

Efficient White Noise Sampling and Coupling for Multilevel Monte Carlo with Nonnested Meshes*

M. Croci[†], M. B. Giles[†], M. E. Rognes[‡], and P. E. Farrell[†]

Abstract. When solving stochastic partial differential equations (SPDEs) driven by additive spatial white noise, the efficient sampling of white noise realizations can be challenging. Here, we present a new sampling technique that can be used to efficiently compute white noise samples in a finite element method (FEM) and multilevel Monte Carlo (MLMC) setting. The key idea is to exploit the finite element matrix assembly procedure and factorize each local mass matrix independently, hence avoiding the factorization of a large matrix. Moreover, in an MLMC framework, the white noise samples must be coupled between subsequent levels. We show how our technique can be used to enforce this coupling even in the case of nonnested mesh hierarchies. We demonstrate the efficacy of our method with numerical experiments. We observe optimal convergence rates for the finite element solution of the elliptic SPDEs of interest in 2D and 3D and we show convergence of the sampled field covariances. In an MLMC setting, a good coupling is enforced and the telescoping sum is respected.

Key words. multilevel Monte Carlo, white noise, nonnested meshes, Matérn Gaussian fields, finite elements, partial differential equations with random coefficients

AMS subject classifications. 65C05, 60G60, 65N30, 60H35, 35R60

DOI. 10.1137/18M1175239

1. Introduction. Gaussian fields are ubiquitous in uncertainty quantification to model the uncertainty in spatially dependent parameters. Common applications are in geology, oil reservoir modeling, biology, and meteorology [6, 23, 27, 33]. Here, let $D \subset \mathbb{R}^d$ be an open spatial domain of interest whose closure is a compact subset of \mathbb{R}^d . Consider the task of sampling from a zero-mean Gaussian field u of Matérn covariance \mathcal{C} ,

$$(1.1) \quad \mathcal{C}(x, y) = \frac{\sigma^2}{2^{\nu-1}\Gamma(\nu)}(\kappa r)^\nu \mathcal{K}_\nu(\kappa r), \quad r = \|x - y\|_2, \quad \kappa = \frac{\sqrt{8\nu}}{\lambda}, \quad x, y \in D,$$

where σ^2 , ν , $\lambda > 0$ are the variance, smoothness parameter, and correlation length of the field, respectively, and \mathcal{K}_ν is the modified Bessel function of the second kind.

In practice, samples of u are needed only at discrete locations $\mathbf{x}_1, \dots, \mathbf{x}_m \in D$, and a simple sampling strategy consists of drawing realizations of a Gaussian vector $\mathbf{u} \sim \mathcal{N}(0, C)$

*Received by the editors March 12, 2018; accepted for publication (in revised form) September 9, 2018; published electronically November 20, 2018.

<http://www.siam.org/journals/juq/6-4/M117523.html>

Funding: This research was supported by EPSRC grant EP/K030930/1, by the EPSRC Centre For Doctoral Training in Industrially Focused Mathematical Modelling (EP/L015803/1) in collaboration with Simula Research Laboratory, and by the Nordic Council of Ministers through Nordforsk grant 74756 (AUQ-PDE).

[†]Mathematical Institute, University of Oxford, Oxford OX2 6GG, UK (matteo.croci@maths.ox.ac.uk, patrick.farrell@maths.ox.ac.uk, mike.giles@maths.ox.ac.uk).

[‡]Department for Numerical Analysis and Scientific Computing, Simula Research Laboratory, 1325 Lysaker, Norway (meg@simula.no).

with $\mathbf{u}_i = u(\mathbf{x}_i)$ and covariance matrix $C_{ij} = \mathbb{E}[u(\mathbf{x}_i)u(\mathbf{x}_j)]$. The simplest approach is usually computationally expensive as it requires the factorization of the dense covariance matrix $C \in \mathbb{R}^{m \times m}$. In fact, if we let $\mathbf{z} \sim \mathcal{N}(0, I)$ be a standard Gaussian vector and factorize $C = HH^T$ with $H \in \mathbb{R}^{m \times n}$, we can sample \mathbf{u} as $\mathbf{u} = H\mathbf{z}$ since

$$(1.2) \quad \mathbb{E}[\mathbf{u}\mathbf{u}^T] = \mathbb{E}[H\mathbf{z}(H\mathbf{z})^T] = H \mathbb{E}[\mathbf{z}\mathbf{z}^T]H^T = HH^T = C.$$

A basic form of this approach uses the Cholesky factorization of C . In this case, H is dense and the factorization has a computational complexity of $O(m^3)$. If the field is smooth enough so that the eigenvalues of C are rapidly decaying, this method can be made competitive by using instead a low-rank approximation [22]. Usually, n is taken to be equal to m so that H is square. However, this is not necessary for (1.2) to hold. For instance, the sampling strategy we present in this work uses $n > m$.

Another family of sampling approaches is based on the expansion of the field u as a (possibly finite or truncated) series of basis functions. Different choices of bases yield different methods. Common choices are the basis of the eigenfunctions of $\mathcal{C}(x, y)$ (Karhunen–Loève expansion), the Fourier basis (circulant embedding [10]), and a finite element basis [5, 11, 27, 40]. The former method is the most flexible as it can be used to sample Gaussian vectors with arbitrary covariance structure. However, it requires either the solution of a dense eigenvalue problem or the factorization of a dense covariance matrix [22]. If we need only the largest eigenvalues or a low-rank factorization, this approach is reasonably efficient [22]. However, if the eigenvalues of $\mathcal{C}(x, y)$ decay slowly (ν small), such operations become expensive as more terms are needed in the expansion. Circulant embeddings are exact and more efficient but rely on the use of the fast Fourier transform, the computation of which typically requires simple geometries and uniform structured meshes.

In this paper we consider the finite element basis method. Whittle showed in [32] that a Matérn field with covariance given by (1.1) is the statistically stationary solution that satisfies the linear elliptic partial differential equation (PDE),

$$(1.3) \quad (\mathcal{I} - \kappa^{-2}\Delta)^k u(x, \omega) = \eta \dot{W}(\cdot, \omega), \quad x \in \mathbb{R}^d, \quad \omega \in \Omega, \quad \nu = 2k - d/2 > 0,$$

where \dot{W} is spatial Gaussian white noise in \mathbb{R}^d , $k > d/4$, and Ω is a suitable sample space. The notation $\dot{W}(\cdot, \omega)$ indicates that \dot{W} is almost surely a generalized function (on \mathbb{R}^d). Here η is a scaling factor that depends on σ , λ , and ν , $d \leq 3$, and the equality has to hold almost surely and be interpreted in the sense of distributions. Boundary conditions are not needed as the stationarity requirement is enough for well-posedness [27]. Equation (1.3) has to be solved on the whole \mathbb{R}^d . However, this is generally not feasible, and \mathbb{R}^d is in practice truncated to a bounded domain D . In this case, artificial boundary conditions must be prescribed on ∂D . Homogeneous Dirichlet or Neumann boundary conditions are often chosen [5, 27], although it usually does not matter for practical purposes as the error in the covariance of the field decays rapidly away from the boundary [33]. After the meshing of D , (1.3) can be solved in linear time with the finite element method (FEM) and an optimally preconditioned Krylov solver. This approach thus scales well in terms of problem size and parallel computation [11]. Moreover, the approach is especially convenient if u appears as a coefficient in a PDE that is

solved using the FEM as it might be possible to reuse finite element bases and computations for both equations.

The main focus of this paper is the generation of white noise samples $\dot{W}(\cdot, \omega)$ for a given sample point $\omega \in \Omega$. More precisely, we study the efficient sampling of the action $\langle \dot{W}, v_h \rangle(\omega)$ of white noise onto a FEM test function v_h . In this work, for the sake of simplicity we specifically consider (1.3). However, the sampling techniques we describe apply to a wider range of stochastic PDEs (SPDEs) with additive spatial white noise forcing (see, e.g., [12, 40]). While solving such equations is relatively straightforward, the sampling of white noise realizations is not, as it requires the sampling of a Gaussian vector with a finite element mass matrix M as covariance. If the finite element spaces involved are other than piecewise constants, M will be sparse but not diagonal. Hence, its Cholesky factor is usually dense, and the sampling requires an offline computational and memory storage cost of $O(m^3)$ for the factorization, and an online cost of $O(m^2)$ for each sample.

To resolve this challenge, different approaches have been adopted in the literature. The generally adopted approach has been to use a diagonal mass matrix, i.e., an approximate representation using piecewise constants or mass-lumping. Osborn, Vassilevski, and Villa [30] use a two-field reformulation of (1.3) for $k = 1$ with Raviart–Thomas elements combined with piecewise constants, while Lindgren, Rue, and Lindström [27] use continuous Lagrange elements and mass-lumping. Both methods compute (or approximate) the action of white noise on the FEM test functions. Another option, adopted in [11, 12, 33], is to approximate the white noise itself by a piecewise constant random function that converges in an appropriate weak sense to the exact white noise.

The sampling becomes more complicated when the Matérn field u is needed within a multilevel Monte Carlo (MLMC) framework [19, 20], which requires the coupling of the field between different approximation levels (i.e., the same sample point ω must be used on both levels). In turn, this requires the white noise samples on each level to be coupled. Drzisga et al. [11] enforce this coupling in the nested grid case with the use of a piecewise constant approximation of white noise [34]. Osborn, Vassilevski, and Villa [30] present a technique that enforces the coupling between nested meshes by using techniques from element-based algebraic multigrid (AMG). Their approach does not require a user-provided hierarchy of nested grids as the hierarchy is constructed algebraically. This operation aggregates the elements of a single user-provided grid into clusters which then constitute the elements of the coarse meshes. The resulting aggregated meshes are nonsimplicial. Furthermore, Osborn et al. [31] use a hierarchy of nested structured grids on which they enforce the white noise coupling and solve the SPDE (1.3). The techniques used for the coupling are the same as those presented in [30]. The sampled Matérn fields are then transferred to a nonnested agglomerated mesh of the domain of interest via a Galerkin projection.

The main contributions of this paper are the following. First, we present a sampling technique for white noise that is exact and that is applicable for a wide range of finite element families, including all types of Lagrange elements. Our technique does not require the expensive factorization of a global mass matrix or a costly two-field splitting of the Laplacian, and it has linear complexity in the number of degrees of freedom. Second, we introduce a coupling technique for coupling white noise between nested or nonnested meshes, applicable for the same class of finite element families. If nonnested meshes are used, this coupling technique

requires the use of a supermesh construction [15, 16, 17]. Third, the existing literature generally focuses on white noise coupling in the h -refinement case, i.e., when the MLMC hierarchy is defined by meshes of decreasing element size [8, 9]. In this paper we also consider the case in which the MLMC levels are defined by increasing the polynomial degree of the FEM interpolant (p -refinement).

Although Osborn, Vassilevski, and Villa [30] also work with nonnested meshes, our approach differs significantly from theirs. They start from one single mesh and algebraically coarsen it to obtain the grid hierarchy. The MLMC levels are thus generated algebraically. Our approach operates on a given arbitrary mesh hierarchy, and the MLMC levels are defined geometrically. In our case, every mesh in the hierarchy is simplicial, and it is thus possible to use standard FEM error estimates (if available) to estimate a priori the MLMC convergence parameters [8, 37].

We adopt the same embedded domain strategy as that in [30]. The advantage of this strategy is that the sampled Matérn field can be transferred to the computational domain of interest exactly and at negligible cost. However, in practical applications defined over complex geometries, a sequence of nested meshes might not be available, making the white noise coupling challenging. This motivated us to design an algorithm that can be used to enforce the coupling between nonnested meshes as well.

The paper is structured as follows. In section 2 we introduce notation and give a brief description of the MLMC method. In section 3 we present an overview of the white noise SPDE sampling approach for Matérn fields and suggest a simple FEM scheme for the solution of (1.3). Moreover, we describe the white noise sampling problem for the cases where both independent and coupled realizations are needed. In section 4 we describe our new sampling technique that allows efficient sampling of independent and coupled white noise realizations. In section 5 we present numerical results corroborating the theoretical results and demonstrating the performance of the technique. Finally, we summarize the results of the paper in section 6.

2. Notation and preliminaries.

2.1. Notation. In this paper we adopt the following notation.

L^2 inner product. For an open domain $D \subseteq \mathbb{R}^d$, we let (\cdot, \cdot) denote the $L^2(D)$ inner product, where $L^2(D)$ is the standard Hilbert space of square-integrable functions on D .

Real-valued random variables. For a given σ -algebra \mathcal{A} and probability measure \mathbb{P} let $(\Omega, \mathcal{A}, \mathbb{P})$ be a probability space, and let $L^2(\Omega, \mathbb{R})$ indicate the space of real-valued random variables of finite second moment.

