NONPARAMETRIC DENSITY ESTIMATION FOR RANDOMLY PERTURBED ELLIPTIC PROBLEMS III: CONVERGENCE, COMPLEXITY, AND GENERALIZATIONS

DONALD ESTEP, MICHAEL J. HOLST, AND AXEL MÅLQVIST

Abstract. This is the last in a series of three papers on nonparametric density estimation for randomly perturbed elliptic problems. In the previous papers [3, 4] an efficient algorithm for propagation of uncertainty into a quantity of interest computed from numerical solutions of an elliptic partial differential equation was presented, analyzed, and applied to different problems in e.g. oil reservoir simulation. In this paper we focus on convergence, complexity, and generalizations. The convergence result is a new and crucial contribution. The proof is based on the assumption that the underlying domain decomposition algorithm converges geometrically. The main ideas of the proof can be applied to a large class of domain decomposition algorithms.

1. Introduction

The practical application of differential equations to model physical phenomena presents both mathematical challenges, e.g., the need to compute approximate solutions to difficult problems, and statistical challenges, e.g., the need to incorporate experimental data and model uncertainty. A prototypical example is accounting for the effects of uncertainty in input parameters on the values of a quantity of interest computed from numerical solutions of an elliptic partial differential equation. In particular, we consider the problem of computing statistical information about a quantity of interest $Q(U)$ of the solution $U \in \mathcal{H}^1_0(\Omega)$ of,

$$\begin{cases} -\nabla \cdot \mathcal{A}\nabla U = f, & \text{in } \Omega, \\ U = 0, & \text{on } \partial\Omega, \end{cases} \tag{1.1}$$

where $f \in L^2(\Omega)$ is a given deterministic function, Ω is a polygon domain with boundary $\partial\Omega$ and $\mathcal{A}(x)$ is a stochastic function that varies randomly according to a given probability structure. The problem (1.1) is interpreted to hold almost surely (a.s.) i.e. with probability 1. Under suitable assumptions, e.g. $\mathcal{A}$ is uniformly bounded and uniformly coercive and has piecewise smooth dependence on its inputs (a.s.) with continuous and bounded covariance functions, $Q(U)$ is a random variable. We may describe its stochastic properties by computing its cumulative probability distribution.

The standard Monte Carlo, i.e. nonparametric density estimation, approach to this problem involves choosing sample values for $\mathcal{A}$, computing the corresponding solutions

Date: May 12, 2010.

A. Målqvist's work is supported by The Göran Gustafsson Foundation.

U along with the corresponding quantity of interest values $Q(U)$, and then computing statistics from the collection of values $\{Q(U)\}$. This is generally expensive due to the cost of computing numerical solutions of (1.1) while the accuracy of the results is affected by both the use of a finite number of samples and the numerical error of each computed sample. The latter error typically varies to a significant extent as the parameter values vary.

In the first two papers of this series of three papers, we attacked these issues. First, we construct an efficient numerical method for approximating the cumulative density function for the output distribution. The method is efficient in the sense that the number of stiffness matrices that are inverted, which is the primary cost of solving (1.1), is fixed independent of the number of samples used to compute the distribution. An interesting way to interpret this method is that it reorders the standard approach of "statistics on the outside and solves on the inside". Second, we derive an a posteriori error estimate for the computed probability distribution that accounts for all sources of discretization errors and sample uncertainties. We may interpret this as a complete uncertainty quantification for the problem (1.1). Third, we devise a general adaptive algorithm based on the estimate which provides the means to balance computational effort to control various sources of error and uncertainty in order to achieve a desired accuracy.

The main goal of this final paper is the derivation of a priori error bounds that guarantee the convergence of the method under weak assumptions on the underlying domain decomposition algorithm on which it is based. In addition, we extend the method to cover a more general type of random perturbation and a wider range of underlying domain decomposition solvers.

The paper is organized as follows. In section 2, we present modeling assumptions, the computational method, improvements, and an a posteriori error estimate. In section 3, we state the main results including convergence and a priori error bounds of the method. In section 4, we make some further observations about the method. Finally, in section 5 we present the proofs of the statements in section 3.

2. Modeling assumption and the computational method

Let $\{\mathcal{A}^n \in L^\infty(\Omega)\}_{n=1}^{\mathcal{N}}$ be a collection of sample values of the stochastic function $\mathcal{A}$. For each sample n, the weak form of (1.1) reads: Compute $U^n \in \mathcal{H}_0^1(\Omega)$ solving

$$(\mathcal{A}^n \nabla U^n, \nabla v) = (f, v), \quad \text{for all } v \in \mathcal{H}_0^1(\Omega),, \tag{2.1}$$

where $f \in L^2(\Omega)$ and $(\cdot,\cdot)$ denotes the $L^2(\Omega)$ scalar product. Standard results imply that (2.1) has a unique solution for each $n = 1, \ldots, \mathcal{N}$.

2.1. A modeling assumption.

We assume that the stochastic diffusion coefficient can be written

$$\mathcal{A} = a + A,$$

where the uniformly coercive, bounded deterministic function a may have multiscale behavior and $|A(x)| \leq \delta a(x)$ for some $0 < \delta < 1$. Given a choice of δ, this insures that $a + A \geq \alpha > 0$ for all $x \in \Omega$ for some $\alpha > 0$.

We now make the modeling assumption that A is a piecewise constant function with random coefficients. We let $\{\Omega_d,\ d = 1, \cdots, \mathcal{D}\}$, be a decomposition of Ω into a finite set of non-overlapping polygonal subdomains with $\cup\Omega_d = \Omega$. We denote the boundaries $\partial\Omega_d$ and outward normals $\boldsymbol{n}_d$. We let χ_{Ω_d} denote the characteristic function for the set Ω_d. We assume that

$$A(x) = \sum_{d=1}^{\mathcal{D}} A^d\ \chi_{\Omega_d}(x), \quad x \in \Omega, \tag{2.2}$$

where $\left(A^d\right)$ is a vector of real numbers.

This assumption is reasonable in context of a common experimental situation in which the coefficients A^d are measured at specific points in the domain Ω. In this context, it is natural to assume that the values A^d are random as a way of describing experimental error. Improving the model requires both taking more measurements A^d corresponding to a finer partition of Ω and decreasing the variation in the measured values.

2.2. **Notation.** We let Ω denote the piecewise polygonal computational domain with boundary $\partial\Omega$ in two or three spatial dimensions. For an arbitrary domain $\omega \subset \Omega$ we denote the interior $L^2(\omega)$ norm and the boundary $L^2(\partial\omega)$ norm by $\| \ \|_{L^2(\omega)}$ and $\| \ \|_{L^2(\partial\omega)}$, respectively. We let $\mathcal{H}^s(\omega)$ denote the standard Sobolev space of smoothness s with standard norm $\| \cdot \|_{\mathcal{H}^s(\Omega)}$, for $s \geq 0$. In particular, $\mathcal{H}^1_0(\Omega)$ denotes the space of functions in $\mathcal{H}^1(\Omega)$ with vanishing trace. See [1] for an extensive discussion on these function spaces.

We assume that any random vector X is associated with a probability space $(\Lambda, \mathcal{B}, P)$ in the usual way. We let $\{X^n, n = 1, \cdots, \mathcal{N}\}$ denote a collection of samples. We assume it is understood how to draw these samples. Let $\mathcal{A}^n = a + A^n$ be a particular sample of the diffusion coefficient with corresponding solution U^n. On Ω_d, we denote a finite set of samples by $\{A^{n,d}, n = 1, \cdots, \mathcal{N}\}$. We let $F(x)$ denote the cumulative distribution function associated with the random variable.

For a function $\mathcal{A}^n$ on Ω, $\mathcal{A}^{n,d}$ means $\mathcal{A}^n$ restricted to Ω_d. For $d = 1, \cdots, \mathcal{D}$, d' denotes the set of indices in $\{1, 2, \cdots, \mathcal{D}\} \setminus \{d\}$ for which the corresponding domains $\Omega_{d'}$ share a common boundary with Ω_d. We let $(\cdot, \cdot)_d$ denote the $L^2(\Omega_d)$ scalar product, $\langle \cdot, \cdot \rangle_d$ denote the $L^2(\partial\Omega_d)$ scalar product where discontinuous functions are evaluated from the Ω_d side, and $\langle \cdot, \cdot \rangle_{d \cap \tilde{d}}$ denote the $L^2(\partial\Omega_d \cap \partial\Omega_{\tilde{d}})$ scalar product for $\tilde{d} \in d'$, here discontinuous functions should be evaluated from the $\Omega_{\tilde{d}}$ side.

