A FULLY ADAPTIVE INTERPOLATED STOCHASTIC SAMPLING METHOD FOR LINEAR RANDOM PDES

FELIX ANKER[1], CHRISTIAN BAYER, MARTIN EIGEL, JOHANNES NEUMANN[2], AND JOHN SCHOENMAKERS

Abstract. A numerical method for the fully adaptive sampling and interpolation of linear PDEs with random data is presented. It is based on the idea that the solution of the PDE with stochastic data can be represented as conditional expectation of a functional of a corresponding stochastic differential equation (SDE). The spatial domain is decomposed by a non-uniform grid and a classical Euler scheme is employed to approximately solve the SDE at grid vertices. Interpolation with a conforming finite element basis is employed to reconstruct a global solution of the problem. An a posteriori error estimator is introduced which provides a measure of the different error contributions. This facilitates the formulation of an adaptive algorithm to control the overall error by either reducing the stochastic error by locally evaluating more samples, or the approximation error by locally refining the underlying mesh. Numerical examples illustrate the performance of the presented novel method.

1. Introduction

It becomes increasingly common that problems in the applied sciences, e.g. in engineering and computational biology, involve uncertainties of model parameters. These can for instance be related to coefficients of random media, i.e. material properties, inexact domains and stochastic boundary data. The uncertainties may result from heterogeneities and incomplete knowledge or inherent stochasticity of parameters. With steadily increasing computing power, the research field of uncertainty quantification (UQ) has become a rapidly growing and vividly active area of research which covers many aspects of dealing with such uncertainties for problems of practical interest.

In this work, we derive a novel adaptive numerical approach for the solution of linear PDEs with stochastic data. The proposed method is based on the presentation in [1] where a similar idea was described in combination with a global regression and without adaptivity (except for the step width in the Euler scheme). The important topic of adaptivity is picked up in this work where a *unique feature of the method is exploited*, namely the completely decoupled and localized parametrization (and thus control) of approximation and stochastic errors. This becomes feasible since

- the pointwise solution in the spatial domain is determined by an appropriate SDE and solved by an adaptive Euler scheme as in [1],

2010 *Mathematics Subject Classification.* 35R60, 47B80, 60H35, 65C20, 65N12, 65N22, 65J10,65C05 .

Key words and phrases. random PDE, stochastic differential equation, Feynman-Kac, interpolation, finite element, a posteriori error estimator, adaptive method, Euler Maruyama.

[1] Research of F. Anker has been supported by grant Jo329/10-2 within the DFG priority programme 1679: Dynamic simulation of interconnected solids processes.

[2] Research of J. Neumann was funded in part by the DFG Matheon project SE13.

- the global solution on the spatial domain is reconstructed based on a triangulation and interpolation in between vertices (i.e. on a discrete finite element space).

Consequently, the overall error is determined by

(i) the accuracy of the single SDE solutions dictated by the step width of the scheme,

(ii) the stochastic error for the pointwise expected value determined by the number of solutions computed on each vertex,

(iii) the number of solution points in the spatial domain determined by the refinement level of the employed mesh.

Hence, the second item related to the stochastic error enables a localized adaptivity for the sampling while the first and third items are related to the approximation quality in the spatial domain. In comparison to other common methods such as the Monte Carlo FEM for stochastic PDEs [2], the proposed method provides means to *adjust the number of samples locally* while also reconstructing global solutions. Moreover, it is also highly parallelizable and thus well suited for modern distributed multi-core architectures.

While the derivation of the method in the next section is rather general, a specific motivation is given by a model relevant in practical problems, namely the Darcy equation related to the modeling of groundwater flow. It reads as

$$-\nabla \cdot (\kappa(x)\nabla u(x)) = f(x), \quad x \in D, \tag{1a}$$

$$u(x) = g(x), \quad x \in \partial D, \tag{1b}$$

where the solution u is the hydraulic head, κ denotes the conductivity coefficient describing the porosity of the medium, f is a source term and the Dirichlet boundary data is defined by g. The computational domain in d dimensions is denoted $D \subset \mathbb{R}^d$ and we suppose that D is a convex polygon. Moreover, all data are supposed to be sufficiently smooth such that the problem always exhibits a unique solution which then is also smooth. A detailed regularity analysis is not in the scope of this paper. In principle, although we restrict our investigations to a stochastic coefficient κ, any data of the PDE can be modeled as being stochastic. The model (1) is quite popular for analytical and numerical examinations since it is one of the simplest models which reveals some major difficulties that also arise in more complex stochastic models. Moreover, the deterministic linear second order elliptic PDE is a well-studied model problem and it is of practical relevance, e.g. in the context of groundwater contamination.