Generalized stochastic fields. Following the definition introduced by Itô [25] we denote with $\mathcal{L}(L^2(D), L^2(\Omega, \mathbb{R}))$ the space of generalized stochastic fields that are continuous linear mappings from $L^2(D)$ to $L^2(\Omega, \mathbb{R})$. For a given $\xi \in \mathcal{L}(L^2(D), L^2(\Omega, \mathbb{R}))$ we indicate the action of ξ onto a function $\phi \in L^2(D)$ with the notation $\xi(\phi) = \langle \xi, \phi \rangle$.

Subsets of compact closure. Given an open domain $G \subseteq D$, we write $G \subset\subset D$ to indicate that the closure of G is a compact subset of D .

Nested and nonnested meshes. Let T_a and T_b be two tessellations of D . We say that T_a is nested within T_b if $\text{vertices}(T_a) \subseteq \text{vertices}(T_b)$ and if for each element $e \in T_a$ there exists a set of elements $E \subseteq T_b$ such that $e = \bigcup_{\hat{e}_i \in E} \hat{e}_i$. We say that T_a and T_b are nonnested if T_a is not nested within T_b and vice versa.

Additionally, we will use the following definition of *white noise*.

Definition 2.1 (white noise; see Example 1.2 and Lemma 1.10 in [24]). Let $D \subseteq \mathbb{R}^d$ be an open domain. The white noise $\dot{W} \in \mathcal{L}(L^2(D), L^2(\Omega, \mathbb{R}))$ is a generalized stochastic field such that for any collection of $L^2(D)$ functions $\{\phi_i\}$, if we let $b_i = \langle \dot{W}, \phi_i \rangle$, then $\{b_i\}$ are joint Gaussian random variables with zero mean and covariance given by $\mathbb{E}[b_i b_j] = (\phi_i, \phi_j)$.

2.2. The multilevel Monte Carlo method. Let $u(x, \omega)$ for $x \in \mathbb{R}^d$, $\omega \in \Omega$ be the solution of an SPDE of interest, e.g., (1.3). Generally we are interested in computing an output functional P of u , namely,

$$(2.1) \quad P(\omega) = \mathcal{P}[x, u(x, \omega)](\omega).$$

Here we assume that $P(\omega)$ is scalar-valued with bounded second moment, i.e., $P \in L^2(\Omega, \mathbb{R})$. For instance, \mathcal{P} could be the average or the L^2 norm of u over its domain of definition. A more complicated, yet common, case is when the computation of \mathcal{P} requires the solution of an additional equation. A typical example is when u is a Matérn field satisfying (1.3) and u appears in the permeability coefficient of another elliptic PDE [9]. In this case, \mathcal{P} is usually a functional of the solution of the latter equation [9]. Although the techniques we describe apply to a wider range of problems, this is the framework considered in this paper.

To quantify the propagation of uncertainty from the input u to the output functional of interest P , one may estimate the expected value \mathbb{E} and variance \mathbb{V} of $P(\omega)$. When P can be approximated at different levels $\ell = 1, \dots, L$ of increasing accuracy and cost, $\mathbb{E}[P]$ and $\mathbb{V}[P]$ can be estimated efficiently with the MLMC method [19]. Letting P_ℓ be the approximated value at level ℓ , we can approximate $\mathbb{E}[P]$ as

$$(2.2) \quad \mathbb{E}[P] \approx \mathbb{E}[P_L] = \sum_{\ell=1}^L \mathbb{E}[P_\ell - P_{\ell-1}], \quad P_0 \equiv 0.$$

The telescoping sum on the right-hand side is at the heart of the MLMC strategy: by approximating each term in the sum with standard Monte Carlo we obtain the MLMC estimator,

$$(2.3) \quad \mathbb{E}[P] \approx \bar{P} = \sum_{\ell=1}^L \left[\frac{1}{N_\ell} \sum_{n=1}^{N_\ell} (P_\ell(\omega_\ell^n) - P_{\ell-1}(\omega_\ell^n)) \right],$$

where $\omega_\ell^n \in \Omega$ is the n th sample point on level ℓ .

In the context of approximating u and a posteriori P using a FEM, the levels of accuracy can be defined by using a hierarchy of meshes (h -refinement) or by increasing the polynomial degree of the finite elements used (p -refinement). As the variance is yet another expectation, the same strategy as described here for $\mathbb{E}[P]$ applies to $\mathbb{V}[P]$.

The increased efficiency of MLMC with respect to standard Monte Carlo relies on the assumption that on coarse levels (small ℓ), many samples are needed for an accurate estimate of the expected value, but each sample is inexpensive to compute. On fine levels (large ℓ), sampling is expensive, but the variance is small because the levels are coupled; i.e., the sample point ω_ℓ^n is the same for both $P_\ell(\omega_\ell^n)$ and $P_{\ell-1}(\omega_\ell^n)$. The coupling makes $P_\ell(\omega_\ell^n)$ and $P_{\ell-1}(\omega_\ell^n)$ strongly correlated. This aspect diminishes the variance of their difference, and therefore

fewer samples are required to estimate the expected value. An alternative interpretation is that once ω_ℓ^n is fixed, $P_\ell(\omega_\ell^n)$ and $P_{\ell-1}(\omega_\ell^n)$ are approximations of different accuracy of the same deterministic problem, and hence their difference becomes smaller as the discretization approaches the infinite dimensional solution.

MLMC can be seen as a variance reduction technique in which the coupling between the levels is one of the key elements. If the coupling is not enforced correctly so that the samples of P_ℓ and $P_{\ell-1}$ become independent, then the variance of each term of the telescoping sum in (2.3) increases, significantly harming its efficiency and convergence properties.

The convergence and cost of the MLMC estimator (2.3) is given by the following theorem.

Theorem 2.2 (MLMC convergence [9, 19]). *Let C_ℓ be the cost of computing one sample of P_ℓ on level ℓ . Suppose there are positive constants α, β, γ , such that $\alpha \geq \min(\beta, \gamma)$ and*

- (1) $|\mathbb{E}[P_\ell - P]| \lesssim 2^{-\alpha\ell}$,
- (2) $\mathbb{V}[P_\ell - P_{\ell-1}] \lesssim 2^{-\beta\ell}$,
- (3) $C_\ell \lesssim 2^{\gamma\ell}$.

Then, for any $\varepsilon < e^{-1}$, there exists a value L and a sequence $\{N_\ell\}_{\ell=1}^L$ such that

$$(2.4) \quad \text{error}(\bar{P}) \equiv \mathbb{E}[(\bar{P} - \mathbb{E}[P])^2]^{1/2} \leq \varepsilon,$$

and the total cost C_{tot} satisfies

$$(2.5) \quad C_{\text{tot}} := \sum_{\ell=1}^L N_\ell C_\ell \lesssim \begin{cases} \varepsilon^{-2}, & \beta > \gamma, \\ \varepsilon^{-2}(\log \varepsilon)^2, & \beta = \gamma, \\ \varepsilon^{-2-(\gamma-\beta)/\alpha}, & \beta < \gamma. \end{cases}$$

The values of L and $\{N_\ell\}_{\ell=1}^L$ are estimated automatically in standard MLMC algorithms; for further details see [19, 20]. The values of the MLMC parameters α, β, γ are sometimes known a priori [8, 37]; otherwise, they need to be estimated. In the uniform h -refinement case, we have $\gamma = d$ and $h_\ell \sim 2^{-c\ell}$, where h_ℓ is the level ℓ mesh size and $c > 0$ [9].

3. The finite element approach to Matérn field sampling. In this section we describe the practical aspects of the numerical solution of (1.3) when either independent (standard Monte Carlo) or coupled (MLMC) Matérn field samples are needed. As we will see, the main complication lies in the sampling of white noise realizations.

Note that the results we present on white noise sampling can be applied to a wider class of elliptic or parabolic SPDEs with additive spatial white noise (see, e.g., [12, 40]).

3.1. Finite element solution of elliptic PDEs with white noise forcing. The solutions of the linear elliptic PDE (1.3) correspond to a Matérn field with covariance given by (1.1). In this paper we assume that k is a positive integer, although it is possible to work with noninteger values as well [6]. The scaling factor η in (1.3) is given by

$$(3.1) \quad \eta = \frac{\sigma}{\hat{\sigma}}, \quad \text{where} \quad \hat{\sigma}^2 = \frac{\Gamma(\nu) \nu^{d/2}}{\Gamma(\nu + d/2)} \left(\frac{2}{\pi}\right)^{d/2} \lambda^{-d},$$

where $\Gamma(x)$ is the Euler gamma function [27]. Note that if $d = 2$, then $\hat{\sigma}^2 = (2/\pi)\lambda^{-2}$, and for $\nu \rightarrow \infty$, $\hat{\sigma}^2 = (2/\pi)^{d/2}\lambda^{-d}$.

Solving (1.3) over the whole of \mathbb{R}^d is generally not feasible. Instead, \mathbb{R}^d is typically truncated to a bounded domain $D \subset \subset \mathbb{R}^d$, and some boundary conditions are chosen—usually homogeneous Neumann or Dirichlet [5, 27]. In what follows, we assume that the Matérn field sample is needed on a domain $G \subset \subset D$. If D is sufficiently large in the sense that the distance between ∂D and ∂G is larger than the correlation length λ , then the error introduced by truncating \mathbb{R}^d to D is negligible [11, 33].

After truncating the domain, (1.3) can be rewritten in the iterative form

$$(3.2) \quad \begin{cases} u_1 - \kappa^{-2} \Delta u_1 = \eta \dot{W} & \text{in } D, \\ u_{j+1} - \kappa^{-2} \Delta u_{j+1} = u_j & \text{in } D, \quad j = 1, \dots, k-1, \\ u_{j+1} = 0 & \text{on } \partial D, \quad j = 0, \dots, k-1, \end{cases}$$

where $u \equiv u_k$. This is the approach suggested by Lindgren, Rue, and Lindström in [27]. As the main focus of this paper is on white noise sampling, we will restrict our attention to the $k = 1$ case, and we will set $\eta = 1$ from now on, in which case (3.2) reduces to

$$(3.3) \quad \begin{aligned} u - \kappa^{-2} \Delta u &= \dot{W} & \text{in } D, \\ u &= 0 & \text{on } \partial D. \end{aligned}$$

The existence and uniqueness of solutions to (3.3) were proven in [4] and [7], respectively.

We will solve (3.3) using the FEM. Let $V_h = \text{span}(\phi_1, \dots, \phi_m) \subseteq H_0^1(D)$ be a suitable finite element approximation subspace (e.g., the ϕ_i could be continuous Lagrange basis functions defined relative to a triangulation D_h of D). A discrete weak form of (3.3) then reads as follows: find $u_h \in V_h$ such that

$$(3.4) \quad (u_h, v_h) + \kappa^{-2} (\nabla u_h, \nabla v_h) = \langle \dot{W}, v_h \rangle \quad \text{for all } v_h \in V_h.$$

The coefficients of the basis function expansion for u_h , i.e., the u_i such that $u_h = \sum_{i=1}^m u_i \phi_i$, are given by the solution of the linear system

$$(3.5) \quad A\mathbf{u} = \mathbf{b}, \quad \text{with} \quad A_{ij} = (\phi_i, \phi_j) + \kappa^{-2} (\nabla \phi_i, \nabla \phi_j), \quad b_i = \langle \dot{W}, \phi_i \rangle.$$

This linear system (3.5) can be solved in $O(m)$ time by using an optimally preconditioned Krylov solver such as the conjugate gradient method preconditioned with geometric or algebraic multigrid. We remark that since the elliptic operator is the same in all equations of (3.2), the same finite element basis and solver can be reused to compute all $\{u_j\}_{j=1}^m$ for the case where $k > 1$.

By Definition 2.1, \mathbf{b} satisfies

$$(3.6) \quad \mathbf{b} \sim \mathcal{N}(0, M), \quad M_{ij} = (\phi_i, \phi_j);$$

i.e., \mathbf{b} is a zero-mean Gaussian vector with the finite element mass matrix M as covariance matrix. Sampling white noise realizations can thus be accomplished by sampling a Gaussian vector of mass matrix covariance.

In section 4, we present a factorization of M in the form HH^T (cf. (1.2)) that is both sparse and computationally efficient to compute, and thus allows for efficient sampling of white noise.

3.2. Multilevel white noise sampling/white noise coupling condition. We now consider the case in which coupled Matérn field realizations are needed in an MLMC setting; i.e., we want to draw samples of $u_\ell(x, \omega)$ and $u_{\ell-1}(x, \omega)$ at two different levels of accuracy ℓ and $\ell - 1$ for the same $\omega \in \Omega$. Since the only stochastic element present in (3.3) is white noise, it is sufficient to use the same white noise sample on both levels to enforce the coupling requirement.