We use the finite element method to compute numerical solutions. Let $\mathcal{T}_h = \{\tau\}$ be a quasiuniform partition into elements that $\cup\tau = \Omega$. We assume that the finite element discretization $\mathcal{T}_h$ is obtained by refinement of $\{\Omega_d\}$. This is natural when the diffusion coefficient a and the data vary on a scale finer than the partition $\{\Omega_d\}$. Associated to $\mathcal{T}_h$, we define the discrete finite element space $\mathcal{V}_h$ consisting of continuous, piecewise linear functions on $\mathcal{T}$ satisfying Dirichlet boundary conditions, with mesh size function $h_\tau = \mathrm{diam}(\tau)$ for $x \in \tau$. Since we assume quasi uniform mesh we let $h = \mathrm{mean}(\{h_\tau\})$. We equip each subdomain with a local finite element space $\mathcal{V}_{h,d}$ by restricting $\mathcal{V}_h$ to domain Ω_d. We let $m_d = \dim(\mathcal{V}_{h,d})$. We further let $H_d = \mathrm{diam}(\Omega_d)$ and assume that the domains are of similar size, i.e. we let $H = \mathrm{mean}(\{H_d\})$ be the typical domain size.

2.3. **Motivation.** Monte-Carlo simulation involving a large number of samples is very expensive in particular when each sample is costly to compute. Traditionally, one solves the differential equation inside a loop over random samples. Since the relative size of the statistical error and the numerical error is unknown a priori a very big sample size and very fine mesh size is needed to guarantee an accurate solution. It is clear that this procedure is very costly, approximately,

$$\mathcal{N} \text{ solves of linear systems with } \sum_{d=1}^{\mathcal{D}} m_d \text{ unknowns}$$

are needed. The method described in Algorithm 1 resolves these problems. First the sample loop is moved inside the differential equation solver. The goal is to do as little work inside the sample loop as possible. In particular we do not want to solve any linear systems of equations in the sample loop so all linear systems are solved outside the sample loop. Second, we derive a posteriori error estimates for the cumulative distribution function of a quantity of interest and give an adaptive algorithm for choosing all critical method parameters automatically guaranteeing both that the statistical error and the numerical error are appropriately small. This means that we can get an optimal mesh and sample size in order to reach a prescribed tolerance.

2.4. **The computational method.** We apply Lions' non-overlapping domain decomposition algorithm to (1.1). The method is iterative, so for a function $\mathcal{U}$ involved in the iteration, $\mathcal{U}_i$ denotes the value at the i$^{\text{th}}$ iteration. We let $\{U_0^{n,d}, d = 1, \cdots, \mathcal{D}\}$ and $\{G_0^{n,d}, d = 1, \cdots, \mathcal{D}\}$ denote a set of initial guesses for solutions in the subdomains. Given the initial conditions, for each $i \geq 1$, we solve the $\mathcal{D}$ problems

$$\begin{cases} -\nabla \cdot \mathcal{A}^n \nabla U_i^{n,d} = f, & x \in \Omega_d, \\ U_i^{n,d} = 0, & x \in \partial\Omega_d \cap \partial\Omega, \\ G_i^{n,d} = 2\lambda U_{i-1}^{n,\tilde{d}} - G_{i-1}^{n,\tilde{d}} & x \in \partial\Omega_d \cap \partial\Omega_{\tilde{d}}, \quad \tilde{d} \in d', \\ \lambda U_i^{n,d} + \boldsymbol{n}_d \cdot \mathcal{A}^n \nabla U_i^{n,d} = G_i^{n,d} & x \in \partial\Omega_d, \end{cases} \tag{2.3}$$

where $\lambda \in \mathbf{R}$ and $\boldsymbol{n}_d$ is the outward unit normal associated with the boundary $\partial\Omega_d$. In the convergence analysis we will give restrictions on λ, it turns out that the optimal choice is $\lambda \sim h^{-1/2}H^{-1/2}$. The formulation is equivalent to the formulation in [3] and [4] since,

$$G_i^{n,d} = 2\lambda U_{i-1}^{n,\tilde{d}} - G_{i-1}^{n,\tilde{d}} = \lambda U_{i-1}^{n,\tilde{d}} - \boldsymbol{n}_{\tilde{d}} \cdot \mathcal{A}^n \nabla U_{i-1}^{n,\tilde{d}}. \tag{2.4}$$

In practice, we compute $\mathcal{I}$ iterations. Note that the subgrid problems can be solved independently.

For each $i \geq 1$, we compute $U_i^{n,d} \in \mathcal{V}_{h,d}$, $d = 1, \cdots, \mathcal{D}$, solving

$$(\mathcal{A}^n \nabla U_i^{n,d}, \nabla v)_d + \lambda \langle U_i^{n,d}, v \rangle_d = (f, v)_d + \langle G_i^{n,d}, v \rangle_d, \text{ all } v \in \mathcal{V}_{h,d}, \tag{2.5}$$

$$G_i^{n,d} = 2\lambda U_{i-1}^{n,\tilde{d}} - G_{i-1}^{n,\tilde{d}}, \qquad x \in \partial\Omega_d \cap \partial\Omega_{\tilde{d}}, \quad \tilde{d} \in d'. \tag{2.6}$$

It is convenient to use the matrix form of (2.5-2.6) when discribing the method. We let $\{\varphi_m^d, m = 1, \cdots, m_d\}$ be the finite element basis functions for the space $\mathcal{V}_{h,d}$, $d = 1, \cdots, \mathcal{D}$.

We let $\vec{U}_i^{n,d}$ denote the vector of basis coefficients of $U_i^{n,d}$ with respect to $\{\varphi_m^d\}$. On each domain Ω_d,

$$(\mathbf{k}^{a,d} + \mathbf{k}^{n,d})\vec{U}_i^{n,d} = \vec{b}^d(f) + \vec{b}^{n,d}\big(G_i^{n,d}\big),$$

where

$$\begin{aligned}
(\mathbf{k}^{a,d})_{lk} &= (a\nabla\varphi_l^d, \nabla\varphi_k^d)_d + \lambda\langle\varphi_l^d, \varphi_k^d\rangle_d,\\
(\mathbf{k}^{n,d})_{lk} &= (A^{n,d}\nabla\varphi_l^d, \nabla\varphi_k^d)_d,\\
(\vec{b}^d)_k &= (f, \varphi_k^d)_d,\\
(\vec{b}^{n,d})_k &= \big\langle G_i^{n,d}, \varphi_k^d\big\rangle_d,
\end{aligned}$$

for $1 \le l, k \le m_d$.

Next we use that $A^{n,d}$ is constant on each Ω_d. Consequently, the matrix $\mathbf{k}^{n,d}$ has coefficients

$$(\mathbf{k}^{n,d})_{lk} = (A^{n,d}\nabla\varphi_l^d, \nabla\varphi_k^d)_d = A^{n,d}(\nabla\varphi_l^d, \nabla\varphi_k^d)_d = A^{n,d}(\mathbf{k}^d)_{lk},$$

where $\mathbf{k}^d$ is the standard stiffness matrix with coefficients $(\mathbf{k}^d)_{lk} = (\nabla\varphi_l^d, \nabla\varphi_k^d)_d$. In Lemma 3.3 we prove that the Neumann series expansion for the inverse of a perturbation of the identity matrix is valid here,

$$\big(\mathbf{k}^{a,d} + A^{n,d}\mathbf{k}^d\big)^{-1} = \sum_{p=0}^{\infty}(-A^{n,d})^p\big((\mathbf{k}^{a,d})^{-1}\mathbf{k}^d\big)^p(\mathbf{k}^{a,d})^{-1}.$$

We compute only $\mathcal{P}$ terms in the Neumann expansion to generate the approximation,

$$\vec{U}_{\mathcal{P},i}^{n,d} = \sum_{p=0}^{\mathcal{P}-1}\Big((-A^{n,d})^p((\mathbf{k}^{a,d})^{-1}\mathbf{k}^d)^p\Big)\,(\mathbf{k}^{a,d})^{-1}\big(\vec{b}^d(f) + \vec{b}^{n,d}(G_i^{n,d})\big). \tag{2.7}$$

Note that $\vec{b}^{n,d}$ is nonzero only at boundary nodes and that $\vec{b}^d(f)$ is independent of n. If $\mathcal{W}_{h,d}$ denotes the set of vectors determined by the finite element basis functions associated with the boundary nodes on Ω_d, then $\vec{b}^{n,d}$ is in the span of $\mathcal{W}_{h,d}$. We let $U_{\mathcal{P},\mathcal{I}}^{n,d}$ denote the finite element functions determined by $\vec{U}_{\mathcal{P},\mathcal{I}}^{n,d}$ for $n = 1, \cdots, \mathcal{N}$ and $d = 1, \cdots, \mathcal{D}$. We let $U_{\mathcal{P},\mathcal{I}}^n$ denote the finite element function which is equal to $U_{\mathcal{P},\mathcal{I}}^{n,d}$ on Ω_d.

We summarize as an Algorithm given in Alg. 1. Note that the number of linear systems that have to be solved in Alg. 1 is independent of $\mathcal{N}$.

Remark 2.1 If the quasi-uniform assumption on $\mathcal{V}_h$ and the partition $\{\Omega_d\}_{d=1}^{\mathcal{D}}$ is dropped, the numbers h and H will no longer be representative for the mesh size everywhere. This also means that $\lambda \sim h^{-1/2}H^{-1/2}$ needs to vary in the domain Ω. In this case we simply include λ inside the integrals and treat it as a function of space. We have made this simplification in order to make the presentation clearer.