The random data used in the PDE model is a stochastic field given with an adequate representation. This can for instance be based on actual measurements, expert-knowledge or simplifying assumptions regarding the statistics. For numerical computations, the representation has to be amenable for the employed method. However, the derived method does not rely on a specific representation of the random fields. In particular, only the generation of realizations is required to carry out numerical computations.

A variety of numerical methods is available to obtain approximate solutions of the model problem (1) with random data and we only refer to [3, 4, 5] for an overview in the context of uncertainty quantification (UQ). These methods often rely on the separation of the deterministic and the stochastic space and introduce separate discretizations [6]. Common methods are based on sampling of the stochastic space, the projection onto an appropriate stochastic basis or a perturbation analysis. The most popular sampling approach is the Monte Carlo (MC) method which is very robust and easy to implement. Recent developments include the quite successful application of multilevel ideas for variance reduction and advances with structured point sequences (Quasi-MC), cf. [2, 7, 8, 9, 10]. (Pseudo-)Spectral methods represent a popular class of projection techniques which can e.g. be based on interpolation (Stochastic Collocation) [11, 12, 13] or orthogonal projections with respect to the energy norm induced by the differential operator of the random PDE (Stochastic Galerkin FEM) [14, 15, 16, 17, 18, 19, 20]. These methods are more involved to analyze and implement but offer the benefit of possibly drastically improved convergence rates when compared to standard Monte Carlo sampling. The deterministic discretization often relies on the finite element method (FEM) which also is employed with MC.

The aim of this paper is the description of a novel highly adaptive numerical approach which is founded on the (classical) observation (see [21, 1]) that the random PDE (1) is directly related to an SDE driven by a stochastic process, namely

$$\mathrm{d}X_t = b(X_t)\ \mathrm{d}t + \sigma(X_t)\ \mathrm{d}W_t, \tag{2}$$

with appropriate coefficients b and σ, Brownian motion W and additional boundary conditions. For deterministic data κ, f, g, for any $x \in D$, the Feynman-Kac formula leads to a collection of random variables $\varphi^x = \varphi^x(\kappa, f, g)$ such that $u(x) = E[\varphi^x]$, i.e. the deterministic solution at x is equivalent to the expectation of the random variable. When the data are stochastic, the solution $u(x)$ of the random PDE at $x \in D$ can be expressed as the *conditional expectation* of u given data κ, f, g, i.e., $u(x) = E[\varphi^x \mid \kappa, f, g]$, and the variance of $u(x)$ can be bounded by the variance of φ^x. To determine φ^x at points $x \in D$, a classical Euler method can be employed. In order to recover a global solution in the spatial domain D, opposite to the previous work [1] where global regression was utilized, we here rely on a mesh $\mathcal{T}$ which is a regular triangulation of D. Sampled approximations of φ^x are then computed at the nodes of the mesh and the values are used for an interpolation in a discrete finite element space. This yields the approximate expectation of the solution $E[u(\omega, \cdot)]$ defined on the entire domain D.

A distinct advantage of this approach is the separation of all error components as mentioned above, which in the case of the discrete interpolation allows for the application of simple finite element (FE) a posteriori error estimates to refine the spatial mesh, i.e. the location of sample points in the domain guided by the global approximation error.

One can regard the proposed method as a combination of sampling and interpolation methods, that make use of classical stochastic solution techniques pointwise and a global interpolation with FE basis functions. When compared to MC which samples a stochastic space $(\Omega, \mathcal{F}, P)$ by (typically) determining a FE solution at every point and subsequently averaging the solutions, our method determines realizations of stochastic solutions at points in the spatial domain D and determines an approximation of the expectation by a global interpolation in the physical space. Thus, the method does not require any type of global deterministic solver and can be parallelized extremely well.