More precisely, let V^ℓ and $V^{\ell-1}$ be the finite element spaces on levels ℓ and $\ell - 1$, respectively, for $\ell > 1$. We consider the following two variational problems coupled by a common white noise sample: find $u_\ell \in V^\ell = \text{span}(\phi_1^\ell, \dots, \phi_{m_\ell}^\ell)$ and $u_{\ell-1} \in V^{\ell-1} = \text{span}(\phi_1^{\ell-1}, \dots, \phi_{m_{\ell-1}}^{\ell-1})$ such that for $\omega_\ell^n \in \Omega$,

$$(3.7) \quad (u_\ell, v_\ell) + \kappa^{-2}(\nabla u_\ell, \nabla v_\ell) = \langle \dot{W}, v_\ell \rangle(\omega_\ell^n) \quad \text{for all } v_\ell \in V^\ell,$$

$$(3.8) \quad (u_{\ell-1}, v_{\ell-1}) + \kappa^{-2}(\nabla u_{\ell-1}, \nabla v_{\ell-1}) = \langle \dot{W}, v_{\ell-1} \rangle(\omega_\ell^n) \quad \text{for all } v_{\ell-1} \in V^{\ell-1},$$

where the terms on the right-hand side are coupled in the sense that they are centered Gaussian random variables with covariance $\mathbb{E}[\langle \dot{W}, v_l \rangle \langle \dot{W}, v_s \rangle] = (v_l, v_s)$ for $l, s \in \{\ell, \ell - 1\}$, as given by Definition 2.1.

Let $\mathbf{u}_\ell \in \mathbb{R}^{m_\ell}$ and $\mathbf{u}_{\ell-1} \in \mathbb{R}^{m_{\ell-1}}$ be the vectors of the finite element expansion coefficients of u_ℓ and $u_{\ell-1}$, respectively. Following the approach from section 3.1, we note that the coefficient vectors solve the following block diagonal linear system:

$$(3.9) \quad \left[\begin{array}{c|c} A^\ell & 0 \\ \hline 0 & A^{\ell-1} \end{array} \right] \begin{bmatrix} \mathbf{u}_\ell \\ \mathbf{u}_{\ell-1} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_\ell \\ \mathbf{b}_{\ell-1} \end{bmatrix}.$$

Alternatively, by letting $\mathbf{u} = [\mathbf{u}_\ell, \mathbf{u}_{\ell-1}]^T$, $\mathbf{b} = [\mathbf{b}_\ell, \mathbf{b}_{\ell-1}]^T$, and $A = \text{diag}(A^\ell, A^{\ell-1})$, we can write this as

$$(3.10) \quad A\mathbf{u} = \mathbf{b}.$$

This system can be solved in linear time with an optimal solver [13].

Furthermore, by Definition 2.1,

$$(3.11) \quad \mathbf{b} \sim \mathcal{N}(0, M),$$

where M can be expressed in block structure as

$$(3.12) \quad M = \left[\begin{array}{c|c} M^\ell & M^{\ell, \ell-1} \\ \hline (M^{\ell, \ell-1})^T & M^{\ell-1} \end{array} \right], \quad \text{with} \quad M_{ij}^{\ell, k} = (\phi_i^\ell, \phi_j^k) \text{ and } M_{ij}^k = (\phi_i^k, \phi_j^k).$$

If we were using independent white noise samples for (3.7) and (3.8), then the off-diagonal blocks of M would vanish. Conversely, the presence of the mixed mass matrix $M^{\ell, \ell-1}$ stems from the use of the same white noise sample on both levels. For this reason, we will refer to (3.11) and (3.12) as the *coupling condition*.

Thus, the problem of sampling coupled Matérn fields in the context of MLMC again reduces to the sampling of a Gaussian vector with a mass matrix as covariance. However,

two additional complications arise. First, M is potentially much larger and not necessarily of full rank (consider the case in which $V^\ell = V^{\ell-1}$; then $M^\ell = M^{\ell-1} = M^{\ell, \ell-1}$). Second, to assemble $M^{\ell, \ell-1}$ we need to compute integrals involving basis functions possibly defined over different, nonnested meshes, which is nontrivial. In section 4, we present a sampling technique that addresses both issues. A supermesh construction [16, 17] is required in the nonnested mesh case.

3.3. Embedded meshes and nonnested grids. We adopt the embedded mesh strategy presented by Osborn, Vassilevski, and Villa [30]. We assume that the Matérn field sample is needed on a user-provided mesh G_h of the domain G , and we take D to be a larger d -dimensional box such that the distance between ∂G and ∂D is at least λ . With modern meshing software, such as Gmsh [18], it is possible to then triangulate D and obtain a mesh D_h in such a way that G_h is nested within D_h ; i.e., each element and vertex of G_h is also an element or vertex of D_h . We then refer to G_h as embedded in D_h or to G_h as an embedded mesh (in D_h). The main advantage of an embedded G_h in D_h is that once (3.3) is solved on D_h , the sampled Matérn field u can be exactly transferred onto G_h at negligible cost. Conversely, if G_h is not embedded in D_h , an additional interpolation step would be required, and this would increase the cost of each sample.

In the MLMC framework with h -refinement, we assume that we are given a possibly nonnested user-provided mesh hierarchy $\{G_h^\ell\}_{\ell=1}^L$. We accordingly generate a hierarchy of meshes $\{D_h^\ell\}_{\ell=1}^L$ on which to perform the sampling. If the $\{D_h^\ell\}_{\ell=1}^L$ are nested, then the techniques used in [11] and [30] can be used to couple the white noise between MLMC levels. However, in the case in which the user-provided meshes $\{G_h^\ell\}_{\ell=1}^L$ are nonnested, these methods are not compatible with the embedded mesh strategy.

Clearly, nonnested grid hierarchies appear naturally in practical computations on complex geometries. For instance, grid hierarchies generated from CAD geometries or through coarsening of a single fine mesh are generally nonnested. Thus, tackling couplings across nonnested meshes is crucial for nontrivial applications. As we will see in the next section, such couplings can be achieved at a small offline cost.

4. White noise sampling. In this section we introduce a new technique for efficient sampling of white noise. We first address the basic case, in which independent white noise samples are needed, before considering the more complicated case in which coupled samples are required.

4.1. Sampling of independent white noise realizations. As discussed in the previous section, the sampling of independent white noise realizations defined over a meshed domain can be cast as the sampling of a Gaussian vector \mathbf{b} of covariance matrix given by a finite element mass matrix $M \in \mathbb{R}^{m \times m}$. In turn, efficient sampling of such a Gaussian vector typically involves computing a factorization of $M = HH^T$. If a Cholesky factorization is used, such sampling may become costly, with an $O(m^3)$ factorization cost and $O(m^2)$ cost per sample. In what follows, we present an alternative factorization strategy which has $O(m)$ fixed cost and $O(m)$ cost per sample.

The core idea is to work elementwise instead of factorizing a global mass matrix. To illustrate this idea, we consider a standard finite element assembly of M over a mesh with

n elements (i.e., cells) and m_e degrees of freedom on each element. Local mass matrices M_e of size $m_e \times m_e$ are computed on each mesh element e before aggregation to form the global mass matrix M . The overall assembly operation can be written in matrix form

$$(4.1) \quad M = L^T \text{diag}_e(M_e)L$$

(see, e.g., [39]), where $\text{diag}_e(M_e)$ is a block diagonal matrix of size $nm_e \times nm_e$ with the local mass matrices on the diagonal, L is a Boolean assembling matrix of size $nm_e \times m$ such that $L^T = [L_1^T \dots L_n^T]$, and the $\{L_e\}_{e=1}^n$ are Boolean matrices of size $m_e \times m$ that encode the local-to-global map. Note that each row of L has exactly one nonzero entry [39].

We can now factorize each local mass matrix M_e independently with a standard Cholesky factorization to obtain $M_e = H_e H_e^T$ for each element e . We then have

$$(4.2) \quad M = L^T \text{diag}_e(H_e H_e^T)L = (L^T \text{diag}_e(H_e))(L^T \text{diag}_e(H_e))^T = HH^T,$$

with $H \equiv L^T \text{diag}_e(H_e)$, and we can sample \mathbf{b} by computing

$$(4.3) \quad \mathbf{b} = Hz, \quad \text{with } z \sim \mathcal{N}(0, I), \quad z \in \mathbb{R}^{m_e n},$$

since (cf. (1.2))

$$(4.4) \quad \mathbb{E}[\mathbf{b}\mathbf{b}^T] = H \mathbb{E}[zz^T]H^T = (L^T \text{diag}_e(H_e))I(L^T \text{diag}_e(H_e))^T = M.$$

Remark 4.1. This sampling strategy allows the splitting of a large global sampling problem into separate small local sampling problems. In fact, if for each element e we let $\mathbf{z}_e \sim \mathcal{N}(0, I)$ be a small standard Gaussian vector of length m_e , we can rewrite $\mathbf{b} = Hz$ as

$$(4.5) \quad \mathbf{b} = Hz = \sum_{e=1}^n L_e^T H_e \mathbf{z}_e = \sum_{e=1}^n L_e^T \mathbf{b}_e,$$

where $\mathbf{b}_e \sim \mathcal{N}(0, M_e)$ is sampled locally. The problem of sampling a global mass matrix covariance Gaussian vector then eventually reduces to the sampling of n independent local mass matrix covariance Gaussian vectors. This sampling approach is therefore trivially parallelizable.

Note that this sampling strategy is efficient since the local Cholesky factorizations can be computed in $O(m_e^3 n)$ time, and the L_e factors can be applied matrix-free for a total $O(m_e^3 n)$ factorization cost and an $O(m_e^2 n)$ memory and sampling cost.

Remark 4.2. In the case in which the transformation to the reference element is affine (such as with Lagrange elements on simplices) this operation can be made much more efficient by noting that the local mass matrices on each element are always the same up to a multiplicative factor, namely $M_e/|e| = \text{const}$ for all e , where $|e|$ is the measure of the element. It is therefore sufficient to factorize a single local mass matrix and to store its Cholesky factor, yielding negligible $O(m_e^3)$ and $O(m_e^2)$ factorization and memory costs, respectively.

We note that the standard Gaussian vector \mathbf{z} used to compute \mathbf{b} is of size m_{en} , which is larger than if a Cholesky factorization were used.¹ In fact, unlike the Cholesky factor, the matrix H here is not square. However, in comparison to the cost of solving (3.3), the sampling cost of the extra Gaussian variables is negligible.

4.2. Sampling coupled white noise realizations for MLMC. We now consider the case of sampling coupled white noise. In what follows, we consider the general setting in which the MLMC levels are defined using h -refinement and the mesh hierarchy is nonnested. At the end of this section, we provide some remarks on the simpler cases in which the function spaces that define the hierarchy are nested (e.g., the grids are nested or p -refinement is used).

4.2.1. Supermesh construction and global mass matrix assembly. Consider the case of sampling $\mathbf{b} \sim \mathcal{N}(0, M)$, where $M \in \mathbb{R}^{m \times m}$ is given by (3.12). The assembly of the off-diagonal blocks of M requires the computation of inner products between basis functions of different FEM approximation subspaces. To address this problem, we use a *supermesh* construction defined as follows.

Definition 4.1 (supermesh [16, 17]). Let $D \subset \subset \mathbb{R}^d$ be an open domain, and let T_a, T_b be two tessellations of D . A supermesh S of T_a and T_b is a common refinement of T_a and T_b . More specifically, S is a triangulation of D such that

1. $\text{vertices}(T_a) \cup \text{vertices}(T_b) \subseteq \text{vertices}(S)$,
2. $\text{volume}(e_S \cap e) \in \{0, \text{volume}(e_S)\}$ for all elements $e_S \in S$, $e \in (T_a \cup T_b)$.

The first condition means that every parent mesh vertex must also be a vertex of the supermesh, while the second states that every supermesh element is completely contained within exactly one element of either parent mesh [17]. As stated in [17, Lemma 2], supermesh elements always lie within the intersection of a single pair of parent mesh elements. The supermesh construction is not unique [17]. We show an example of supermesh construction in Figure 1. Efficient algorithms for computing supermeshes are available from [29].

Remark 4.3 (on the complexity of the supermesh construction). If the supermesh construction is performed with a local supermeshing algorithm, then its complexity is $O(n_\ell + K)$, where n_ℓ is the number of elements of D_h^ℓ and K is the number of intersecting elements [15]. The number of supermesh elements is proportional to the number of intersecting elements and it is therefore $O(K)$. Theoretically, K is bounded by $K \leq c(d)n_\ell n_{\ell-1}$, where $n_{\ell-1}$ is the number of elements of $D_h^{\ell-1}$, $c = 4$ in 2D and $c = 45$ in 3D [15, 17]. In practice, this is a pessimistic bound. If a typical element of the first mesh intersects with \bar{k} elements of the second mesh, then the total number of intersections is $K = O(\bar{k}n_\ell)$. For instance, the lib-supermesh software [29] uses heuristics to eliminate near-degenerate supermesh elements and ensures that \bar{k} is always bounded by a constant. For practical computations, the supermesh construction and the number of supermesh elements is therefore $O(n_\ell)$.