Algorithm 1 Monte-Carlo Domain Decomposition Finite Element Method

```
for $d = 1, \cdots, D$ (number of domains) do
    for $p = 1, \cdots, \mathcal{P}$ (number of terms) do
        Compute $\vec{y}_d^p = \big((\mathbf{k}^{a,d})^{-1}\mathbf{k}^d\big)^p(\mathbf{k}^{a,d})^{-1}\vec{b}^d(f)$
        Compute $\mathbf{y}_d^p = \big((\mathbf{k}^{a,d})^{-1}\mathbf{k}^d\big)^p(\mathbf{k}^{a,d})^{-1}\mathcal{W}_{h,d}$
    end for
end for
for $i = 1, \cdots, \mathcal{I}$ (number of iterations) do
    for $d = 1, \cdots, D$ (number of domains) do
        for $p = 1, \cdots, \mathcal{P}$ (number of terms) do
            for $n = 1, \cdots, \mathcal{N}$ (number of samples) do
                Compute $\vec{U}^{n,d}_{\mathcal{P},i} = \sum_{p=0}^{\mathcal{P}-1}(-A^{n,d})^p\big(\mathbf{y}^p\vec{b}^{n,d}(G_i^{n,d}) + \vec{y}_d^p\big)$
            end for
        end for
    end for
end for
```

2.5. **Computational cost.** Since the aim of the technique is to reduce the amount of work needed to compute the samples of the solution, it is important to analyze how efficient the method is. A critical part of the method is the construction of $\mathbf{y}_d^p$ in Algorithm 1. Even though this construction is independent of number of samples it still appears to involve computing inverses of local matrices which is very expensive. Other critical parts of the method is how much work we need to do within each loop over samples and how much storage the method requires.

We start with the computations done outside the sample loop. We want to avoid computing the inverse of $\mathbf{k}^{a,d}$ since this is expensive and produces a full matrix. On the other hand, we want to exploit the fact that we have the same operator acting on $\mathcal{N}$ right hand sides, where $\mathcal{N}$ is a very large number compared to the degrees of freedom in $\mathcal{V}_{h,d}$. From papers [3, 4] we see that the error in the cumulative distribution function is typically of order $h^2 + 1/\sqrt{N}$ for errors in quantities of interest, i.e. in two spatial dimensions we e.g. get,

$$\mathcal{N} \sim h^{-4} \sim h^{-2}\sum_{d=1}^{\mathcal{D}} m_d. \tag{2.8}$$

As indicated in Algorithm 1, we solve as many linear systems as there are boundary nodes in order to get hold of $\mathbf{y}_d^p$. If we continue with the two dimensional case, the number of right hand sides $\mathcal{W}_{h,d}$ is proportional to $m_d^{1/2}$. This means that the number of systems of equations needed to be solved in order to get $\{\mathbf{y}_d^p\}_{p=0}^{\mathcal{P}-1}$ is proportional to $\mathcal{P} \cdot m_d^{1/2}$. The dimension of each problem is m_d and the number of domains is $\mathcal{D}$. The total amount of work is proportional to

$$\sum_{d=1}^{\mathcal{D}} \left(\mathcal{P} \cdot m_d^{1/2} \text{ solves of linear systems with } m_d \text{ unknowns} \right).$$

It will be clear that this amount of work is small compared to the work within the sample loop if the number in light of equation (2.8).

Inside the sample loop, we again focus on the $\mathbf{y}_d^p$ part since it clearly more expensive then the $\vec{y}_d^p$ component of the computation. We note that we only need to compute $\vec{U}_{\mathcal{P},i}^{n,d}$ at the boundary in order to update G. The work in the matrix-matrix multiplication $\mathbf{y}_d^p(G_i^{n,d})$, where only boundary terms are computed, is $m_d \cdot \mathcal{N}$ since $G_i^{n,d}$ depends on n. The total amount of work in the sample loop is then,

$$\mathcal{I} \cdot \mathcal{N} \cdot \mathcal{P} \cdot \sum_{d=1}^{\mathcal{D}} m_d,$$

Note that the number of linear systems of equations needed to solve now is independent of the number of samples $\mathcal{N}$. Still it is clear that the amount of work in the sample loop is much greater then the work needed to pre-compute the solutions to the linear systems of equations on each subdomain. As long as the size of the local problems m_d is fairly small, not solving linear systems in the sample loop, leads to a very fast way computational method.

The storage needed for the algorithm during the computation is mainly the value of $G_i^{n,d}$ at the boundary. Then the desired output quantity is evaluated in the last iteration. This means that the storage needed is proportional to the storage needed when using domain decomposition without truncated Neumann series for the original problem, with multiple diffusion coefficients. Storage is an important factor when $\mathcal{N}$ becomes large. It is therefore crucial to only evaluate quantities of interest of the solution in the end. The entire solution may be difficult to store.

2.6. **Natural extensions of the method.** Here we consider two natural extensions of the method, overlapping domain decomposition methods and piecewise polynomial perturbation. It turn out that these two extensions lead to very similar modifications of the proposed method.

2.6.1. *Overlapping domain decomposition algorithm.* There are various non-overlapping and overlapping domain decomposition algorithms, see [9, 10]. In the original derivation of the method, we used Lions' non-overlapping algorithm. The main reason for this is that if the random perturbation is piecewise constant, the domains can be chosen so that it is constant on the domains. Here we show that this assumption is not crucial for the construction of the method. It only gives a very efficient way of implementing it. If we allow the subdomains Ω_d to overlap, we end up with a similar method. If we assume piecewise constant random perturbation on the coarse mesh, the number of random variables active on domain Ω_d is the same as the number of coarse elements that intersects the domain,

$A^{n,d} = \sum_{m=1}^{\mathcal{M}} A_m^{n,d}\chi_m$, if $\mathcal{M}$, is this number of intersected coarse elements and χ_m is the indicator function for these elements. The following linear systems of equations arises,

$$\vec{U}_{\mathcal{P},i}^{n,d} = \sum_{p=0}^{\mathcal{P}-1} \left(\sum_{m=1}^{\mathcal{M}} (-A_m^{n,d})(\mathbf{k}^{a,d})^{-1}\mathbf{k}^{m,d} \right)^p (\mathbf{k}^{a,d})^{-1} \left(\vec{b}^d(f) + \vec{b}^{n,d}\big(G_i^{n,d}\big) \right), \tag{2.9}$$

where $\mathbf{k}^{m,d}$ is a weighted stiffness matrix with coefficients $(\mathbf{k}^{m,d})_{lk} = (\chi_m \nabla\varphi_l^d, \nabla\varphi_k^d)_d$. The matrix $\mathbf{k}^{a,d}$ and vector $\vec{b}^d(f) + \vec{b}^{n,d}\big(G_i^{n,d}\big)$ should be interpreted as matrix and vector that the overlapping domain decomposition algorithm gives for the local solves. Again we only compute $\mathcal{P}$ terms in the Neumann expansion to generate the approximation.

Overlapping methods can lead to faster converges but the extra cost involved with using equation (2.9) with $\mathcal{M} > 1$ needs to be considered. The convergence proof presented in Theorem 3.1 can quite easily be generalized to other domain decomposition algorithms. The assumption of geometric convergence is however crucial.

2.6.2. *Piecewise polynomial random perturbation.* So far piecewise constant perturbations have been considered. The constant perturbation is important for the truncated Neumann series idea since the randomness can be expressed as a multiplication with a single number on each subdomain. It is possible to extend this idea to piecewise polynomial perturbations which also makes it possible to consider continuous perturbations that are very useful if a is continuous. Otherwise there will be an artificial loss of regularity in the problem. It turns out that allowing piecewise polynomial perturbation leads to very similar complications as the ones discussed for overlapping domain decomposition above.

Assume that $A^{n,d} = \sum_{m=1}^{\mathcal{M}} A_m^{n,d}\phi_m^d$, where $\{\phi_m^d\}_{m=1}^{\mathcal{M}}$ is a basis for polynomials on subdomain Ω_d. On triangles $\mathcal{M} = 3$ in order to get linear functions for example. This will effect the $\mathbf{k}^{n,d}$ matrix in the following way,

$$(\mathbf{k}^{n,d})_{lk} = (A^{n,d}\nabla\varphi_l^d, \nabla\varphi_k^d)_d = \sum_{m=1}^{\mathcal{M}} A_m^{n,d}(\phi_m^d \nabla\varphi_l^d, \nabla\varphi_k^d)_d = \sum_{m=1}^{\mathcal{M}} A_m^{n,d}(\mathbf{k}^{m,d})_{lk},$$

where $\mathbf{k}^{m,d}$ is a weighted stiffness matrix with coefficients $(\mathbf{k}^{m,d})_{lk} = (\phi_m^d \nabla\varphi_l^d, \nabla\varphi_k^d)_d$.