To the best of our knowledge, the regression/interpolation approach based on the Feynman-Kac representation proposed in [1] and the present paper is new for the field of PDEs with stochastic coefficients. Of course, similar methods have a long history for deterministic PDEs, and we refer to the monograph [21] for a thorough overview. There have also been some works on numerics for SPDEs in the sense of time-dependent PDEs driven by temporal (Brownian) noise. In particular, we refer to [22] for applications to stochastic Navier-Stokes equations. Due to the non-linearity, the Feynman-Kac representation cannot be directly applied in that problem, but a layer method based on linearized problems is constructed.

The structure of the paper is as follows: In Section 2, we elaborate on the representation of deterministic and stochastic PDEs in terms of stochastic differential equations (SDEs). Moreover, we recall the employed numerical methods to determine pointwise stochastic solutions, namely the Euler method and Monte Carlo sampling. Additionally, based on a deterministically chosen set of stochastic solutions in the spatial domain, the reconstruction of a global approximation by means of an interpolation in a discrete FE basis is described. This allows for a fully adaptive algorithm which is derived in Section 3. There, the different error components are identified and a practical approach for the determination of the discretization parameters is explained. The paper is concluded with several numerical examples in Section 4 where the performance of the new method is demonstrated.

2. Stochastic Sampling and Interpolation

In this section we provide a brief overview of the method presented in [1]. As a motivation to the approach, we want to construct an SDE such that the solution $u(\omega, x)$ of the SPDE at some point $x \in D$ can be expressed as conditional expectation of some functional of the solution of an appropriate SDE. This forms the basis of the method which is then extended to a fully adaptive scheme.

For the sake of concreteness, we present the method specifically in the case of Darcy's law and refer once more to [1] for the general form. For a more comprehensive treatment of the well-known basic stochastic theory, we refer the reader for instance to [21].

2.1. **Stochastic representations of PDEs.** The starting point is the following SDE

$$\mathrm{d}X_t = \nabla\kappa(X_t)\,\mathrm{d}t + \sqrt{2\kappa(X_t)}\,\mathrm{d}W_t, \qquad X_0 = x, \tag{3a}$$

where $x \in D \subset \mathbb{R}^d$ is a deterministic point, W is a d-dimensional standard Brownian motion defined on some probability space $(\Omega, \mathcal{F}, P)$ and $\kappa : \Omega \times D \to \mathbb{R}$ is the *stochastic* field of conductivity coefficients associated to the medium. Furthermore, it is important to require the stochastic field κ and the Brownian motion W to be independent.[1] Additionally, we shall sometimes consider the derived process

$$Z_t \coloneqq \int_0^t f(X_s)\,\mathrm{d}s, \tag{3b}$$

for the (again, possibly random, but independent from W) source term $f : \Omega \times D \to \mathbb{R}$. If we want to stress the dependence on the initial value, we write $X_t^x \coloneqq X_t$. The process Z depends on x as well and we shall write Z_t^x if we want to stress this dependence. Notice that existence and uniqueness of solutions to the SDE can be obtained simply by applying standard theory after conditioning on the random fields κ and f, provided that the fields are regular enough. In particular, we need to require that $\nabla\kappa$ exists and is Lipschitz continuous in x a.s. For more details on convenient regularity assumptions we again refer to [1].

The (random) Dirichlet problem (1) now admits the following representation in terms of the solutions of the above SDE: Let $\tau = \tau_x$ denote the first hitting time of the solution $X = X^x$ (started at $x \in \overline{D}$) at the boundary ∂D. Then the random solution $u(x)$ and its expectation $E[u(x)]$ satisfy the *Feynman-Kac theorem*:

$$u(x) = E\left[g\left(X_{\tau_x}^x\right) + Z_{\tau_x}^x \;\middle|\; \kappa,\, f\right], \qquad E[u(x)] = E\left[g\left(X_{\tau_x}^x\right) + Z_{\tau_x}^x\right]. \tag{4}$$

This means that the random solution $u(x)$ is obtained from the random variable $g\left(X_{\tau_x}^x\right) + Z_{\tau_x}^x$ by taking expectations only w.r.t. the Brownian motion W. On the other hand, for $E\left[u(x)\right]$, we simply take the total expectation as there is no need for iterating the expectations.

[1] This may require to enhance the original probability space Ω, on which κ is defined. See [1] for details.