Evaluating (3.12) involves L^2 inner products of functions that are only *piecewise* polynomial on each element of D_h^ℓ and $D_h^{\ell-1}$. This lack of smoothness affects the convergence of standard quadrature schemes. The supermesh construction provides a resolution to this

¹A Cholesky factor would be of size $m \times m$, yielding a standard Gaussian vector of length m .

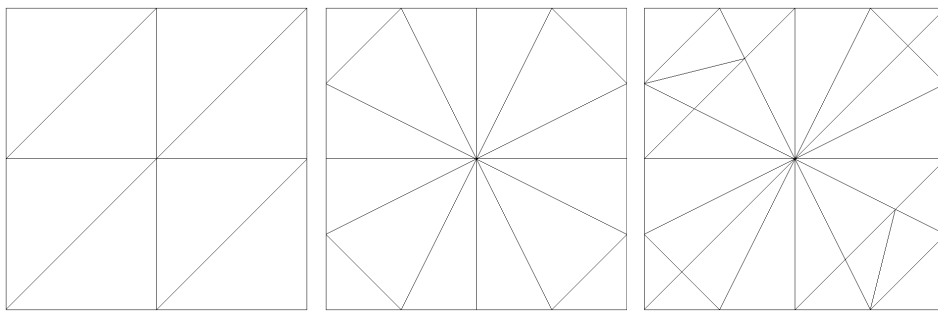


Figure 1. An example of a supermesh construction. The first two meshes on the left are the parent meshes, and the mesh on the right is a supermesh. As stated in [17, Lemma 2], every supermesh element is completely contained within a unique pair of parent mesh elements.

problem: on each element of a supermesh of D_h^ℓ and $D_h^{\ell-1}$ the integrands are polynomial, and standard quadrature schemes apply.

Overall, our strategy for evaluating (3.12) is to construct a supermesh of each pair of meshes $D_h^\ell, D_h^{\ell-1}$. Note that since each supermesh element lies in the intersection of exactly one pair $(e_\ell, e_{\ell-1})$ of parent mesh elements $e_\ell \in D_h^\ell, e_{\ell-1} \in D_h^{\ell-1}$, we need only account for the basis functions that are nonzero over e_ℓ and $e_{\ell-1}$. Let m_{e_ℓ} and $m_{e_{\ell-1}}$ denote the number of degrees of freedom defined by the finite element spaces V^ℓ and $V^{\ell-1}$ over elements e_ℓ and $e_{\ell-1}$, respectively. Then, only the inner products between $m_e = m_{e_\ell} + m_{e_{\ell-1}}$ basis functions will be nonzero.

We can thus assemble M given by (3.12) by the following two-step algorithm.

1. Let n be the number of supermesh elements. For each supermesh element e , use quadrature rules over e to compute the local mass matrix

$$(4.6) \quad M_e = \left[\begin{array}{c|c} M_e^\ell & M_e^{\ell, \ell-1} \\ \hline (M_e^{\ell, \ell-1})^T & M_e^{\ell-1} \end{array} \right], \quad (M_e^{\ell, \ell-1})_{ij} = \int_e \phi_i^\ell \phi_j^{\ell-1} dx,$$

where $\{\phi_i^\ell\}_{i=1}^{m_{e_\ell}}$ and $\{\phi_j^{\ell-1}\}_{j=1}^{m_{e_{\ell-1}}}$ are sets of the basis functions of V^ℓ and $V^{\ell-1}$, respectively, that have nonzero support over e . M_e is of size $m_e \times m_e$, M_e^ℓ is of size $m_{e_\ell} \times m_{e_\ell}$, $M_e^{\ell-1}$ is of size $m_{e_{\ell-1}} \times m_{e_{\ell-1}}$, and $M_e^{\ell, \ell-1}$ is of size $m_{e_\ell} \times m_{e_{\ell-1}}$.

2. Let L^ℓ and $L^{\ell-1}$ be the supermesh assembling matrices of the finite element spaces V^ℓ and $V^{\ell-1}$, respectively, mapping the local supermesh cell degrees of freedom to the global degrees of freedom of V^ℓ . Assemble the local supermesh contributions together as follows:

$$(4.7) \quad M = \left[\begin{array}{c|c} (L^\ell)^T \text{diag}_e(M_e^\ell) L^\ell & (L^\ell)^T \text{diag}_e(M_e^{\ell, \ell-1}) L^{\ell-1} \\ \hline (L^{\ell-1})^T \text{diag}_e(M_e^{\ell, \ell-1})^T L^\ell & (L^{\ell-1})^T \text{diag}_e(M_e^{\ell-1}) L^{\ell-1} \end{array} \right].$$

Observe that (4.7) and (3.12) agree since

$$(4.8) \quad \begin{aligned} M^l &= (L^l)^T \text{diag}_e(M_e^l) L^l \quad \text{for } l \in \{\ell, \ell-1\}, \\ M^{\ell, \ell-1} &= (L^\ell)^T \text{diag}_e(M_e^{\ell, \ell-1}) L^{\ell-1}. \end{aligned}$$

Note that the above is again just the assembly of the contributions of each supermesh element to the global mass matrices in matrix form. As we will see next, we actually do not need to assemble M but only the local mass matrices M_e^ℓ and $M_e^{\ell-1}$ for each supermesh element e .

4.2.2. From global to local: The local coupling condition. We again use a divide-and-conquer strategy to split the global sampling problem into smaller local subproblems (cf. Remark 4.1). Suppose that we can sample a local Gaussian vector $\mathbf{b}_e \sim \mathcal{N}(0, M_e)$ on each supermesh element e . We can then separate \mathbf{b}_e into two Gaussian vectors \mathbf{b}_e^ℓ and $\mathbf{b}_e^{\ell-1}$ such that $\mathbf{b}_e = [(\mathbf{b}_e^\ell)^T, (\mathbf{b}_e^{\ell-1})^T]^T$ and

$$(4.9) \quad \mathbf{b}_e^\ell \sim \mathcal{N}(0, M_e^\ell), \quad \mathbf{b}_e^{\ell-1} \sim \mathcal{N}(0, M_e^{\ell-1}), \quad \mathbb{E}[\mathbf{b}_e^\ell (\mathbf{b}_e^{\ell-1})^T] = M_e^{\ell, \ell-1}.$$

Since (4.9) is the local equivalent of (3.11), we refer to it as the *local coupling condition*. Finally, we can use the same approach as that in (4.5) and assemble the coupled vectors \mathbf{b}^ℓ and $\mathbf{b}^{\ell-1}$ as

$$(4.10) \quad \mathbf{b}^\ell = \sum_{e=1}^n (L_e^\ell)^T \mathbf{b}_e^\ell, \quad \mathbf{b}^{\ell-1} = \sum_{e=1}^n (L_e^{\ell-1})^T \mathbf{b}_e^{\ell-1},$$

where n is the number of supermesh elements. This enforces the correct distribution since sums of Gaussian random variables are Gaussian, and the covariance structure is correct. In particular,

$$(4.11) \quad \begin{aligned} \mathbb{E}[\mathbf{b}^l (\mathbf{b}^l)^T] &= \sum_{i,j=1}^n (L_i^l)^T \mathbb{E}[\mathbf{b}_i^l (\mathbf{b}_j^l)^T] L_j^l = \sum_{i=1}^n (L_i^l)^T \mathbb{E}[\mathbf{b}_i^l (\mathbf{b}_i^l)^T] L_i^l \\ &= (L^l)^T \text{diag}_i(M_i^l) L^l = M^l \quad \text{for } l \in \{\ell, \ell-1\} \end{aligned}$$

and

$$(4.12) \quad \begin{aligned} \mathbb{E}[\mathbf{b}^\ell (\mathbf{b}^{\ell-1})^T] &= \sum_{i,j=1}^n (L_i^\ell)^T \mathbb{E}[\mathbf{b}_i^\ell (\mathbf{b}_j^{\ell-1})^T] L_j^{\ell-1} = \sum_{i=1}^n (L_i^\ell)^T \mathbb{E}[\mathbf{b}_i^\ell (\mathbf{b}_i^{\ell-1})^T] L_i^{\ell-1} \\ &= (L^\ell)^T \text{diag}_i(M_i^{\ell, \ell-1}) L^{\ell-1} = M^{\ell, \ell-1}, \end{aligned}$$

where we have used the facts that \mathbf{b}_i^l and \mathbf{b}_j^l are independent for $i \neq j$ for $l \in \{\ell, \ell-1\}$ and that \mathbf{b}_i^ℓ is independent of $\mathbf{b}_j^{\ell-1}$ if $i \neq j$. Thus, again the global sampling problem can be recast as a series of much smaller, independent, local sampling problems.

Finally, it remains to devise a strategy for sampling realizations of the local vectors \mathbf{b}_e on a given supermesh element e . The following result demonstrates that the covariance matrix of \mathbf{b}_e is singular and how such a sampling can be simplified.

Lemma 4.2. *Let V^ℓ and $V^{\ell-1}$ be finite element spaces over two tessellations $D_h^\ell, D_h^{\ell-1}$ of the same domain. Let S be a supermesh of D_h^ℓ and $D_h^{\ell-1}$. Let $\phi_i^\ell, \phi_j^{\ell-1}$ for $i = 1, \dots, m_{e_\ell}, j = 1, \dots, m_{e_{\ell-1}}$ be the basis functions of V^ℓ and $V^{\ell-1}$, respectively, that have nonzero support over e . Let $V^\ell|_e = \text{span}(\phi_1^\ell|_e, \dots, \phi_{m_{e_\ell}}^\ell|_e)$ and $V^{\ell-1}|_e = \text{span}(\phi_1^{\ell-1}|_e, \dots, \phi_{m_{e_{\ell-1}}}^{\ell-1}|_e)$ be the*

restrictions of V^ℓ and $V^{\ell-1}$ to e . Assume that $V^{\ell-1}|_e \subseteq V^\ell|_e$, i.e., that the restrictions are nested and that M_e^ℓ as defined in (4.6) is nonsingular; then

$$(4.13) \quad \text{rank}(M_e) = \text{rank}(M_e^\ell) \quad \text{and} \quad M_e^{\ell-1} = (M_e^{\ell,\ell-1})^T (M_e^\ell)^{-1} M_e^{\ell,\ell-1}.$$

Proof. Since $V^{\ell-1}|_e \subseteq V^\ell|_e$, then we have that, for all j , $\phi_j^{\ell-1} \in V^\ell|_e$, which in turn means that there exists a set of coefficients $r_{ji} \in \mathbb{R}$ such that $\phi_j^{\ell-1} = \sum_i r_{ji} \phi_i^\ell$. Now, let R_e be an $m_{e_\ell} \times m_{e_{\ell-1}}$ matrix such that $(R_e)_{i,j} = r_{ji}$, and define the vector functions

$$(4.14) \quad \phi^{\ell-1} = \begin{bmatrix} \phi_1^{\ell-1} \\ \vdots \\ \phi_{m_{e_{\ell-1}}}^{\ell-1} \end{bmatrix}, \quad \phi^\ell = \begin{bmatrix} \phi_1^\ell \\ \vdots \\ \phi_{m_{e_\ell}}^{\ell-1} \end{bmatrix}.$$

We then have that

$$(4.15) \quad \phi^{\ell-1} = R_e^T \phi^\ell.$$

This implies that we can now rewrite $M_e^{\ell-1}$ as

$$(4.16) \quad M_e^{\ell-1} = \int_e \phi^{\ell-1} (\phi^{\ell-1})^T dx = R_e^T \int_e \phi^\ell (\phi^\ell)^T dx R_e = R_e^T M_e^\ell R_e$$

since R_e is constant. Similarly, for $M_e^{\ell,\ell-1}$ we have

$$(4.17) \quad M_e^{\ell,\ell-1} = \int_e \phi^\ell (\phi^{\ell-1})^T dx = \int_e \phi^\ell (\phi^\ell)^T dx R_e = M_e^\ell R_e.$$

Combining (4.16) and (4.17) with the assumption that M_e^ℓ is invertible thus yields the second equation in (4.13) since

$$(4.18) \quad (M_e^{\ell,\ell-1})^T (M_e^\ell)^{-1} M_e^{\ell,\ell-1} = R_e^T M_e^\ell (M_e^\ell)^{-1} M_e^\ell R_e = R_e^T M_e^\ell R_e = M_e^{\ell-1}.$$

Pulling (4.6), (4.16), and (4.17) together, we can now express M_e as

$$(4.19) \quad M_e = \left[\begin{array}{c|c} M_e^\ell & M_e^\ell R_e \\ \hline R_e^T M_e^\ell & R_e^T M_e^\ell R_e \end{array} \right] = \left[\begin{array}{c|c} I & 0 \\ \hline R_e^T & I \end{array} \right] \left[\begin{array}{c|c} M_e^\ell & 0 \\ \hline 0 & 0 \end{array} \right] \left[\begin{array}{c|c} I & R_e \\ \hline 0 & I \end{array} \right],$$

where we have used the fact that M_e^ℓ is symmetric. Since M_e is symmetric and the two block triangular matrices on the right-hand side of (4.19) are invertible, Sylvester's law of inertia [36] gives that

$$(4.20) \quad \text{rank}(M_e) = \text{rank} \left(\left[\begin{array}{c|c} M_e^\ell & 0 \\ \hline 0 & 0 \end{array} \right] \right) = \text{rank}(M_e^\ell),$$

which concludes the proof. ■

The assumptions of Lemma 4.2 are mild and are satisfied by most finite element families, e.g., Lagrange elements (continuous piecewise polynomials defined relative to the tessellations).