As long as $A^{n,d}(x) \leq \delta a(x)$ the truncated Neumann series approach can be used. The proof of Lemma 3.3 can be applied directly. We get,

$$\big(\mathbf{k}^{a,d} + \sum_{m=1}^{\mathcal{M}} A^{n,d}\mathbf{k}_m^d\big)^{-1} = \sum_{p=0}^{\infty} \Big((\mathbf{k}^{a,d})^{-1} \sum_{m=1}^{\mathcal{M}} (-A_m^{n,d})\mathbf{k}_m^d \Big)^p (\mathbf{k}^{a,d})^{-1}.$$

Again we compute only $\mathcal{P}$ terms in the Neumann expansion to generate the approximation,

$$\vec{U}_{\mathcal{P},i}^{n,d} = \sum_{p=0}^{\mathcal{P}-1} \Big(\sum_{m=1}^{\mathcal{M}} (-A_m^{n,d})(\mathbf{k}^{a,d})^{-1}\mathbf{k}_m^d \Big)^p (\mathbf{k}^{a,d})^{-1}\big(\vec{b}^d(f) + \vec{b}^{n,d}(G_i^{n,d})\big). \tag{2.10}$$

The main computational cost is that more linear systems need to be solved and more matrix-matrix products are needed in the sample loop than in the piecewise constant case. By expanding the sum, one sees that $(\mathcal{M}^{\mathcal{P}} - 1)/(\mathcal{M} - 1)$ systems needs to be solved, when

$\mathcal{M} > 1$. In the piecewise constant case, we have $\mathcal{P}$ systems that need to be solved. The number of products in the sample loop increases in the same way.

2.7. **A posteriori error estimate and adaptive computation.** In general, it appears that the stochastic nature of the computed quantity of interest $Q(U)$ is quite complex. In those circumstances, it is natural to compute the cumulative distribution function $F(x) = P(Q(U) < x)$. We approximate this using the empirical distribution function $\tilde{F}_{\mathcal{N}}$ given $\mathcal{N}$ samples of the numerical approximation $Q(\tilde{U})$ of the quantity of interest. If we further assume that we have an error bound for each realization $|Q(U^n) - Q(\tilde{U}^n)| \leq \mathcal{E}^n$, e.g. derived using standard duality arguments, see Sec. 5 in [3], we have the following error estimate presented in Theorem 4.1 in [3].

Theorem 2.1. *For any* $0 < \epsilon < 1$*,*

$$|F(x) - \tilde{F}_{\mathcal{N}}(x)| \leq \left(\frac{F(x)(1-F(x))}{\mathcal{N}\epsilon}\right)^{1/2} + L \max_{n=1,\ldots,\mathcal{N}} \mathcal{E}^n + 2\left(\frac{\log(\epsilon^{-1})}{2\mathcal{N}}\right)^{1/2},$$

with probability greater then $1-\epsilon$*, where* L *is the Lipschitz constant of* F*.*

Given the bound in Theorem 2.1, it is very natural to construct an adaptive algorithm where the statistical error, depending on $\mathcal{N}$, and the numerical error, depending on $\mathcal{E}$, should be of similar size and bounded by some given tolerance. This is accomplished by iteratively solving the problem, computing the error bound and then increase $\mathcal{N}$ or decrease $\mathcal{E}$ depending on if they meet the stopping criteria.

3. Convergence of numerical method

The domain decomposition technique used in the method was introduced by P. L. Lions in [6].The convergence properties are well studied, see e.g. [5, 7, 8]. But, we have analyze the effect of the approximation we make by solving the linear system of equations on each subdomain using a truncated Neumann series, and in particular, if this destroys the convergence of the domain decomposition algorithm. It suffices to consider one sample of A^n since the same approximation technique is use on all samples simultaneously and different samples have no communication with each other. We therefore drop all superscripts n in this section and view $\mathcal{A} = a + A$ as a single realization of $\mathcal{A}$ fulfilling the assumption $|A(x)| \leq \delta a(x)$ for all $x \in \Omega$ for some given $0 < \delta < 1$.

To begin, we specify a norm used to measure distances between the reference $\{\hat{U}, \hat{G}\}$ and the approximate solution $\{U_i, G_i\}$ after i iterations. Note that we also drop the $\mathcal{P}$ subscript to make the presentation clearer. We let

$$|||\{\hat{U}, \hat{G}\} - \{U_i, G_i\}|||^2 = \sum_{d=1}^{\mathcal{D}} \|\sqrt{\mathcal{A}}\nabla(\hat{U}^d - U_i^d)\|^2_{L^2(\Omega_d)} + \sum_{d=1}^{\mathcal{D}} \|\hat{G}^d - G_i^d\|^2_{L^2(\partial\Omega_d)}.$$

By reference solution $\{\hat{U}, \hat{G}\}$, we mean the finite element solution of equation (1.1) using the space $\mathcal{V}_h$. This solution coincides with the converged solution using standard Lions' non-overlapping domain decomposition algorithm. We now introduce a notation for an operator $\hat{T}$ that performs one iteration in Lions' domain decomposition algorithm using exact solution of the linear systems of equation on the subdomains. Let $\{\hat{U}_{i+1}, \hat{G}_{i+1}\} = \hat{T}(\{\hat{U}_i, \hat{G}_i\})$. By $\hat{T}^i$, we denote repeated use of $\hat{T}$ i.e. $\{\hat{U}_i, \hat{G}_i\} = \hat{T}(\{\hat{U}_{i-1}, \hat{G}_{i-1}\}) = \cdots = \hat{T}^i(\{U_0, G_0\})$. For the reference solution, we have the following results.

Lemma 3.1. *Since Lions' domain decomposition algorithm converges, we have for the reference solution $\{\hat{U}, \hat{G}\}$,*

$$\{\hat{U}, \hat{G}\} = \hat{T}(\{\hat{U}, \hat{G}\}). \tag{3.1}$$

Furthermore it holds,

$$\|\sqrt{\mathcal{A}}\nabla\hat{U}\|_{L^2(\Omega)} \le C_{f,\alpha,\Omega}, \tag{3.2}$$

for a constant $C_{f,\alpha,\Omega} = \frac{C_{PF}}{\sqrt{\alpha}}\|f\|_{L^2(\Omega)}$ where C_{PF} is the Poincare-Friedrich constant fulfilling, $\|v\|_{L^2(\Omega)} \le C_{PF}\|\nabla v\|_{L^2(\Omega)}$ for all $v \in \mathcal{H}_0^1(\Omega)$.

Proof. See section 5.1. □

We need a crucial assumption on $\hat{T}$ to prove convergence. We first introduce a set of functions. Let,

$$\mathcal{B} = \{\{W, J\} : \text{ such that } J \in L^2(\cup_{d=1,\dots,\mathcal{D}}\partial\Omega_d) \text{ and } W \in H^1(\cup_{d=1,\dots,\mathcal{D}}\Omega_d) \text{ and} W|_{\partial\Omega} = 0\}.$$

Given $\{W, J\} \in \mathcal{B}$, with a particular relation between W and J, namely, let $W^d \in \mathcal{V}_{h,d}$, $d = 1, \cdots, \mathcal{D}$, solve

$$(\mathcal{A}\nabla W^d, \nabla v)_d + \lambda\big\langle W^d, v\big\rangle_d = (f, v)_d + \big\langle J^d, v\big\rangle_d, \text{ all } v \in \mathcal{V}_{h,d}, \tag{3.3}$$

it holds,

$$|||\{\hat{U}, \hat{G}\} - \hat{T}^i(\{W, J\})||| \le C_2 L^i |||\{\hat{U}, \hat{G}\} - \{W, J\}|||, \tag{3.4}$$

where C_2 is a given constant and $0 \le L < 1$, i.e. we assume *Geometric convergence*, see Definition 2.2 in [8]. This result is proven for the method we use in [7, 8] under various assumptions on the partition $\{\Omega_d\}_{d=1}^{\mathcal{D}}$. The optimal choice of the method parameter is $\lambda \sim h^{-1/2}H^{-1/2}$, see [8].

From [8], we get that $L = 1 - C_1(C_0)^N h^{1/2}H^{-1/2}$ where $C_1 > 0$ depends on $\mathcal{A}$, $C_0 \in (0, 1)$, and N is the winding number of the partition $\{\Omega_d\}_{d=1}^{\mathcal{D}}$. The winding number depends on the structure of the partition. Roughly speaking each subdomain is associated with a number measuring the closest path from the current subdomain to the boundary. The distance is measured in number of subdomains that need to be crossed. Furthermore, two paths of different subdomains with the same number should never cross. The winding number is the largest value one gets among the subdomains. It is clear that a large winding number leads to slow convergence. A more extensive discussion of this can be found in [7, 8]. In this paper we assume equation (3.4) to hold motivated by [7, 8].

Since we solve the linear systems on the subdomains using a truncated Neumann series, we get a perturbed solution operator T i.e. $\{U_{i+1}, G_{i+1}\} = T(\{U_i, G_i\})$. The goal is to bound $|||\{\hat{U}, \hat{G}\} - \{U_{\mathcal{I}}, G_{\mathcal{I}}\}|||$. We start by stating two more Lemmas. The proofs can be found in Section 5.

Lemma 3.2. *For arbitrary $\{W_a, J_a\}, \{W_b, J_b\} \in \mathcal{B}$ it holds,*

$$|||\hat{T}^2(\{W_a, J_a\}) - \hat{T}^2(\{W_b, J_b\})||| \leq C_\lambda |||\hat{T}(\{W_a, J_a\}) - \hat{T}(\{W_b, J_b\})|||,$$

where $C_\lambda = (1 + 1/(4\lambda))^{1/2}$.