For a better understanding of (4), let us note that the result is classical[2] in the special case of deterministic coefficients κ and f, see, for instance, [21]. Of course, in that case the conditioning is trivial. As the coefficients κ and f are assumed to be independent from the Brownian motion W driving the SDE (3), the stochastic representation easily extends from the deterministic to the random case in the sense of (4), provided that the regularity conditions for the deterministic case hold "uniformly in the randomness induced by κ, f". Specifically, if we are interested in the solution $u = u(\omega)$ at a *fixed* realization $\kappa(\cdot, \omega)$ and $f(\cdot, \omega)$ of the random field, we can simply apply the Feynman-Kac formula for the case of deterministic coefficients $\kappa(\cdot, \omega)$ and $f(\cdot, \omega)$. Hence, (4) is an immediate consequence of the Feynman-Kac formula for deterministic coefficients.

Equation (4) can also be used to estimate the variance of the random solution u, since

$$\mathrm{Var}\left[u(x)\right] \leq \mathrm{Var}\left[g\left(X^x_{\tau_x}\right) + Z^x_{\tau_x}\right].$$

This inequality is a simple consequence of the law of total variance. Note, however, that the inequality is usually strict. Intuitively, the difference of the right-hand side and the left-hand side corresponds to the variance induced by the Brownian motion W.

Hence, for any $x \in D$ for which we want to obtain the solution $E[u(x)]$ of (1), we have to compute a solution of the SDE (3). We thus have to solve two problems,

(i) Find an approximation $\overline{X}_N$ of X_t, which is actually computable.

(ii) Given such an approximation, the sought solution $E\left[g\left(\overline{X}_{\overline{\tau}}\right) + \overline{Z}_{\overline{\tau}}\right]$ is computed by a (quasi) Monte Carlo method, where over-lined expressions denote computable approximations.

Remark 2.1. As indicated above for the variance, higher moments of the stochastic representation $g\left(X^x_{\tau_x}\right) + Z^x_{\tau_x}$ generally only give upper bounds for the corresponding moments of the solution u as a simple consequence of Jensen's inequality. Hence, the method presented in this work can only be used to compute upper bounds of higher moments of u.

2.2. **Discretization of the SDE.** Clearly the most popular approximation method for SDEs is a straight-forward generalization of the Euler scheme for ODEs. Indeed, let $0 = t_0 < \cdots < t_N = t$ be a time grid, set

$$\Delta t_i \coloneqq t_i - t_{i-1}, \quad \Delta W_i \coloneqq W_{t_i} - W_{t_{i-1}}, \quad \Delta t_{\max} \coloneqq \max_i \Delta t_i, \quad \overline{X}_0 \coloneqq x,$$

and iteratively define as a discretization of (2)

$$\overline{X}_i \coloneqq \overline{X}_{i-1} + b\left(\overline{X}_{i-1}\right)\Delta t_i + \sigma\left(\overline{X}_{i-1}\right)\Delta W_i, \quad i = 1, \ldots, N. \tag{5}$$

Under weak assumptions we have *strong* convergence with rate 1/2, i.e.,

$$\mathrm{E}\left[\left|X_t - \overline{X}_N\right|\right] \leq C\sqrt{\Delta t_{\max}}$$

for some constant C independent of $\Delta t_{\max}$. More relevant in most applications is the concept of *weak* approximation. Fortunately, the Euler scheme typically exhibits first order weak convergence, i.e., for any suitable test function $F : \mathbb{R}^d \to \mathbb{R}$, it holds that

$$\left|\mathrm{E}[F(X_t)] - \mathrm{E}\left[F\left(\overline{X}_N\right)\right]\right| \leq C\Delta t_{\max} \tag{6}$$

with a constant C independent of $\Delta t_{\max}$ for fixed times t. However, the stochastic representation (4) involves the process stopped at the first hitting time of the boundary ∂D of the domain. This stopping time is approximated by the first exit time $\overline{\tau}$ of the discrete time process $\overline{X}_i$, $i = 0, \ldots, N$.

[2] For instance, when $\kappa, \nabla\kappa, f$ are Lipschitz, κ is strictly positive and D is a convex polygon, but these conditions can certainly be relaxed.