Using Lemma 4.2, we can now sample $\mathbf{b}_e = [(\mathbf{b}_e^\ell)^T, (\mathbf{b}_e^{\ell-1})^T]^T$ by enforcing the local coupling condition (4.9) as follows. For each supermesh element e , do the following:

1. Compute $M_e^{\ell, \ell-1}$ and the Cholesky factorization $M_e^\ell = H_e H_e^T$.
2. Sample $\mathbf{z}_e \sim \mathcal{N}(0, I)$ of length m_{e_ℓ} and set $\mathbf{b}_e^\ell = H_e \mathbf{z}_e$.
3. Compute $\mathbf{b}_e^{\ell-1}$ as $\mathbf{b}_e^{\ell-1} = (M_e^{\ell, \ell-1})^T H_e^{-T} \mathbf{z}_e$.

Note that the \mathbf{b}_e^ℓ and $\mathbf{b}_e^{\ell-1}$ sampled this way satisfy the local coupling condition (4.9) since, by (4.13) and the fact that H_e is the Cholesky factor of M_e^ℓ , we have that, first,

$$(4.21) \quad \mathbb{E}[\mathbf{b}_e^\ell (\mathbf{b}_e^\ell)^T] = H_e \mathbb{E}[\mathbf{z}_e \mathbf{z}_e^T] H_e^T = M_e^\ell;$$

second,

$$(4.22) \quad \begin{aligned} \mathbb{E}[\mathbf{b}_e^{\ell-1} (\mathbf{b}_e^{\ell-1})^T] &= (M_e^{\ell, \ell-1})^T H_e^{-T} \mathbb{E}[\mathbf{z}_e \mathbf{z}_e^T] H_e^{-1} M_e^{\ell, \ell-1} \\ &= (M_e^{\ell, \ell-1})^T (H_e H_e^T)^{-1} M_e^{\ell, \ell-1} \\ &= (M_e^{\ell, \ell-1})^T (M_e^\ell)^{-1} M_e^{\ell, \ell-1} = M_e^{\ell-1}; \end{aligned}$$

and third,

$$(4.23) \quad \mathbb{E}[\mathbf{b}_e^\ell (\mathbf{b}_e^{\ell-1})^T] = H_e \mathbb{E}[\mathbf{z}_e \mathbf{z}_e^T] H_e^{-1} M_e^{\ell, \ell-1} = H_e H_e^{-1} M_e^{\ell, \ell-1} = M_e^{\ell, \ell-1}.$$

In the case in which the transformation to the reference element is affine (such as with Lagrange elements on simplices), the sampling can be made more efficient by sampling white noise directly on the supermesh and then interpolating it onto the parent meshes. This strategy exploits the following result.

Corollary 4.3 (of Lemma 4.2). *Let V^ℓ and $V^{\ell-1}$ be FEM approximation subspaces over two triangulations $D_h^\ell, D_h^{\ell-1}$ of the same domain. Let S be a supermesh of D_h^ℓ and $D_h^{\ell-1}$, and let V^S be a FEM approximation subspace over S . With the same notation as that in Lemma 4.2, for each supermesh element e let $V^S|_e, V^\ell|_e$, and $V_e^{\ell-1}$ be the restrictions of V^S, V^ℓ , and $V^{\ell-1}$ to e . Let M_e^S be the local mass matrix over $V^S|_e$. Assume that $V^\ell|_e, V_e^{\ell-1} \subseteq V^S|_e$, i.e. that the parent mesh restrictions are nested within the supermesh restriction. Then there exist local interpolation matrices $(R_e^\ell)^T$ and $(R_e^{\ell-1})^T$ such that*

$$(4.24) \quad M_e^\ell = (R_e^\ell)^T M_e^S R_e^\ell, \quad M_e^{\ell-1} = (R_e^{\ell-1})^T M_e^S R_e^{\ell-1}, \quad M_e^{\ell, \ell-1} = (R_e^\ell)^T M_e^S R_e^{\ell-1}.$$

Proof. Let $l \in \{\ell, \ell-1\}$. The proof for the first two equations in (4.24) follows from the first part of the proof of Lemma 4.2 by replacing $\ell-1$ with l and ℓ with S . This argument gives us that

$$(4.25) \quad \phi^l = (R_e^l)^T \phi^S,$$

from which we also obtain the last relation in (4.24) since

$$(4.26) \quad M_e^{\ell, \ell-1} = \int_e \phi^\ell (\phi^{\ell-1})^T dx = (R_e^\ell)^T \int_e \phi^S (\phi^S)^T dx R_e^{\ell-1} = (R_e^\ell)^T M_e^S R_e^{\ell-1}. \quad \blacksquare$$

By using this result and the strategy highlighted in Remark 4.2, we can sample \mathbf{b}_e^l for $l \in \{\ell, \ell - 1\}$ by computing

$$(4.27) \quad \mathbf{b}_e^l = (R_e^l)^T H_r |e_r|^{-1/2} |e|^{1/2} \mathbf{z}_e \quad \text{with} \quad \mathbf{z}_e \sim \mathcal{N}(0, I)$$

since Remark 4.2 yields the relation $M_e^S/|e| = H_r H_r^T/|e_r| = \text{const}$, where H_r is the Cholesky factor of the local mass matrix over the reference element e_r . Note that H_r has to be computed only once. The advantage of performing this operation is that it avoids the assembly and factorization of each supermesh element local mass matrix.

Remark 4.4 (simpler cases: nested meshes and p -refinement). In the case in which the meshes of the MLMC hierarchy are nested, everything discussed is still valid by taking the supermesh to be the finer of the two meshes that define the MLMC level. In the case in which the MLMC hierarchy is constructed by using p -refinement, there is only one mesh in the hierarchy, and everything still applies by taking this mesh to be the “supermesh.” In both cases, a supermesh construction is not required in practice.

Remark 4.5. The coupling approach presented can also be used to couple the same white noise sample over the whole hierarchy of meshes. This enables the use of geometric full-multigrid [38] to solve the problem given by (3.7)–(3.8) on the finer grid with optimal multigrid complexity.

5. Numerical results. In this section we investigate the performance of the techniques presented. We consider the PDE

$$(5.1) \quad \begin{aligned} -\nabla \cdot (e^{u(x,\omega)} \nabla q(x, \omega)) &= 1, & x \in G = (-0.5, 0.5)^d, & \omega \in \Omega, \\ q(x, \omega) &= 0, & x \in \partial G, & \omega \in \Omega, \end{aligned}$$

where u is a Matérn field as given by (1.1) with mean and variance chosen so that $e^{u(x,\omega)}$ has mean 1 and standard deviation 0.2. We choose $D = (-1, 1)^d$ as the outer computational domain on which to solve (3.3). The output functional of interest we consider here is the $L^2(G)$ norm of q squared, namely $P(\omega) = \|q\|_{L^2(G)}^2(\omega)$.

We solve (3.3) and (5.1) with the FEniCS software package [28] and discretize the two problems by using continuous Lagrange finite elements of the same degree. For the linear solver, we use the BoomerAMG algebraic multigrid algorithm from Hypre [14] as a preconditioner and use the conjugate gradient routine of PETSc [3] for all equations. As a convergence criterion for the solver we require that the absolute size of the preconditioned residual norm be below a tolerance of 10^{-15} . We use the libsupermesh software package [29] for the supermesh constructions.

When using h -refinement, we construct the MLMC mesh hierarchies $\{D_h^\ell\}_{\ell=1}^L$ and $\{G_h^\ell\}_{\ell=1}^L$ in such a way that G_h^ℓ is embedded within D_h^ℓ for all ℓ , but $D_h^{\ell-1}$ and $G_h^{\ell-1}$ are not nested, respectively, within D_h^ℓ and G_h^ℓ for all $\ell > 1$. As the meshes are nonnested, a supermesh construction is required to couple each MLMC level. The mesh hierarchies we use are composed of $L = 9$ meshes in 2D and $L = 5$ meshes in 3D. The coarsest mesh in each hierarchy is uniform, while the other meshes are nonuniform and unstructured. Since the convergence behavior of the FEM is dependent on the quality of the mesh used, we try to obtain numerical results that are independent of grid quality by choosing meshes whose quality indicators do

not vary excessively throughout the hierarchies. Basic properties of the different meshes and number of elements of the constructed supermeshes are summarized in Tables 1 and 2. Note that the number of elements in the supermesh is, in practice, always bounded by a constant times the number of elements of the finer parent mesh. This constant is dimension dependent and larger in 3D than in 2D (cf. Table 2).

Table 1

Properties of the 2D mesh hierarchy: mesh level ℓ , maximal element size h_ℓ , number of elements n_ℓ , minimal and maximal element radius ratios RR_{\min} and RR_{\max} , and number of elements of the supermesh constructed using the meshes on levels ℓ and $\ell - 1$ as parent meshes n_{S_ℓ} . RR is computed as $d \times r_{\text{in}}^e / r_{\text{circ}}^e$, where r_{in}^e and r_{circ}^e are the in-radius and the circumradius of element e , respectively. Note that the element size roughly decreases proportional to $2^{-\ell}$.

ℓ (2D)	h_ℓ	n_ℓ	(RR_{\min}, RR_{\max})	n_{S_ℓ}/n_ℓ
1	0.707	32	(0.83, 0.83)	n/a
2	0.416	120	(0.61, 1)	2.03
3	0.194	500	(0.61, 1)	2.32
4	0.098	2106	(0.55, 1)	2.45
5	0.049	8468	(0.45, 1)	2.44
6	0.024	33686	(0.46, 1)	2.46
7	0.012	134170	(0.41, 1)	2.46
8	0.006	535350	(0.42, 1)	2.46
9	0.003	2143162	(0.42, 1)	2.47

Table 2

Properties of the 3D mesh hierarchy: mesh level ℓ , maximal element size h_ℓ , number of elements n_ℓ , minimal and maximal element radius ratios RR_{\min} and RR_{\max} , minimal and maximal element dihedral angles DA_{\min} and DA_{\max} , and number of elements of the supermesh constructed using the meshes on levels ℓ and $\ell - 1$ as parent meshes. Note that the element size of the last three levels roughly decreases proportional to $2^{-\ell}$.

ℓ (3D)	h_ℓ	n_ℓ	(RR_{\min}, RR_{\max})	(DA_{\min}, DA_{\max})	n_{S_ℓ}/n_ℓ
1	0.866	384	(0.72, 0.72)	(0.79, 1.57)	n/a
2	0.437	7141	(0.22, 1)	(0.21, 2.82)	17
3	0.280	22616	(0.18, 1)	(0.21, 2.83)	66
4	0.138	190081	(0.13, 1)	(0.21, 2.85)	42
5	0.070	1519884	(0.12, 1)	(0.21, 2.85)	46

When using p -refinement, we define the MLMC levels by taking the coarsest mesh in the 2D hierarchy and by increasing the polynomial degree of the FEM subspaces linearly so that $p_\ell = \ell$ for $\ell = 1, \dots, L$, with $L = 9$. We do not consider p -refinement in the 3D case as it does not offer any additional complications other than an increased computational cost.

5.1. Matérn field convergence. We first address the convergence of the solution of (3.3) to the Matérn field of interest. In practice, the exact solution of (3.3) is not known, so we consider instead the coupled equations (3.7) and (3.8). We monitor the quantities

$$(5.2) \quad \left| \mathbb{E} \left[\|u\|_{L^2(G)}^2 - \|u_{\ell-1}\|_{L^2(G)}^2 \right] \right|, \quad \mathbb{V} \left[\|u_\ell\|_{L^2(G)}^2 - \|u_{\ell-1}\|_{L^2(G)}^2 \right].$$

Note that the value of $\mathbb{E}[\|u\|_{L^2(G)}^2]$ is known up to the error introduced by truncating \mathbb{R}^d to D since we can exchange the order of expectation and integration,

$$(5.3) \quad \mathbb{E}[\|u\|_{L^2(G)}^2] = \mathbb{E}\left[\int_G u^2 \, dx\right] = \int_G \mathbb{E}[u^2] \, dx \approx \sigma^2 |G|,$$

where we have used the fact that $\mathbb{E}[u^2] \approx \sigma^2$ for all $x \in G$ (the relation only holds approximately due to domain truncation error).