Proof. See section 5.2. □

Lemma 3.3. *Let $\{W, J\} \in \mathcal{B}$ be an arbitrary initial guess and let $\{W_1, J_1\} = T(\{W, J\})$ be computed using $\mathcal{P}$ terms in the Neumann series. The Neumann series converges as $\mathcal{P} \to \infty$ and it holds,*

$$|||T(\{W, J\}) - \hat{T}(\{W, J\})|||^2 \leq 2\delta^{2\mathcal{P}} \sum_{d=1}^{\mathcal{D}} \|\sqrt{a}\nabla W_1^d\|^2_{L^2(\Omega_d)},$$

and furthermore,

$$|||\hat{T}(\hat{T}(\{W, J\})) - \hat{T}(T(\{W, J\}))|||^2 \leq \lambda C_\lambda^2 \delta^{2\mathcal{P}} \sum_{d=1}^{\mathcal{D}} \|\sqrt{a}\nabla W_1^d\|^2_{L^2(\Omega_d)}.$$

when $\mathcal{P}$ is large enough for $\delta^{\mathcal{P}} \leq 1/2$.

Proof. See section 5.3. □

We are now ready to present the main theorem. We use the fact that geometric convergence means that there exists a number i_0 such that the original method is a contraction after i_0 iterations, i.e. $|||\{\hat{U}, \hat{G}\} - \hat{T}^{i_0}(\{W, J\})||| \leq \gamma |||\{\hat{U}, \hat{G}\} - \{W, J\}|||$ for some $0 \leq \gamma < 1$. The error committed by using truncated Neumann series perturbs this result, but the size of the perturbation can be bounded in terms of $\delta^{\mathcal{P}}$ using Lemma 3.2 and 3.3.

Theorem 3.1. *Let $\{W, J\} \in \mathcal{B}$ be an arbitrary initial guess and let $\{\hat{U}, \hat{G}\}$ be the reference solution. For a fixed integer $i_0 \geq 1$ fulfilling $C_2 L^{i_0} < 1/4$ and $\mathcal{P}$ large enough so that $C_{i_0,\lambda} C_2 L \delta^{\mathcal{P}} < 1/4$ it holds,*

$$|||\{\hat{U}, \hat{G}\} - \{U_{ki_0}, G_{ki_0}\}||| \leq \gamma^k |||\{\hat{U}, \hat{G}\} - \{W, J\}||| + 2C_{i_0,\lambda} C_{f,\alpha,\Omega} \delta^{\mathcal{P}}, \tag{3.5}$$

where $\gamma < 1/2$ and $C_{i_0,\lambda} = 12(\lambda^{1/2} C_\lambda + 4)(C_\lambda^{i_0} - 1)/[(1 - \delta)(C_\lambda - 1)]$ and k is any positive integer.

Proof. See section 5.4. □

So far, we have compared the approximate solution using the domain decomposition together with the truncated Neumann series to a direct solve using the finite element method on the fine mesh using piecewise linear basis functions. If we let $\{U, G\}$ be the exact solution to (1.1) and assume we have,

$$|||\{U, G\} - \{\hat{U}, \hat{G}\}||| \leq C_{\text{ex}} h^\alpha, \tag{3.6}$$

for some $\alpha > 0$ and $C_{\text{ex}} > 0$ that depends on $\mathcal{A}$, f, Ω, and an interpolation constant. Given this a priori estimate we get the following result.

Corollary 3.1. *Let $\{W, J\} \in \mathcal{B}$ be an arbitrary initial guess and let $\{U, G\}$ be the exact solution to (1.1). Then it holds,*

$$\lim_{k\to\infty} |||\{U, G\} - \{U_{ki_0}, G_{ki_0}\}||| \leq C_{ex}h^{\alpha} + 2C_{i_0,\lambda}C_{f,\alpha,\Omega}\delta^{\mathcal{P}}, \tag{3.7}$$

for some fix integer $i_0 \geq 1$.

Proof. The corollary follows immediately by combining (3.6) and (3.5), using the triangle inequality, and taking the limit. □

There are various constants in the error analysis. We discuss them in two remarks below.

Remark 3.1 First of all, since $\lambda \sim h^{-1/2}H^{-1/2}$ the constant $C_\lambda > 1$ is directly computable and very close to 1. This means that

$$\frac{C_\lambda^{i_0} - 1}{C_\lambda - 1} \approx i_0.$$

The number of iterations i_0 needed in order to reduce the error to a quarter in Lions' method i.e. $C_2L^{i_0} < 1/4$ is very problem dependent but is in most cases a number of moderate size. The constants C_2 and L can be approximated by measuring distance between iterates in the algorithm. The constants $C_{f,\alpha,\Omega} = C_{\mathrm{PF}}\|f\|_{L^2(\Omega)}/\sqrt{\alpha}$ and $C_{i_0,\lambda}$ are directly computable. Furthermore, $C_{\mathrm{ex}} \approx C_{\mathrm{int}}\|f\|_{L^2(\Omega)}/\sqrt{\alpha}$ where C_{int} is an interpolation constant associated with $\mathcal{V}_h$.

Remark 3.2 We note that in order to equidistribute the error between the two error contributions in (3.7) the number of terms in the truncated Neumann series should roughly be chosen so that the following holds,

$$\delta^{\mathcal{P}} \approx \frac{C_{\mathrm{ex}}h^{\alpha}}{2C_{i_0,\lambda}C_{f,\alpha,\Omega}}.$$

All these constants are relatively easy to compute. The restriction on $\mathcal{P}$ for which Theorem 3.1 is valid boils down to,

$$\delta^{\mathcal{P}} \leq \frac{1}{4C_2LC_{i_0,\lambda}},$$

where C_2, L, and i_0 depends on the underlying domain decomposition algorithm, and can be approximated by comparing errors between iterates, and $C_{i_0,\lambda}$ is directly computable.

4. Coarse grid correction

Non-overlapping domain decomposition algorithms tend to have slow convergence when the number of subdomains increases. To overcome this issue, coarse grid correction is often used. Here we briefly describe how coarse grid correction can be used within the proposed framework.

Given a previous iterate $\{U_i^{n,d}, G_i^{n,d}\}_{d=1}^{\mathcal{D}}$, the new iterate is computed as a sum of the old iterate, a coarse grid correction, and a fine grid solution that will be computed using the same method described above, i.e.

$$G_{i+1}^{n,d} = 2\lambda U_i^{n,\tilde{d}} - G_i^{n,\tilde{d}}, \tag{4.1}$$

$$U_{i+1}^{n} = U_i^n + C_{i+1}^n + \sum_{d=1}^{\mathcal{D}} F_{i+1}^n, \tag{4.2}$$

where C_{i+1}^n in the coarse space associated with the mesh given by $\{\Omega_d\}_{d=1}^{\mathcal{D}}$ solves,

(4.3)

$$\begin{aligned}\sum_{d=1}^{\mathcal{D}}(\mathcal{A}^n\nabla C_{i+1}^n\nabla v)_d + \lambda\langle C_{i+1}^n, v\rangle_d &= \sum_{d=1}^{\mathcal{D}}(f,v)_d + \langle G_{i+1}^n, v\rangle_d - (\mathcal{A}^n\nabla U_i^n\nabla v)_d + \lambda\langle U_i^n, v\rangle_d \\ &= \sum_{d=1}^{\mathcal{D}}\langle G_{i+1}^n - G_i^n, v\rangle_d\end{aligned}$$

for all coarse test functions v, and $F_{i+1}^{n,d} \in \mathcal{V}_{h,d}$ solves,

$$(\mathcal{A}^n\nabla F_{i+1}^{n,d}\nabla v)_d + \lambda\langle F_{i+1}^{n,d}, v\rangle_d = \langle G_{i+1}^n - G_i^n, v\rangle_d - (\mathcal{A}^n\nabla C_{i+1}^n\nabla v)_d - \lambda\langle C_{i+1}^n, v\rangle_d \tag{4.4}$$

for all $v \in \mathcal{V}_{h,d}$ and all $d = 1, \ldots, \mathcal{D}$. For simplicity we have assumed that $\{\Omega_d\}_{d=1}^{\mathcal{D}}$ serves as a finite element mesh. If this is not the case we just need to divide the subsdomains Ω_d into appropriate elements.

The coarse grid correction equation (4.3) is solved using brute force for each sample, which is alright since the degrees of freedom is fairly small. The local fine grid problems (4.4) are then solved using the proposed method with truncated Neumann series. The storage and computational cost, in each iteration, are similar to solving the original problem with truncated Neumann series. This procedure will speed up the convergence of the method when the number of subdomains is large.

5. Proofs of the theoretical results

In this section, we collect the proofs of the three Lemmas and the main Theorem. In Lemma 3.1 we prove basic results for the converged discrete reference solution. In Lemma 3.2, we prove a version of Lipschitz continuity for the exact map $\hat{T}$. Similar results can be found in e.g. [5]. Lemma 3.3, is an extension of a result presented in [3] to a different setting. Finally, Theorem 3.5 is the main result of the paper.