Note that there are two sources of errors in the approximation $\overline{X}_{\overline{\tau}}$ of X_τ:

(i) the error in the approximation of X by $\overline{X}$
(ii) the possibility that exit of X occurs between two grid points t_i and t_{i+1} while $\overline{X}$ does not exit–or vice versa.

Indeed, even if two paths X and $\overline{X}$ are close in, say, infinity norm, their exit times can be substantially different. Hence, the second source of error reduces the weak error rate, i.e., the approximation error for $E[u(x)]$ decreases to the rate $1/2$. For adaptive time-step refinements which can recover the order of convergence to an observed order 1 again, we refer to the references given in [1].

2.3. **Monte Carlo approximation of the expectation.**

Another step in the discretization is the approximation of the expected value as in (4) with a Monte Carlo estimator. Multilevel Monte Carlo methods are also possible but require the construction of two related realizations of $\overline{X}_M$ for each sample on the individual levels as described in [23]. Here, we restrict ourselves to the classical Monte Carlo estimator which is defined as

$$\mathrm{E}^{\mathrm{MS}}_{M,N}[u(x)] \coloneqq \frac{1}{N}\sum_{i=1}^{N} g\big(\overline{X}^x_{\overline{\tau}}(\omega_i)\big) + \overline{Z}^x_{\overline{\tau}}(\omega_i), \tag{7}$$

where $(\omega_i)_{i=1}^N$ is a set of $N \in \mathbb{N}$ samples drawn from the probability space $(\Omega, \mathcal{F}, P)$ and M describes the number of steps for the discrete diffusion process. Note that this random space represents both the Brownian motion W and the stochastic field κ from the initial Darcy problem (1). Recall that

$$\overline{Z}^x_\tau = \int_0^\tau f(X_s)\,\mathrm{d}s,$$

which we approximate by a simple left-point rule $\overline{Z}_{\overline{\tau}}$. Hence,

$$\begin{aligned} \overline{X}_0 &= x, \\ \overline{X}_m &= \overline{X}_{m-1} + \nabla\kappa(\overline{X}_{m-1})\Delta t_m + \sqrt{2\kappa(\overline{X}_{m-1})\Delta t_m}\;\Xi, \end{aligned} \tag{8}$$

where M is the last index with $\overline{X}_m \in D$ for $m = 0, \dots, M$, $\overline{X}_{M+1} \notin D$, and Ξ is a standard normal distributed random variable. Here, $\mathcal{N}_{0,1}$ denotes the standard normal distribution in d dimensions. The stopping position and the stopping time t_τ of the diffusion process is now approximated by the projection

$$\overline{X}_{\overline{\tau}} \coloneqq \arg\min\big\{\|x - \overline{X}_M\| \;\big|\; x \in \partial D, x = \overline{X}_M + s(\overline{X}_{M+1} - \overline{X}_M), s \geq 0\big\} \tag{9}$$

of $\overline{X}_M$ onto the boundary ∂D in direction of $\overline{X}_{M+1}$. The stopping time τ is approximated by $\overline{\tau} \coloneqq t_M$. Equation (7) provides the estimator with a simple one-point integration rule via

$$\mathrm{E}^{\mathrm{MS}}_{M,N}[u(x)] \coloneqq \frac{1}{N}\sum_{i=1}^{N}\left(g\big(\overline{X}_{\overline{\tau}}(\omega_i)\big) + \sum_{m\in(0,\dots,M,\overline{\tau})} f\big(\overline{X}_m(\omega_i)\big)\Delta t_m \right).$$

Elementary time adaptivity is applied by choosing $\Delta t_j = \mathrm{dist}(\partial D, \overline{X}_{j-1})\Delta t_0$ where $\mathrm{dist}(\partial D, x) \coloneqq \min\{\|x_d - x\| \mid x_d \in \partial D\}$ is the Euclidean distance to the boundary of the domain. The entire process is depicted in Algorithm 1 where a simple rectangle integration method is used. Moreover, Figure 1 sketches the described process.

Alongside the approximate expectation we can also estimate the sample variance of this approach at the location $x \in D$, since

$$\mathrm{Var}^{\mathrm{MS}}_{M,N}[u(x)] \coloneqq \frac{1}{N}\sum_{i=1}^{N}\left(g(\overline{X}_{\overline{\tau}}(\omega_i)) + \sum_{m\in(0,\ldots,M,\overline{\tau})} f\left(\overline{X}_m\left(\omega_i\right)\right)\Delta t_m - \mathrm{E}^{\mathrm{MS}}_{M,N}[u(x)]\right)^2 .$$

Note that this estimated variance has contributions from both the random field of the PDE and the SDE.