The following result by Bolin, Kirchner, and Kovács [5] establishes a theoretical estimate for the expected convergence rates.

Theorem 5.1 (Theorem 2.10 and Corollary 2.4 in [5]). *Let u be the solution of (3.3) ($k = 1$ case), and let u_h be its FEM approximation obtained by using continuous Lagrange elements over a mesh of maximum element size h . Then there exist constants c_1 and c_2 such that*

$$(5.4) \quad \mathbb{E}[\|u - u_h\|_{L^2(D)}^2]^{1/2} \leq c_1 h^{2-d/2},$$

$$(5.5) \quad \left| \mathbb{E}[\|u\|_{L^2(D)}^2 - \|u_h\|_{L^2(D)}^2] \right| \leq c_2 h^{4-d}.$$

Note that the norms appearing in the error estimates of Theorem 5.1 refer to the outer domain D , while the norms we consider in (5.2) are taken over the inner domain G . As $G \subset D$, we expect to observe the same convergence behavior. We are not aware of any error estimates in the literature for the variance in (5.2), but the convergence order observed in practice is usually twice that of the expectation (see, for example, [9]), provided that the polynomial degree of the FEM basis is sufficiently high.

We consider the convergence behavior of the FEM approximation of the solution of (1.3) in the h -refinement case with the sampling strategy described in section 4. We fix $\lambda = 0.2$ and consider Matérn fields of smoothness $\nu = 1$, $\nu = 3$ ($k = 1$ and $k = 2$, respectively, in 2D) and $\nu = 1/2$ ($k = 1$ in 3D). For the $\nu = 1$ and $\nu = 1/2$ cases we use continuous piecewise linear (P1) elements, while for the $\nu = 3$ case we use continuous piecewise quadratic (P2) elements.

Since each sample drawn by solving (1.3) is computationally expensive, we are unable to take large numbers of samples as is generally done for 1D stochastic differential equations [20]. We therefore take $N_\ell = 5000$ Monte Carlo samples on all levels in 2D and $N_\ell = 1000$ samples in 3D. To verify that these numbers of samples are sufficient for accurate representation, we compute approximate 99.73% confidence intervals (CIs) for all the quantities of interest as $3\bar{\sigma}_\ell/\sqrt{N_\ell}$, where $\bar{\sigma}_\ell$ is the sample standard deviation of the output functional of interest on level ℓ . In all but one of the cases considered here, the FEM error dominates and the CIs are negligibly small (so small that they would not be visible on the convergence plots). The relatively small number of samples only becomes a problem in the $\nu = 3$ case, where the FEM convergence is much faster and the Monte Carlo error dominates. In this case, we replace the $\|u\|_{L^2(G)}$ term in the expectation in (5.2) with $\|u_\ell\|_{L^2(G)}$, and we instead monitor the convergence of the following quantity:

$$(5.6) \quad \left| \mathbb{E}[\|u_\ell\|_{L^2(G)}^2 - \|u_{\ell-1}\|_{L^2(G)}^2] \right|.$$

The advantage of doing this is that the variance of this error measure decreases with the level (see Figure 2), and 5000 samples are enough to obtain good accuracy.

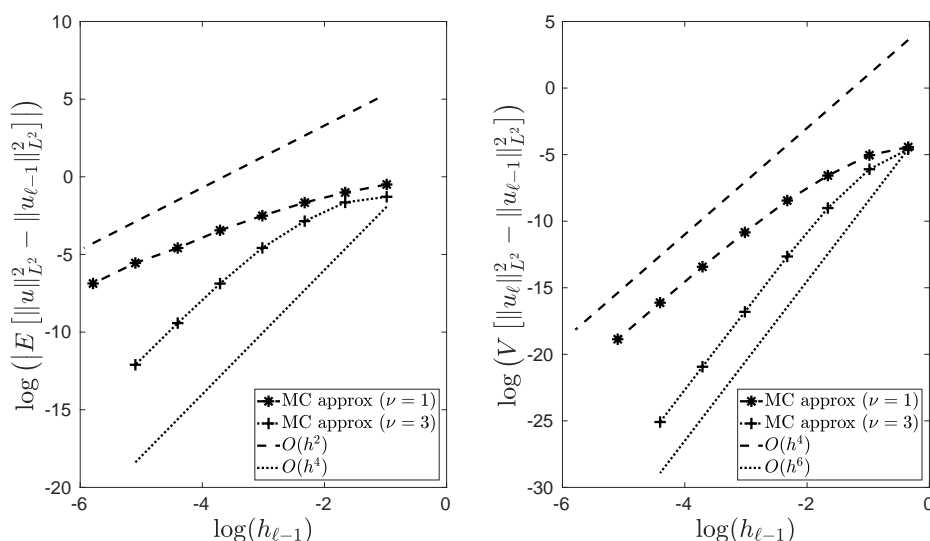


Figure 2. Convergence behavior of the FEM approximation to (1.3) with h -refinement in 2D. Plots show (the natural logarithm of) the expected value \mathbb{E} (left) and variance \mathbb{V} (right) versus maximal mesh size h_{ℓ} . For each level ℓ , the fields u_{ℓ} and $u_{\ell-1}$ have been sampled by coupling white noise realizations as described in section 4. As mentioned in the text, to compute the expected value in the $\nu = 3$ case we have replaced $\|u\|_{L^2(G)}$ with $\|u_{\ell}\|_{L^2(G)}$.

Results are shown in Figures 2 (2D) and 3 (3D). For both the 2D and 3D experiments, we observe the theoretically predicted convergence rates in terms of the mesh size (after a preasymptotic regime). However, we note how convergence is less regular than expected (especially in the 3D case) because of the unstructured meshes employed. This behavior does not appear when uniform meshes are used (not shown). Apart from the $\nu = 3$ case, the convergence order of the variance seems to be twice the convergence order of the expectation. In the $\nu = 3$ case, we observe order 6 for the variance with P2 elements (Figure 2) and order 8 with P3 elements (not shown). We conjecture that the variance convergence order is bounded by $2(p+1)$, where p is the polynomial degree of the FEM basis functions.

In Figure 4, we demonstrate how the Matérn covariance and the field coupling are correctly enforced by our technique. We compare the covariances of the coupled Matérn fields obtained by solving (3.7) and (3.8) on the finest level of the MLMC hierarchy with the exact Matérn covariance given by (1.1). The estimated covariances closely match each other and the exact expression, demonstrating that our coupling technique is accurate also in practice.

As a final verification step, we check that the coupled fields are consistent with the telescoping sum in (2.2) and (2.3); i.e., if we let a , b , c be the Monte Carlo approximations of $\mathbb{E}[\|u_{\ell}\|_{L^2(G)}^2 - \|u_{\ell-1}\|_{L^2(G)}^2]$, $\mathbb{E}[\|u_{\ell}\|_{L^2(G)}^2]$, and $\mathbb{E}[\|u_{\ell-1}\|_{L^2(G)}^2]$, respectively, we aim to verify that

$$(5.7) \quad a - b + c \approx 0,$$

at least to within the Monte Carlo accuracy. In Figure 5, we plot the quantity

$$(5.8) \quad T(a, b, c) \equiv \frac{|a - b + c|}{3(\sqrt{\mathbb{V}_a} + \sqrt{\mathbb{V}_b} + \sqrt{\mathbb{V}_c})}$$

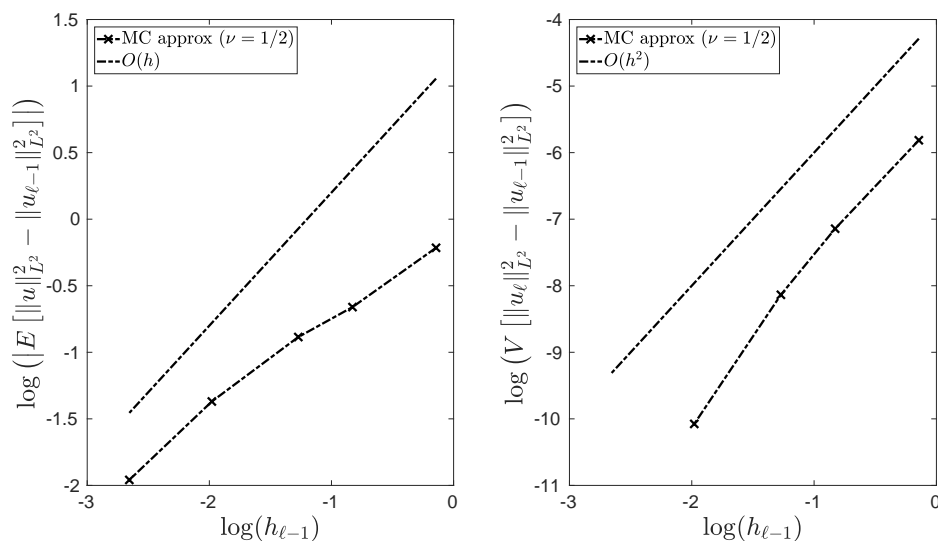


Figure 3. Convergence behavior of the FEM approximation to (1.3) with h -refinement in 3D. Plots show (the natural logarithm of) the expected value \mathbb{E} (left) and variance \mathbb{V} (right) versus maximal mesh size h_{ℓ} . The fields u_{ℓ} and $u_{\ell-1}$ have been sampled by coupling white noise realizations as described in section 4.

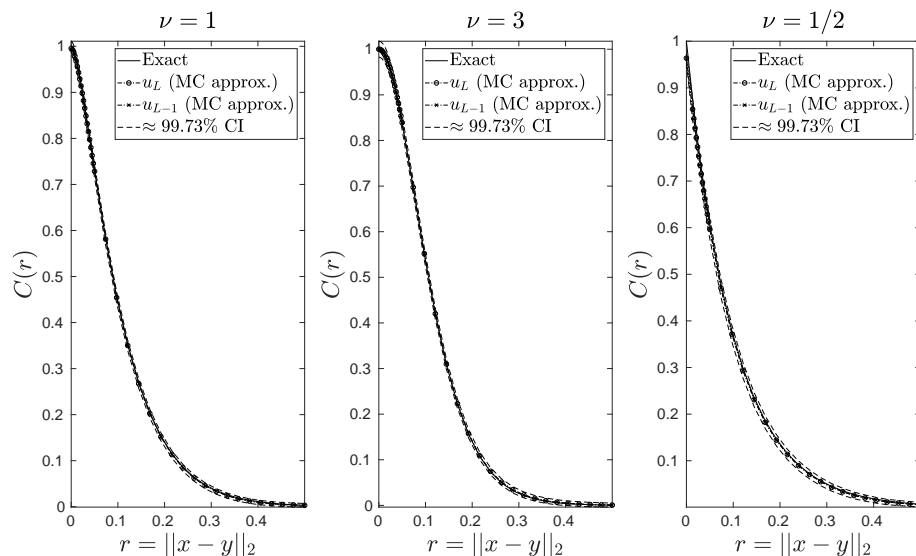


Figure 4. Plot of exact covariances and sample covariances versus distance r of the FEM solutions of (3.7) and (3.8) for three different values of ν in the h -refinement case. The exact covariance $C(r)$ is given by (1.1). For the $\nu = 3$ case, an extra elliptic PDE solve is needed; see (3.2).

for different levels and Matérn smoothness parameters ν , where \mathbb{V}_a , \mathbb{V}_b , and \mathbb{V}_c are the Monte Carlo approximations of the variances of a , b , and c . The probability of this ratio T being greater than 1 is less than 0.3% (for further details, see [20]). We observe that T ranges between

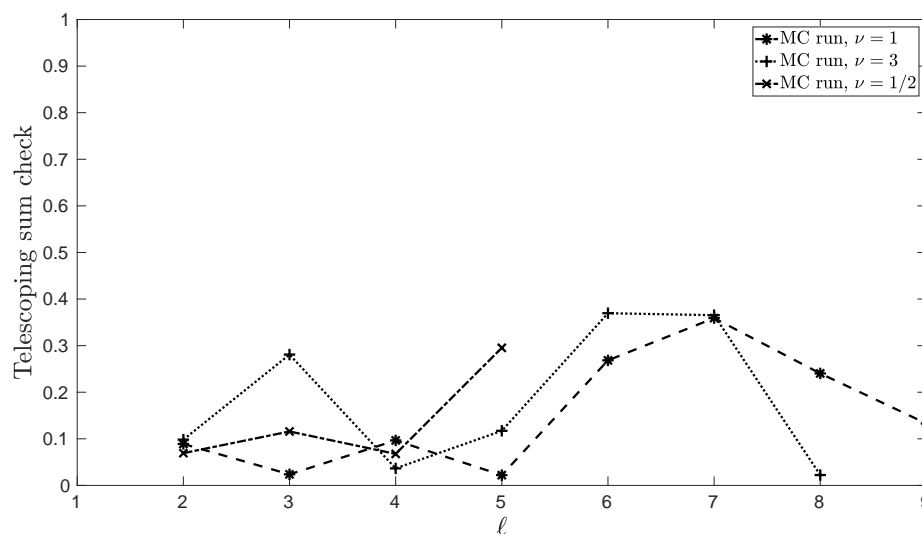


Figure 5. *Telescoping sum consistency check. Plot of $T(a, b, c)$ as defined by (5.8) versus level ℓ for $a = \mathbb{E}[\|u_\ell\|_{L^2(G)}^2 - \|u_{\ell-1}\|_{L^2(G)}^2]$, $b = \mathbb{E}[\|u_\ell\|_{L^2(G)}^2]$, and $c = \mathbb{E}[\|u_{\ell-1}\|_{L^2(G)}^2]$ for different smoothness parameters ν .*

0 and 0.4 for the levels and smoothness parameters tested (Figure 5) and in particular is well below 1. This indicates that our implementation of the MLMC algorithm correctly satisfies the telescoping summation formulation.