5.1. **Proof of Lemma 3.1.**

Proof. The converged solution solves: find $\hat{U} \in \mathcal{V}_h$ such that, $(\mathcal{A}\nabla\hat{U}, \nabla v) = (f, v)$ for all $v \in \mathcal{V}_h$. We can pick $v = \hat{U}$ to get $\|\sqrt{\mathcal{A}}\nabla\hat{U}\|_{L^2(\Omega)}^2 \leq \|f\|_{L^2(\Omega)}\|\hat{U}\|_{L^2(\Omega)} \leq \frac{C_{\mathrm{PF}}}{\sqrt{\alpha}}\|f\|_{L^2(\Omega)}\|\sqrt{\mathcal{A}}\nabla\hat{U}\|_{L^2(\Omega)}$. The Lemma follows by dividing with $\|\sqrt{\mathcal{A}}\nabla\hat{U}\|_{L^2(\Omega)}$. If $\nabla\hat{U} = 0$ it holds trivially. □

5.2. **Proof of Lemma 3.2.**

Proof. We let $\{\hat{W}_{j,a}, \hat{J}_{j,a}\} = \hat{T}^j(\{W_a, J_a\})$ and $\{\hat{W}_{j,b}, \hat{J}_{j,b}\} = \hat{T}^j(\{W_b, J_b\})$, for $j = 1, 2$. Applying equation (2.5) with $U_i^{n,d}$ first equal to $\hat{W}_{1,a}^d$ then $\hat{W}_{1,b}^d$ and $v = \hat{W}_{1,a}^d - \hat{W}_{1,b}^d$ for both equations, and subtraction yields,

$$\sum_{d=1}^{\mathcal{D}} \|\sqrt{\mathcal{A}}\nabla(\hat{W}_{1,a}^d - \hat{W}_{1,b}^d)\|_{L^2(\Omega_d)}^2 + \lambda\|\hat{W}_{1,a}^d - \hat{W}_{1,b}^d\|_{L^2(\partial\Omega_d)}^2 = \sum_{d=1}^{\mathcal{D}} \langle \hat{J}_{1,a}^d - \hat{J}_{1,b}^d, \hat{W}_{1,a}^d - \hat{W}_{1,b}^d\rangle_d. \tag{5.1}$$

Furthermore, we can use equation (2.3) to get,

$$\sum_{d=1}^{\mathcal{D}} \|\hat{J}_{2,a}^d - \hat{J}_{2,b}^d\|_{L^2(\partial\Omega_d)}^2 = \sum_{d=1}^{\mathcal{D}} \|2\lambda(\hat{W}_{1,a}^{\tilde{d}} - \hat{W}_{1,b}^{\tilde{d}}) - (\hat{J}_{1,a}^{\tilde{d}} - \hat{J}_{1,b}^{\tilde{d}})\|_{L^2(\partial\Omega_d)}^2 \tag{5.2}$$

$$\begin{aligned} &= \sum_{d=1}^{\mathcal{D}} \|\hat{J}_{1,a}^{\tilde{d}} - \hat{J}_{1,b}^{\tilde{d}}\|_{L^2(\partial\Omega_d)}^2 + 4\lambda^2 \sum_{d=1}^{\mathcal{D}} \|\hat{W}_{1,a}^{\tilde{d}} - \hat{W}_{1,b}^{\tilde{d}}\|_{L^2(\partial\Omega_d)}^2 \\ &\qquad - 4\lambda \sum_{d=1}^{\mathcal{D}} \langle \hat{J}_{1,a}^{\tilde{d}} - \hat{J}_{1,b}^{\tilde{d}}, \hat{W}_{1,a}^d - \hat{W}_{1,b}^d\rangle_d \end{aligned} \tag{5.3}$$

$$= \sum_{d=1}^{\mathcal{D}} \|\hat{J}_{1,a}^{\tilde{d}} - \hat{J}_{1,b}^{\tilde{d}}\|_{L^2(\partial\Omega_d)}^2 - 4\lambda \sum_{d=1}^{\mathcal{D}} \|\sqrt{\mathcal{A}}\nabla(\hat{W}_{1,a}^d - \hat{W}_{1,b}^d)\|_{L^2(\Omega_d)}^2$$

$$\le \sum_{d=1}^{\mathcal{D}} \|\hat{J}_{1,a}^{\tilde{d}} - \hat{J}_{1,b}^{\tilde{d}}\|_{L^2(\partial\Omega_d)}^2, \tag{5.4}$$

$$\le \sum_{d=1}^{\mathcal{D}} \|\hat{J}_{1,a}^d - \hat{J}_{1,b}^d\|_{L^2(\partial\Omega_d)}^2,$$

using equation (5.1) and changing the order of the sums in the last step. Each interior edge will have exactly two elements associated with it, d and $\tilde{d}$. We let $v = \hat{W}_{2,a}^d - \hat{W}_{2,b}^d$ and $U_i^{n,d}$ to be first $\hat{W}_{2,a}^d$ and then $\hat{W}_{2,b}^d$ in equation (2.5) and then subtract to get,

$$\sum_{d=1}^{\mathcal{D}} \|\sqrt{\mathcal{A}}\nabla(\hat{W}_{2,a}^d - \hat{W}_{2,b}^d)\|_{L^2(\Omega_d)}^2 + \lambda \sum_{d=1}^{\mathcal{D}} \|(\hat{W}_{2,a}^d - \hat{W}_{2,b}^d)\|_{L^2(\partial\Omega_d)}^2 = \sum_{d=1}^{\mathcal{D}} \langle \hat{J}_{2,a}^d - \hat{J}_{2,b}^d, \hat{W}_{2,a}^d - \hat{W}_{2,b}^d\rangle_d \tag{5.5}$$

$$\le \lambda \sum_{d=1}^{\mathcal{D}} \|(\hat{W}_{2,a}^d - \hat{W}_{2,b}^d)\|_{L^2(\partial\Omega_d)}^2 + \frac{1}{4\lambda} \sum_{d=1}^{\mathcal{D}} \|(\hat{J}_{2,a}^d - \hat{J}_{2,b}^d)\|_{L^2(\partial\Omega_d)}^2, \tag{5.6}$$

since $\epsilon > 0$ and,

$$|ab| \le \frac{\epsilon}{2} b^2 + \frac{a^2}{2\epsilon}, \quad \text{for all } a, b \in \mathbf{R}, \tag{5.7}$$

and ϵ can be chosen to be equal to 2λ. We conclude,

$$|||\{\hat{W}_{2,a}, \hat{J}_{2,a}\}-\{\hat{W}_{2,b}, \hat{J}_{2,b}\}||| \leq C_\lambda \left(\sum_{d=1}^{\mathcal{D}} \|\hat{J}^{\tilde{d}}_{1,a} - \hat{J}^{\tilde{d}}_{1,b}\|^2_{L^2(\partial\Omega_d)} \right)^{1/2} \leq C_\lambda |||\{\hat{W}_{1,a}, \hat{J}_{1,a}\}-\{\hat{W}_{1,b}, \hat{J}_{1,b}\}|||,$$

where $C_\lambda = \left(1 + \frac{1}{4\lambda}\right)^{1/2}$. Again we change the order in the sum. □

5.3. **Proof of Lemma 3.3.**

Proof. We consider one particular subdomain Ω_d. Let the matrix $\mathbf{m} = -(A^{n,d}\mathbf{k}^{a,d})^{-1}\mathbf{k}^d$. First we want to show that $(1-\mathbf{m})^{-1} = \sum_{p=0}^{\infty} \mathbf{m}^p$, i.e. the Neumann series converges. We want to study $\mathbf{m} : \mathbf{R}^{n_d} \to \mathbf{R}^{n_d}$. Let $m : \mathcal{V}_{h,d} \to \mathcal{V}_{h,d}$ be the corresponding map in the finite element space. Let $z = mx$ for an arbitrary $v \in \mathcal{V}_{h,d}$. Then we have,

$$(a\nabla z, \nabla v)_d + \lambda \langle z, v\rangle_d = -(A^{n,d}\nabla x, \nabla v), \quad \text{for all } v \in \mathcal{V}_{h,d}.$$

This means in particular that,

$$\|\sqrt{a}\nabla z\|^2_{L^2(\Omega_d)} + \lambda\|z\|^2_{L^2(\partial\Omega_d)} \leq |\delta(a\nabla x, \nabla z)| \leq \delta\|\sqrt{a}\nabla x\|_{L^2(\Omega_d)}\|\sqrt{a}\nabla z\|_{L^2(\Omega_d)}.$$

We use equation (5.7) to get,

$$\|z\|_d^2 := \|\sqrt{a}\nabla z\|^2_{L^2(\Omega_d)} + 2\lambda\|z\|^2_{L^2(\partial\Omega_d)} \leq \delta^2\|\sqrt{a}\nabla x\|^2_{L^2(\Omega_d)},$$

where we also define a short notation for this norm. In this norm we have, $\|mx\|_d \leq \delta\|\sqrt{a}\nabla x\|_{L^2(\Omega_d)}$. Furthermore, we can repeat the argument for mz and $m^p z$ to get, $\|m^p x\|_d \leq \delta^p\|\sqrt{a}\nabla x\|_{L^2(\Omega_d)}$. We know that $\mathrm{id} - \mathbf{m}^{\mathcal{P}} = (\mathrm{id} - \mathbf{m})\sum_{p=0}^{\mathcal{P}-1}\mathbf{m}^p$, where id is the identity operator. We take limit on both sides. Note that $\mathbf{m}^{\mathcal{P}} \to 0$ as $\mathcal{P} \to \infty$, since $\lim_{\mathcal{P}\to\infty}\sup_{\|x\|_d=1}\|m^p x\|_d = 0$. We get, $\mathrm{id} = (\mathrm{id} - \mathbf{m})\sum_{p=0}^{\infty}\mathbf{m}^p$, which proves that the Neumann series converges.

