u(x, y) &= g(x) g_y(y), & x, y &\in \partial D.
\end{align*}
\]

The solution of this problem is just the product \( u(x, y) = u_x(x) u_y(y) \) with \( u_x, u_y \in H^1(D) \) being defined by

\[
\begin{align*}
-\Delta u_x &= f_x \text{ in } D, & u_x &= g_x \text{ on } \partial D, \\
-\Delta u_y &= f_y \text{ in } D, & u_y &= g_y \text{ on } \partial D.
\end{align*}
\]

Thus, choosing

\[
\begin{align*}
g_x(x_1, x_2) &= x_1^2 + x_2^2, & f_x(x_1, x_2) &= -4, \\
g_y(y_1, y_2) &= (y_1 - 1)(y_2 - 1), & f_y(y_1, y_2) &= 0,
\end{align*}
\]

the solution of (5.1) is

\[
u(x_1, x_2, y_1, y_2) = (x_1^2 + x_2^2)(y_1 - 1)(y_2 - 1).
\]

Our numerical implementation is so far for two dimensions, based on the piecewise linear parametric finite elements as introduced in Subsection 3.2. We choose \( D \) as the unit circle, represented by four three-sided patches (see the left plot of Figure 2).

We first compute the \( L^2 \)-errors of the approximation, namely the errors

\[
\begin{align*}
E_1 &= \| u - u_j^{\partial D, \partial D} \|_{L^2(\partial D \times \partial D)}, \\
E_2 &= \| u - u_j^{\partial D, D} \|_{L^2(\partial D \times D)} + \| u - u_j^{D, \partial D} \|_{L^2(D \times \partial D)}, \\
E_3 &= \| u - u_j \|_{L^2(\partial D \times \partial D)}.
\end{align*}
\]

The corresponding relative errors are plotted in Figure 1. We also plotted, entitled by “\( E_4 \)”, the relative \( L^2(D \times D) \)-error of the full tensor product approximation. Second order approximation \( O(4^{-j}) \), realized by the full tensor product approximation, is indicated by the dashed lines. Indeed, the sparse tensor product approximation is essentially of this order. It is even nearly as good as the the full tensor product approximation. But on level 7 we have only 302300 unknowns in the sparse tensor
product space instead of 1 billion unknowns, required to discretize in the full tensor product space.

![Figure 1](image)

**Figure 1.** The $L^2$-errors of the sparse tensor product approximation.

The sparse multilevel frame approach is highly efficient even though the over-all computing times exhibit strong logarithmic factors. We need 0.1, 1, 6, 44, 295, 1837, 11061 seconds for the computations with 1, 2, . . . , 7 levels on a standard PC with 2.2 GHz Intel Core 2 Duo processor and 4 GB memory.

5.2. Variance computations. We consider the boundary value problem

$$-\Delta u(x, \omega) = f(x, \omega), \quad x \in D, \quad \omega \in \Omega,$$

$$u(x, \omega) = g(x, \omega), \quad x \in \partial D, \quad \omega \in \Omega,$$

with $D$ being the unit circle. Firstly, we consider $f(x, \omega) = 4$ and $g(x, \omega)$ with $E_g(x) = 0$ and Gaussian correlation $\text{Cor}_g(x, y) = \exp(-\|x-y\|^2)$. Secondly, we consider $g(x, \omega) = 0$ and $f(x, \omega)$ with $E_f(x) = 4$ and Gaussian correlation $\text{Cor}_f(x, y) = 50\cdot\exp(-\|x-y\|^2)$. In both cases, we get the mean shown in Figure 2. The associated variance functions are shown in Figure 3.

We emphasize, that in case of stochastic domains with $\overline{D}$ being the unit circle, $f = 2$ and $g = 0$ and random boundary variations with Gaussian correlation we
Figure 2. Triangulation of the circle (level 4) and the mean of $u$.

Figure 3. Variance of the solution in case of Gaussian correlation and zero load (left) resp. homogeneous Dirichlet data (right).

obtain the same variance as in the left plot of Figure 3. It turns out that the solution’s variance increases when approaching the boundary of the domain, i.e., the solution’s sensitivity with respect to boundary perturbations is the larger the nearer the boundary.
REFERENCES


**Helmut Harbrecht, Institut für Numerische Simulation, Universität Bonn, Wegelerstr. 6, 53115 Bonn, Germany.**

*E-mail address: harbrecht@ins.uni-bonn.de*
Verzeichnis der erschienenen Preprints ab No. 385

385. Albeverio, Sergio; Hryniv, Rostyslav; Mykytyuk, Yaroslav: Factorisation of Non-Negative Fredholm Operators and Inverse Spectral Problems for Bessel Operators

386. Otto, Felix: Optimal Bounds on the Kuramoto-Sivashinsky Equation

387. Gottschalk, Hanno; Thaler, Horst: A Comment on the Infra-Red Problem in the AdS/CFT Correspondence

388. Dickopf, Thomas; Krause, Rolf: Efficient Simulation of Multi-Body Contact Problems on Complex Geometries: A Flexible Decomposition Approach Using Constrained Minimization


390. Nemitz, Oliver; Nielsen, Michael Bang; Rumpf, Martin; Whitaker, Ross: Finite Element Methods on Very Large, Dynamic Tubular Grid Encoded Implicit Surfaces

391. Barbu, Viorel; Marinelli, Carlo: Strong Solutions for Stochastic Porous Media Equations with Jumps

392. Albeverio, Sergio; Cebulla, Christof: Synchronizability of Stochastic Network Ensembles in a Model of Interacting Dynamical Units; erscheint in: Physica A: Statistical Mechanics and its Applications

393. Albeverio, Sergio; Cebulla, Christof: Synchronizability of a Stochastic Version of FitzHugh-Nagumo Type Neural Oscillator Networks

394. Bacher, Kathrin: On Borell-Brascamp-Lieb Inequalities on Metric Measure Spaces

395. Croce, Roberto; Griebel, Michael; Schweitzer, Marc Alexander: Numerical Simulation of Droplet-Deformation by a Level Set Approach with Surface Tension

396. Schweizer, Nikolaus: Spectral Gaps of Simulated Tempering for Bimodal Distributions

397. Koivisto, Juha; Rost, Martin; Alt, Wolfgang: A Phase Field Model for Phospholipid Surfactant

398. Reiter, Philipp; Felix, Dieter; von der Mosel, Heiko; Alt, Wolfgang: Energetics and Dynamics of Global Integrals Modeling Interaction between Stiff Filaments

400. Lapid, Erez; Müller, Werner: Spectral Asymptotics for Arithmetic Quotients of $SL(n,\mathbb{R})/SO(n)$

401. Finis, Tobias; Lapid, Erez; Müller, Werner: On The Spectral Side of Arthur’s Trace Formula II

402. Griebel, Michael; Knapek, Stephan: Optimized General Sparse Grid Approximation Spaces for Operator Equations

403. Griebel, Michael; Jager, Lukas: The BGY3dM Model for the Approximation of Solvent Densities

404. Griebel, Michael; Klitz, Margit: Homogenization and Numerical Simulation of Flow in Geometries with Textile Microstructures

405. Dickopf, Thomas; Krause, Rolf: Weak Information Transfer between Non-Matching Warped Interfaces

406. Engel, Martin; Griebel, Michael: A Multigrid Method for Constrained Optimal Control Problems

407. Nepomnyaschikh, Sergey V.; Scherer, Karl: A Domain Decomposition Method for the Helmholtz-Problem

408. Harbrecht, Helmut: A Finite Element Method for Elliptic Problems with Stochastic Input Data