5.2. MLMC convergence. We now consider the convergence of the MLMC method applied to (5.1). In the case where u is sampled exactly, the assumptions of the MLMC convergence theorem (Theorem 2.2) hold for the h -refinement case with constants $\alpha = 2$ and $\beta = 4$ [8]. Furthermore, since we use multigrid to solve (5.1) and (3.3), we have $\gamma = d$. In the case where $\nu > 1$, the Matérn field smoothness increases [1], and we expect higher convergence rates for the solution of (5.1). For integer ν and exact sampling of u , if the domain G is of class $C^{\nu+1}$, then the MLMC parameter values are given by $\alpha = \min(\nu + 1, p + 1)$ and $\beta = 2\alpha$, where p is the polynomial degree of the Lagrange elements used [26].

In our case, u is approximated with the FEM, and this could affect convergence. To verify that this is not what happens in practice, we first solve (1.3) with FEM for the same parameter values as those in subsection 5.1, namely $\lambda = 0.2$, $\nu = 1$, and $\nu = 3$ ($k = 1$ and $k = 2$, respectively, in 2D) and $\nu = 1/2$ ($k = 1$ in 3D) using P1 elements for $\nu = 1/2$ and $\nu = 1$ and P2 elements for $\nu = 3$. We then use the approximated Matérn fields computed this way as coefficients in (5.1), which we solve again using the same choice of finite elements.

Results are shown in Figures 6 and 7. We observe that the convergence is unaffected by the approximation of the Matérn fields and that the estimated convergence orders agree with the theory [8]—apart from some discrepancies in the 3D case. This irregular behavior is probably due to the nonuniformity of the hierarchy (as we see from Table 2, the quality of the 3D meshes decreases with the level). This issue does not arise if the same numerical experiment is performed using a uniform hierarchy (hierarchy mesh sizes are given by $h_\ell = 1.732 \times 2^{-\ell}$; see Figure 8).

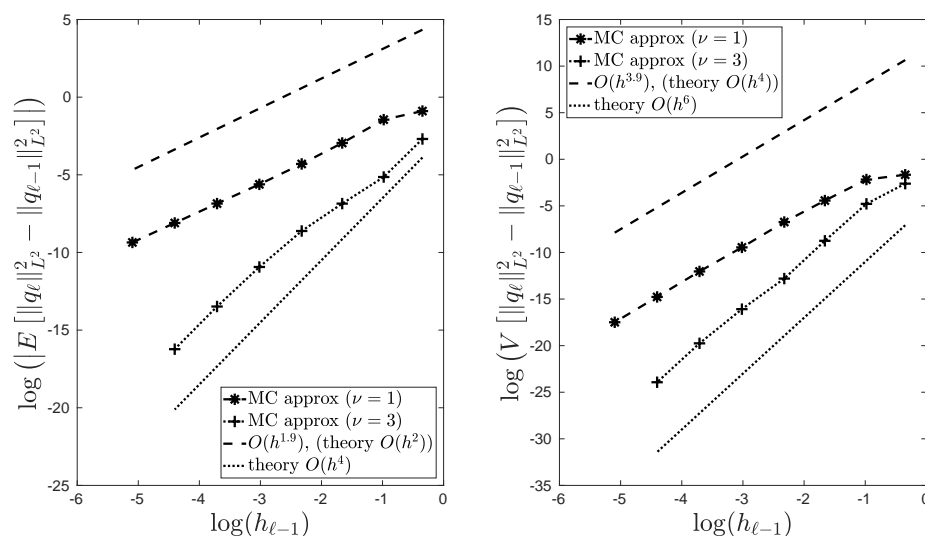


Figure 6. Convergence behavior of the FEM approximation to the solution of (5.1) with h -refinement in 2D. The estimated convergence orders agree with the theory [8, 26].

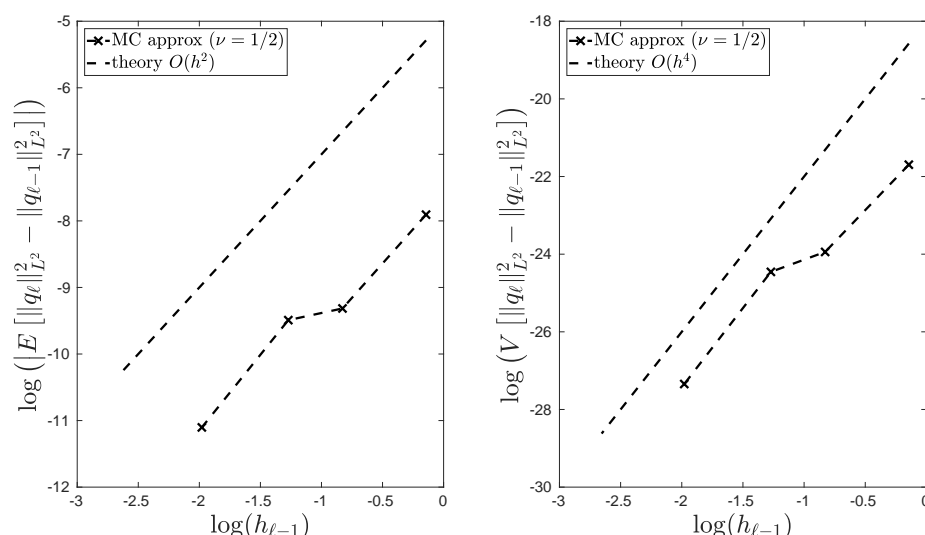


Figure 7. Convergence behavior of the FEM approximation to the solution of (5.1) with h -refinement in 3D.

We now investigate how MLMC performs in practice. We use standard Monte Carlo and MLMC to estimate $\mathbb{E}[\|q\|_{L^2(G)}]$ to the same accuracy for $\nu = 1$ (2D), P1 elements, and different error tolerances ε (cf. (2.4)). Again, the coefficient u of (5.1) is also approximated with the FEM. We keep track of the total computational cost C_{tot} and, in the MLMC case, of the number of samples N_ℓ taken on each level.

Results are shown in Figure 9. We observe that the number of levels used increases as the tolerance ε decreases (Figure 9 (left)). This behavior reflects the targeted weak error accuracy

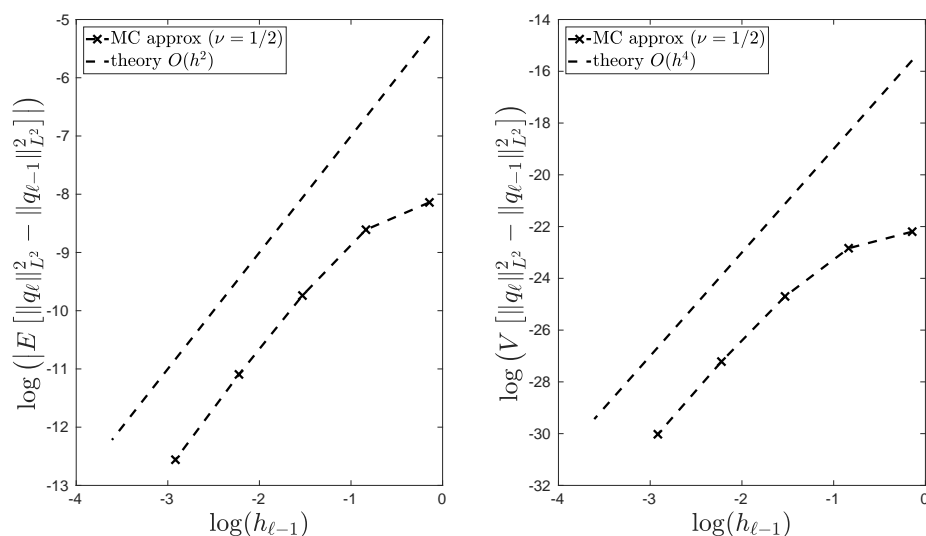


Figure 8. Convergence behavior of the FEM approximation to the solution of (5.1) with h -refinement in 3D using a hierarchy of uniform meshes. The estimated convergence orders agree with the theory [8]. The mesh sizes are given by $h_\ell = 1.732 \times 2^{-\ell}$.

[20]: the number of samples is chosen by the MLMC algorithm so as to optimize the total computational effort [19], and it decreases with the level, with many samples on the coarse levels and only a few on the fine levels. As $\beta > \gamma$ (cf. Theorem 2.2), we expect the total cost of the MLMC algorithm C_{tot} to be proportional to ε^{-2} . Figure 9 (right) shows $\varepsilon^2 C_{\text{tot}}$ versus ε , and indeed, we observe a near constant $\varepsilon^2 C_{\text{tot}}$ for the MLMC algorithm across multiple choices of ε . Figure 9 also compares the MLMC cost with the cost of obtaining an estimate of the same accuracy with standard Monte Carlo. We observe that the MLMC algorithm offers significant computational savings compared to standard Monte Carlo, with an improvement in the total cost C_{tot} of up to three orders of magnitude (Figure 9, right).

Finally, we consider the convergence of MLMC with p -refinement. We follow the same procedure as that in the h -refinement case and solve (5.1) after approximating the coefficient u by solving (1.3) with FEM. This time, however, we fix the mesh to be the coarsest mesh in the 2D hierarchy (cf. Table 1), and we consider a hierarchy of continuous piecewise polynomial elements of increasing polynomial degree $p = 1, \dots, 8$. We investigate the convergence behavior for different values of ν , namely $\nu \in \{1, 7, 31\}$ (corresponding to $k \in \{1, 3, 15\}$).

We observe in Figure 10 that convergence is geometric (the error decreases exponentially as the polynomial degree p grows). The solution of (5.1) is actually almost surely not analytic, and we would therefore expect algebraic convergence (i.e., the error decreases polynomially as p grows) [2]. We hypothesize that this better-than-expected convergence is in fact preasymptotic behavior and that the geometric convergence will eventually plateau and switch to a slower algebraic rate that depends on the smoothness of u (the larger ν , the faster the convergence) [21, 35]. However, apart from the $\nu = 1$ case for which the convergence plot begins to tail off, this is not observed for the polynomial degrees considered. We note that the larger the smoothness parameter ν , the faster the convergence of the expected value. The variance convergence order, on the other hand, seems to be unaffected by the value of ν .

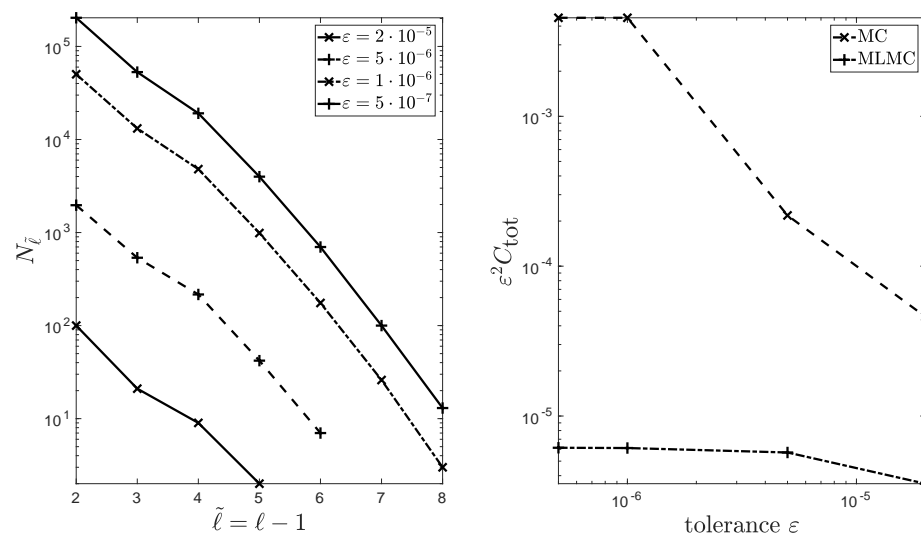


Figure 9. MLMC convergence for the solution of (5.1). In the plot on the left we show how the MLMC algorithm automatically selects the optimal number of samples $N_{\tilde{\ell}}$ on each level to achieve a given tolerance ε . Note that the MLMC routine uses the second mesh in the hierarchy described in Table 1 to define the first level $\tilde{\ell}$. The first mesh in Table 1 is dropped since it is too coarse, and it would not bring any significant advantage to the performance of MLMC [20]. In the plot on the right we compare the efficiency of MLMC with standard Monte Carlo for different tolerances. The savings of MLMC with respect to standard Monte Carlo are considerable.