Furthermore,

$$\|((1-m)^{-1} - \sum_{p=0}^{\mathcal{P}-1} m^p)x\|_d = \|m^{\mathcal{P}}(1-m)^{-1}x\|_d \leq \delta^{\mathcal{P}}\|\sqrt{a}\nabla((1-m)^{-1}x)\|_{L^2(\Omega_d)}$$

If we apply this to all subdomains Ω_d with x as the finite element function corresponding to $(\mathbf{k}^{a,d})^{-1}(\vec{b}^d(f)+\vec{b}^d(J^d))$ on each domain, see Algorithm (1), and let, $\{W_1, J_1\} = T(\{W, J\}))$ and $\{\hat{W}_1, \hat{J}_1\} = \hat{T}(\{W, J\})$ we get,

$$\sum_{d=1}^{\mathcal{D}} \left(\|\sqrt{a}\nabla(W_1^d - \hat{W}_1^d)\|^2_{L^2(\Omega_d)} + 2\lambda\|W_1^d - \hat{W}_1^d\|^2_{L^2(\partial\Omega_d)} \right) \leq \sum_{d=1}^{\mathcal{D}} \delta^{2\mathcal{P}}\|\sqrt{a}\nabla\hat{W}_1^d\|^2_{L^2(\Omega_d)}.$$

One can easily replace $\hat{W}_1^d$ with W_1^d in the right hand side using a simple kick back argument. Assuming $\delta^{\mathcal{P}} < 1/2$ yields,

(5.8)
$$\sum_{d=1}^{\mathcal{D}} \left(\|\sqrt{a}\nabla(W_1^d - \hat{W}_1^d)\|^2_{L^2(\Omega_d)} + 4\lambda\|W_1^d - \hat{W}_1^d\|^2_{L^2(\partial\Omega_d)} \right) \leq 2\sum_{d=1}^{\mathcal{D}} \delta^{2\mathcal{P}}\|\sqrt{a}\nabla W_1^d\|^2_{L^2(\Omega_d)}.$$

Since $\hat{J}_1 = J_1$ we immediately get the first result,

$$|||W_1 - \hat{W}_1|||^2 \leq 4 \sum_{d=1}^{\mathcal{D}} \delta^{2\mathcal{P}} \|\sqrt{a}\nabla W_1^d\|^2_{L^2(\Omega_d)},$$

where we use $|a/\mathcal{A}| = 1/(1 + A/a) \geq 1/(1+\delta) \geq 1/2$. Next we let $\{\hat{W}_2, \hat{J}_2\} = \hat{T}(\{\hat{W}_1, \hat{J}_1\})$ and $\{\tilde{W}_2, \tilde{J}_2\} = \hat{T}(\{W_1, J_1\})$. We use equation (5.3,5.5) and (5.8) and that $\hat{J}_1 = J_1$ to get,

$$\begin{aligned}
|||\{\hat{W}_2, \hat{J}_2\} - \{\tilde{W}_2, \tilde{J}_2\}||| &= \sum_{d=1}^{\mathcal{D}} \Big(\|\sqrt{\mathcal{A}}\nabla(\hat{W}_2^d - \tilde{W}_2^d)\|^2_{L^2(\Omega_d)} + \|\hat{J}_2^d - \tilde{J}_2^d\|^2_{L^2(\partial\Omega_d)} \Big) \\
&\leq C_\lambda^2 \sum_{d=1}^{\mathcal{D}} \|\hat{J}_2^d - \tilde{J}_2^d\|^2_{L^2(\partial\Omega_d)} \\
&\leq 4\lambda^2 C_\lambda^2 \sum_{d=1}^{\mathcal{D}} \|\hat{W}_1^d - W_1^d\|^2_{L^2(\partial\Omega_d)} \\
&\leq \lambda C_\lambda^2 \delta^{2\mathcal{P}} \sum_{d=1}^{\mathcal{D}} \|\sqrt{a}\nabla W_1^d\|^2_{L^2(\Omega_d)}.
\end{aligned}$$

The Lemma follows immediately. □

5.4. **Proof of Theorem 3.5.**

Proof. We first consider the difference between $\mathcal{I}$ iterations of the exact domain decomposition algorithm $\{\hat{U}_\mathcal{I}, \hat{G}_\mathcal{I}\} = \hat{T}^\mathcal{I}(\{W, J\})$ and $\mathcal{I}$ iterations of the algorithm using truncated Neumann series $\{U_\mathcal{I}, G_\mathcal{I}\} = T^\mathcal{I}(\{W, J\})$. We have,

$$\begin{aligned}
&|||\{\hat{U}_\mathcal{I}, \hat{G}_\mathcal{I}\} - \{U_\mathcal{I}, G_\mathcal{I}\}||| \\
&\quad \leq |||\{\hat{U}_\mathcal{I}, \hat{G}_\mathcal{I}\} - \hat{T}(\{U_{\mathcal{I}-1}, G_{\mathcal{I}-1}\})||| + |||\hat{T}(\{U_{\mathcal{I}-1}, G_{\mathcal{I}-1}\}) - T(\{U_{\mathcal{I}-1}, G_{\mathcal{I}-1})\}||| \\
&\quad \leq |||\{\hat{U}_\mathcal{I}, \hat{G}_\mathcal{I}\} - \hat{T}^2(\{U_{\mathcal{I}-2}, G_{\mathcal{I}-2}\})||| + |||\hat{T}^2(\{U_{\mathcal{I}-2}, G_{\mathcal{I}-2}\}) - \hat{T}(\{U_{\mathcal{I}-1}, G_{\mathcal{I}-1}\})||| \\
&\qquad + |||\hat{T}(\{U_{\mathcal{I}-1}, G_{\mathcal{I}-1}\}) - T(\{U_{\mathcal{I}-1}, G_{\mathcal{I}-1})\}||| \\
&\quad \leq C_\lambda |||\{\hat{U}_{\mathcal{I}-1}, \hat{G}_{\mathcal{I}-1}\} - \hat{T}(\{U_{\mathcal{I}-2}, G_{\mathcal{I}-2}\})||| + C_\lambda \delta^\mathcal{P} \left(\lambda \sum_{d=1}^{\mathcal{D}} \|\sqrt{a}\nabla U_{\mathcal{I}-1}^d\|^2_{L^2(\Omega_d)} \right)^{1/2} \\
&\qquad + |||\hat{T}(\{U_{\mathcal{I}-1}, G_{\mathcal{I}-1}\}) - T(\{U_{\mathcal{I}-1}, G_{\mathcal{I}-1})\}|||
\end{aligned}$$

(5.9)

$$\begin{aligned}
&\quad \leq C_\lambda |||\{\hat{U}_{\mathcal{I}-1}, \hat{G}_{\mathcal{I}-1}\} - \{U_{\mathcal{I}-1}, G_{\mathcal{I}-1}\}||| + \big(C_\lambda \lambda^{1/2} + 2\big)\delta^\mathcal{P} \left(\sum_{d=1}^{\mathcal{D}} \|\sqrt{a}\nabla U_{\mathcal{I}-1}^d\|^2_{L^2(\Omega_d)} \right)^{1/2} \\
&\qquad + 2\delta^\mathcal{P} \left(\sum_{d=1}^{\mathcal{D}} \|\sqrt{a}\nabla U_\mathcal{I}^d\|^2_{L^2(\Omega_d)} \right)^{1/2}
\end{aligned}$$

using Lemma 3.2 and 3.3. We use equation (5.9) for all $\mathcal{I}$ down to 0 and let $C^\lambda = 2(\lambda^{1/2}C_\lambda + 4)$ and note that $a(1-\delta) \leq \mathcal{A} \leq a(1+\delta)$ by assumption. We conclude,

$$|||\{\hat{U}_\mathcal{I}, \hat{G}_\mathcal{I}\} - \{U_\mathcal{I}, G_\mathcal{I}\}||| \leq C^\lambda \delta^\mathcal{P} \sum_{i=0}^{\mathcal{I}} C_\lambda^i \left(\sum_{d=1}^{\mathcal{D}} \|\sqrt{a}\nabla U_i^d\|^2_{L^2(\Omega_d)} \right)^{1/2}. \tag{5.10}$$

We now add and subtract appropriate terms to get,

(5.11)

$$|||\{\hat{U}_\mathcal{I}, \hat{G}_\mathcal{I}\} - \{U_\mathcal{I}, G_\mathcal{I}\}|||$$