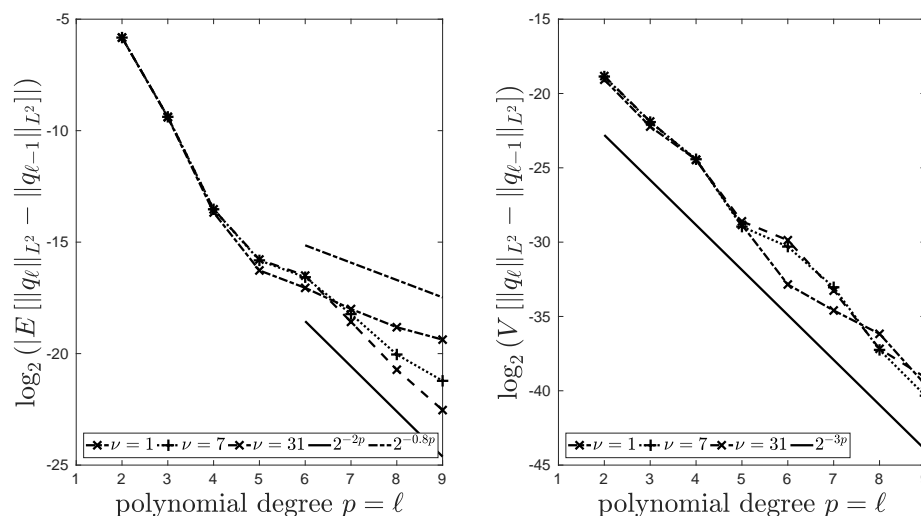


Figure 10. Convergence behavior of the FEM approximation to the solution of (5.1) with p -refinement in 2D. The approximate FEM solution q_{ℓ} on level ℓ is obtained by using Lagrange elements of degree $p = \ell$. For the polynomial degrees considered we are only able to observe a preasymptotic behavior, in which the convergence is geometric. The dashed and solid lines in the left plot indicate the estimated convergence order of the expected value (for $\nu = 1$ and $\nu = 31$, respectively). The solid line in the right plot indicates the estimated convergence order of the variance for all the values of ν considered.

6. Conclusions. In this work, we have presented a new sampling technique for efficient computation of the action of white noise realizations—even when coupled samples are required within an MLMC framework. This technique applies for general L^2 -conforming finite element spaces, and it allows the coupling of samples between nonnested meshes without resorting to a computationally costly interpolation or projection step. The numerical results show that our technique works well in practice: the convergence orders observed agree with existing theory, the number of supermesh elements grows linearly with the finer parent mesh size, the covariance structure of the sampled fields converges to the exact Matérn covariance, and the consistency of the telescoping sum is respected. As a concluding remark, we note that our sampling technique is not limited to Matérn field sampling but extends naturally to any application in which spatial white noise realizations are needed within a finite element framework.

Acknowledgments. The authors would like to acknowledge useful discussions with Endre Süli, Andrew J. Wathen, Abdul-Lateef Haji-Ali, and Alberto Paganini. The authors would also like to express their thanks to James R. Maddison for his assistance with the implementation of an interface between libsupermesh and FEniCS.

REFERENCES

- [1] P. ABRAHAMSEN, *A Review of Gaussian Random Fields and Correlation Functions*, 2nd ed., Norwegian Computing Center, 1997.
- [2] I. BABUŠKA AND M. SURI, *The p and h - p versions of the finite element method, basic principles and properties*, SIAM Rev., 36 (1994), pp. 578–632, <https://doi.org/10.1137/1036141>.
- [3] S. BALAY, S. ABHYANKAR, M. ADAMS, J. BROWN, P. BRUNE, K. BUSCHELMAN, L. DALCIN, V. EIJKHOUT, W. GROPP, D. KAUSHIK, M. G. KNEPLEY, D. MAY, L. CURFMAN MCINNES, T. MUNSON, K. RUPP, P. SANAN, B. SMITH, S. ZAMPINI, H. ZHANG, AND H. ZHANG, *PETSc Users Manual Revision 3.8*, Tech. report ANL-95/11, Argonne National Laboratory (ANL), 2017.
- [4] G. BENFATTO, G. GALLAVOTTI, AND F. NICOLÒ, *Elliptic equations and Gaussian processes*, J. Funct. Anal., 36 (1980), pp. 343–400.
- [5] D. BOLIN, K. KIRCHNER, AND M. KOVÁCS, *Numerical Solution of Fractional Elliptic Stochastic PDEs with Spatial White Noise*, preprint, <https://arxiv.org/abs/1705.06565>, 2018.
- [6] D. BOLIN AND F. LINDGREN, *Spatial models generated by nested stochastic partial differential equations, with an application to global ozone mapping*, Ann. Appl. Stat., 5 (2011), pp. 523–550.
- [7] R. BUCKDAHN AND E. PARDOUX, *Monotonicity methods for white noise driven quasi-linear SPDEs*, in Diffusion Processes and Related Problems in Analysis, Vol. 1, Progr. Probab. 22, Birkhäuser, 1990, pp. 219–233.
- [8] J. CHARRIER, R. SCHEICHL, AND A. L. TECKENTRUP, *Finite element error analysis of elliptic PDEs with random coefficients and its application to multilevel Monte Carlo methods*, SIAM J. Numer. Anal., 51 (2013), pp. 322–352, <https://doi.org/10.1137/110853054>.
- [9] K. A. CLIFFE, M. B. GILES, R. SCHEICHL, AND A. L. TECKENTRUP, *Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients*, Comput. Vis. Sci., 14 (2011), pp. 3–15.
- [10] C. R. DIETRICH AND G. N. NEWSAM, *Fast and exact simulation of stationary Gaussian processes through circulant embedding of the covariance matrix*, SIAM J. Sci. Comput., 18 (1997), pp. 1088–1107, <https://doi.org/10.1137/S1064827592240555>.
- [11] D. DRZISGA, B. GMEINER, U. RÜDE, R. SCHEICHL, AND B. WOHLMUTH, *Scheduling massively parallel multigrid for multilevel Monte Carlo methods*, SIAM J. Sci. Comput., 39 (2017), pp. S873–S897, <https://doi.org/10.1137/16M1083591>.
- [12] Q. DU AND T. ZHANG, *Numerical approximation of some linear stochastic partial differential equations driven by special additive noises*, SIAM J. Numer. Anal., 40 (2002), pp. 1421–1445, <https://doi.org/10.1137/S0036142901387956>.
- [13] H. C. ELMAN, D. J. SILVESTER, AND A. J. WATHEN, *Finite Elements and Fast Iterative Solvers: With*

- Applications in Incompressible Fluid Dynamics*, 2nd ed., Oxford University Press, 2014.
- [14] R. D. FALGOUT AND U. M. YANG, *Hypre: A library of high performance preconditioners*, in Proceedings of the International Conference on Computational Science (ICCS 2002), Lecture Notes in Comput. Sci. 2331, Springer, 2002, pp. 632–641.
 - [15] P. E. FARRELL, *Galerkin Projection of Discrete Fields Via Supermesh Construction*, Ph.D. thesis, Imperial College London, 2009.
 - [16] P. E. FARRELL AND J. R. MADDISON, *Conservative interpolation between volume meshes by local Galerkin projection*, Comput. Methods Appl. Mech. Engrg., 200 (2011), pp. 89–100.
 - [17] P. E. FARRELL, M. D. PIGGOTT, C. C. PAIN, G. J. GORMAN, AND C. R. WILSON, *Conservative interpolation between unstructured meshes via supermesh construction*, Comput. Methods Appl. Mech. Engrg., 198 (2009), pp. 2632–2642.
 - [18] C. GEUZAIN AND J.-F. REMACLE, *Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities*, Internat. J. Numer. Methods Engrg., 79 (2009), pp. 1309–1331.
 - [19] M. B. GILES, *Multilevel Monte Carlo path simulation*, Oper. Res., 56 (2008), pp. 607–617.
 - [20] M. B. GILES, *Multilevel Monte Carlo methods*, Acta Numer., 24 (2015), pp. 259–328.
 - [21] W. HACKBUSCH, *Elliptic Differential Equations: Theory and Numerical Treatment*, Springer-Verlag, 1992.
 - [22] H. HARBRECHT, M. PETERS, AND R. SCHNEIDER, *On the low-rank approximation by the pivoted Cholesky decomposition*, Appl. Numer. Math., 62 (2012), pp. 428–440.
 - [23] H. HARBRECHT, M. D. PETERS, AND M. SCHMIDLIN, *Uncertainty quantification for PDEs with anisotropic random diffusion*, SIAM J. Numer. Anal., 55 (2017), pp. 1002–1023, <https://doi.org/10.1137/16M1085760>.
 - [24] T. HIDA, H.-H. KUO, J. POTTHOFF, AND L. STREIT, *White Noise: An Infinite Dimensional Calculus*, Springer Science+Business Media, 1993.
 - [25] K. ITÔ, *Stationary random distributions*, Mem. Coll. Sci. Univ. Kyoto Ser. A Math., 28 (1954), pp. 209–223.
 - [26] X. LI AND J. LIU, *A Multilevel Approach Towards Unbiased Sampling of Random Elliptic Partial Differential Equations*, preprint, <https://arxiv.org/abs/1605.06349>, 2016.
 - [27] F. LINDGREN, H. RUE, AND J. LINDSTRÖM, *An explicit link between Gaussian fields and Gaussian Markov random fields: The stochastic partial differential equation approach*, J. R. Stat. Soc. Ser. B Stat. Methodol., 73 (2011), pp. 423–498.
 - [28] A. LOGG, K.-A. MARDAL, AND G. N. WELLS, EDS., *Automated Solution of Differential Equations by the Finite Element Method. The FEniCS Book*, Lecture Notes in Comput. Sci. Eng. 84, Springer, 2012.
 - [29] J. R. MADDISON, P. E. FARRELL, AND I. PANOURGIAS, *Parallel Supermeshing for Multimesh Modelling*, Tech. report eCSE03-08, Archer, 2013.
 - [30] S. OSBORN, P. S. VASSILEVSKI, AND U. VILLA, *A Multilevel, Hierarchical Sampling Technique for Spatially Correlated Random Fields*, preprint, <https://arxiv.org/abs/1703.08498>, 2017.
 - [31] S. OSBORN, P. ZULIAN, T. BENSON, U. VILLA, R. KRAUSE, AND P. S. VASSILEVSKI, *Scalable Hierarchical PDE Sampler for Generating Spatially Correlated Random Fields using Non-Matching Meshes*, preprint, <https://arxiv.org/abs/arXiv:1712.06758>, 2017.
 - [32] P. WHITTLE, *On stationary processes in the plane*, Biometrika, 41 (1954), pp. 434–449.
 - [33] R. POTSEPAEV AND C. L. FARMER, *Application of stochastic partial differential equations to reservoir property modelling*, in Proceedings of the 12th European Conference on the Mathematics of Oil Recovery (ECMOR XII), Vol. 2, EAGE Publications, 2010.
 - [34] R. SCHEICHL, *Personal communication*, 2017.
 - [35] C. SCHWAB, *p- and hp-Finite Element Methods. Theory and Applications in Solid and Fluid Mechanics*, Clarendon Press, 1998.
 - [36] G. STRANG, *Introduction to Linear Algebra*, 5th ed., Wellesley-Cambridge Press, 2016.
 - [37] A. L. TECKENTRUP, R. SCHEICHL, M. B. GILES, AND E. ULLMANN, *Further analysis of multilevel Monte Carlo methods for elliptic PDEs with random coefficients*, Numer. Math., 125 (2013), pp. 569–600.
 - [38] U. TROTTEBERG, C. W. OOSTERLEE, AND A. SCHULLER, *Multigrid*, Academic Press, 2000.
 - [39] A. J. WATHEN, *Realistic eigenvalue bounds for the Galerkin mass matrix*, IMA J. Numer. Anal., 7 (1987), pp. 449–457.
 - [40] Z. ZHANG, B. ROZOVSKII, AND G. E. KARNIADAKIS, *Strong and weak convergence order of finite element methods for stochastic PDEs with spatial white noise*, Numer. Math., 134 (2016), pp. 61–89.