(5.12)

$$\leq \frac{C^\lambda \delta^\mathcal{P}}{1-\delta} \sum_{i=0}^{\mathcal{I}} C_\lambda^i \left(\sum_{d=1}^{\mathcal{D}} \|\sqrt{\mathcal{A}}\big(\nabla U_i^d - \nabla \hat{U}_i^d + \nabla \hat{U}_i^d - \nabla \hat{U} + \nabla \hat{U}\big)\|^2_{L^2(\Omega_d)} \right)^{1/2}$$

$$\leq \frac{3C^\lambda \delta^\mathcal{P}}{1-\delta} \sum_{i=0}^{\mathcal{I}} C_\lambda^i \big(|||\{U_i, G_i\} - \{\hat{U}_i, \hat{G}_i\}||| + |||\{\hat{U}_i, \hat{G}_i\} - \{\hat{U}, \hat{G}\}||| + \|\sqrt{A}\nabla \hat{U}\|_{L^2(\Omega)} \big)$$

(5.13)

$$\leq \frac{3C^\lambda \delta^\mathcal{P}}{1-\delta} \sum_{i=0}^{\mathcal{I}} C_\lambda^i \big(|||\{U_i, G_i\} - \{\hat{U}_i, \hat{G}_i\}||| + C_2 L^i |||\{W, J\} - \{\hat{U}, \hat{G}\}||| + C_{f,\alpha,\Omega} \big)$$

Let $j^* \in 1, \dots, \mathcal{I}$ be the index for the largest error,

$$\max_{i=1,\dots,\mathcal{I}} |||\{\hat{U}_i, \hat{G}_i\} - \{U_i, G_i\})||| = |||\{\hat{U}_{j^*}, \hat{G}_{j^*}\} - \{U_{j^*}, G_{j^*}\})|||.$$

We apply equations (5.11-5.13) with $\mathcal{I} = j^*$ to get,

$$\begin{aligned} |||\{\hat{U}_{j^*}, \hat{G}_{j^*}\} - \{U_{j^*}, G_{j^*}\})||| &\leq \frac{3C^\lambda \delta^\mathcal{P}}{1-\delta} \sum_{i=0}^{j^*} C_\lambda^i |||\{U_i, G_i\} - \{\hat{U}_i, \hat{G}_i\}||| \\ &\quad + \frac{3C^\lambda \delta^\mathcal{P}}{1-\delta} \frac{C_\lambda^{j^*} - 1}{C_\lambda - 1} \big(C_2 L |||\{W, J\} - \{\hat{U}, \hat{G}\}||| + C_{f,\alpha,\Omega} \big) \\ &\leq \frac{3C^\lambda \delta^\mathcal{P}}{1-\delta} \frac{C_\lambda^{j^*} - 1}{C_\lambda - 1} |||\{U_{j^*}, G_{j^*}\} - \{\hat{U}_{j^*}, \hat{G}_{j^*}\}||| \\ &\quad + \frac{3C^\lambda \delta^\mathcal{P}}{1-\delta} \frac{C_\lambda^{j^*} - 1}{C_\lambda - 1} \big(C_2 L |||\{W, J\} - \{\hat{U}, \hat{G}\}||| + C_{f,\alpha,\Omega} \big) \end{aligned} \tag{5.14}$$

We now assume $\mathcal{P}$ be large enough so that $\delta^\mathcal{P} \leq C_{i_0,\lambda}^{-1} = (C_\lambda - 1)(1-\delta)/(6C^\lambda(C_\lambda^{i_0} - 1))$ (where $i_0 = \mathcal{I}$, see below). We note that $C_{i_0,\lambda}$ is easily computable, see the Remarks in the end of Section 3. We conclude,

$$\begin{aligned} |||\{\hat{U}_\mathcal{I}, \hat{G}_\mathcal{I}\} - \{U_\mathcal{I}, G_\mathcal{I}\})||| &\leq |||\{\hat{U}_{j^*}, \hat{G}_{j^*}\} - \{U_{j^*}, G_{j^*}\})||| \\ &\leq C_{i_0,\lambda} \delta^\mathcal{P} \big(C_2 L |||\{W, J\} - \{\hat{U}, \hat{G}\}||| + C_{f,\alpha,\Omega} \big) \end{aligned} \tag{5.15}$$

We are now ready to bound the the error after $\mathcal{I}$ iterations. We have,

$$\begin{aligned} |||\{\hat{U},\hat{G}\}-\{U_{\mathcal{I}},G_{\mathcal{I}}\}||| &\leq |||\{\hat{U},\hat{G}\}-\{\hat{U}_{\mathcal{I}},\hat{G}_{\mathcal{I}}\})|||+|||\{\hat{U}_{\mathcal{I}},\hat{G}_{\mathcal{I}}\}-\{U_{\mathcal{I}},G_{\mathcal{I}}\}||| \\ &\leq (C_2L^{\mathcal{I}}+C_{i_0,\lambda}C_2L\delta^{\mathcal{P}})|||\{\hat{U},\hat{G}\}-\{W,J\}|||+C_{i_0,\lambda}C_{f,\alpha,\Omega}\delta^{\mathcal{P}} \end{aligned}$$

We now fix $\mathcal{I}=i_0$ to be the smallest number so that $C_2L^{i_0}<1/4$ and choose $\mathcal{P}$ so that $C_{i_0,\lambda}C_2L\delta^{\mathcal{P}}<1/4$ and let $\gamma=(C_2L^{\mathcal{I}}+C_{i_0,\lambda}C_2L\delta^{\mathcal{P}})<1/2$. This means that,

$$|||\{\hat{U},\hat{G}\}-\{U_{i_0},G_{i_0}\}|||\leq\gamma|||\{\hat{U},\hat{G}\}-\{W,J\}|||+C_{i_0,\lambda}C_{f,\alpha,\Omega}\delta^{\mathcal{P}}.$$

For an arbitrary integer $k\geq 1$ we have,

$$\begin{aligned} |||\{\hat{U},\hat{G}\}-\{U_{ki_0},G_{ki_0}\}||| &\leq \gamma|||\{\hat{U},\hat{G}\}-\{U_{(k-1)i_0},G_{(k-1)i_0}\}|||+C_{i_0,\lambda}C_{f,\alpha,\Omega}\delta^{\mathcal{P}} \\ &\leq \gamma^k|||\{\hat{U},\hat{G}\}-\{W,J\}|||+\sum_{j=0}^{k-1}\gamma^jC_{i_0,\lambda}C_{f,\alpha,\Omega}\delta^{\mathcal{P}} \\ &\leq \gamma^k|||\{\hat{U},\hat{G}\}-\{W,J\}|||+2C_{i_0,\lambda}C_{f,\alpha,\Omega}\delta^{\mathcal{P}} \end{aligned}$$

which proves the theorem. □

References

[1] R. A. Adams, *Sobolev spaces,* volume 65 of Pure and Applied Mathematics, Academic Press, New York, 1975.

[2] T. C. Rebolla and E. C. Vera, *Study of a non-overlapping domain decomposition method: Poisson and Stoke problems,* Apl. Numer. Math., 48 (2004) 169–194.

[3] D. Estep, A. Målqvist, and S. Tavener, *Nonparametric density estimation for elliptic problems with random perturbations I: computational method, a posteriori analysis, and adaptive error control,* SIAM J. Sci. Comp., 31, (2009), 2935–2959.

[4] D. Estep, A. Målqvist, and S. Tavener, *Nonparametric density estimation for elliptic problems with random perturbations II: applications and adaptive modeling,* Int. J. Numer. Methods Engrg. 80 (2009) 846-867.

[5] W. Guo and L. S. Hou, *Generalizations and accelerations of Lions' nonoverlapping domain decomposition method for linear elliptic PDE,* SIAM J. Numer. Anal. 41, (2003), no. 6, 2056-2080.

[6] P. L. Lions, *On the Schwarz alternating methods III: a variant for nonoverlapping subdomains,* in Third International Symposium on Domain Decomposition Methods for Partial Differential Equations, T. F. Chan, R. Glowinski, J. Periaux, and O. B. Wildlund, eds., SIAM, Philadelfia, PA, (1990), 202-231.

[7] L. Qin, Z. Shi, and X. Xu, *On the convergence rate of a parallel nonoverlapping domain decomposition method,* Science in China Series A: Mathematics Aug., 2008, Vol. 51, No. 8, 1461-1478

[8] L. Qin and X. Xu, *On a parallel robin-type nonoverlapping domain decomposition method,* SIAM J. Numer. Anal., Vol. 44, No. 6, (2008), 2539–2558.

[9] B. Smith, P. Bjørstad, and W. Gropp, *Domain decomposition Parallel Multilevel Methods for Elliptic Partial Differential Equations,* Cambridge University Press 1996.

[10] J. Xu and J. Zou, *Some nonoverlapping domain decomposition methods,* SIAM Rev., Vol. 40, No. 4, (1998), 857–914.

Department of Mathematics, Colorado State University
E-mail address: estep@math.colostate.edu
URL: http://www.math.colostate.edu/~estep

Department of Mathematics, University of California, San Diego
E-mail address: mholst@cam.ucsd.edu
URL: http://www.cam.ucsd.edu/~mholst

Department of Information Technology, Uppsala University
E-mail address: axel.malqvist@it.uu.se
URL: http://user.it.uu.se/~axelm