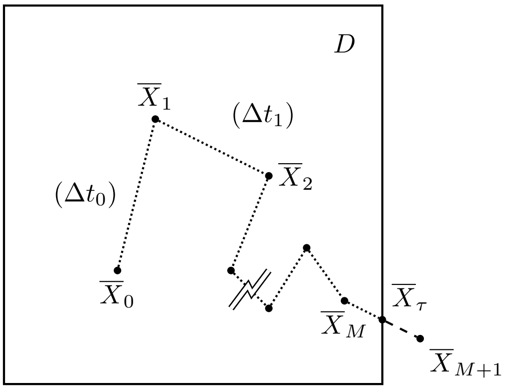


Figure 1. Sketch of a discrete diffusion process realization with endpoint projection and indicated step width.

```
In  : point $x \in D$, number of samples $N$, initial time step $\Delta t_0$
Out: $\mathrm{E}^{\mathrm{MS}}_{M,N}[u(x)]$, $\mathrm{Var}^{\mathrm{MS}}_{M,N}[u(x)]$

for $i = 1, \ldots, N$ do
    $\overline{X}_0 = x$
    $F = 0$
    $m = 1$
    sample $\kappa^i = \kappa(\omega_i)$ with $\omega_i \in \Omega$
    while $\overline{X}_m \in D$ do
        $F = F + f(\overline{X}_{m-1})\Delta t_{m-1}$
        $\Delta t_m = \min\left\{\mathrm{dist}(\partial D, \overline{X}), 1\right\}\Delta t_0$
        sample $\Xi$ from $\mathcal{N}_{0,1}$
        $\overline{X}_m = \overline{X}_{m-1} + \nabla\kappa^i(\overline{X}_{m-1})\Delta t_m + \sqrt{2\kappa^i(\overline{X}_{m-1})\Delta t_m}\Xi$
        $m = m + 1$
    compute $\overline{X}_{\overline{\tau}}$ and $t_{\overline{\tau}}$ according to (9)
    $F = F + f(\overline{X}_{\overline{\tau}})\Delta t_{\overline{\tau}}$
    $u^i = g(\overline{X}_{\overline{\tau}}) + F$
return $N^{-1}\sum_{i=1}^{N} u^i$ and $N^{-1}\sum_{i=1}^{N}\left(u^i - N^{-1}\sum_{i=1}^{N} u^i\right)^2$
```

Algorithm 1. Point estimate algorithm to compute the estimators $\mathrm{E}^{\mathrm{MS}}_{M,N}[u(x)]$ and $\mathrm{Var}^{\mathrm{MS}}_{M,N}[u(x)]$ using the simple one-point rectangle method for integration.

2.4. **Extension to the whole domain.** In the following, the pointwise approximation of the solution at some $x \in D$ obtained by some Monte Carlo estimator is extended to the whole domain D using interpolation techniques on a mesh. This allows applying finite element a posteriori error control with respect to a global (in D) approximation of the solution which we define as follows: Consider some given mesh $\mathcal{T}_h$ as a triangulation of the spatial domain D with vertices $\mathcal{N}_h = (\nu_h^i)_{i=1}^{|\mathcal{N}_h|}$ and edges $\mathcal{E}_h = \{E_i\}_{i=1}^{|\mathcal{E}_h|}$. Note that the coefficients of a Courant P_1 function correspond to the function values at the vertices (nodes) of the mesh and that the nodal interpolation operator denoted by $\mathcal{I}_h$ is defined by these values. Hence, let the discrete solution $\mathrm{E}^{\mathrm{MS}}_{\boldsymbol{M},\boldsymbol{N}}[u_h]$ with $\boldsymbol{M} = (M_i)_{i=1}^{|\mathcal{N}_h|}$ and $\boldsymbol{N} = (N_i)_{i=1}^{|\mathcal{N}_h|}$ be a Courant P_1 function on this mesh determined by setting the nodal values,

$$\mathrm{E}^{\mathrm{MS}}_{\boldsymbol{M},\boldsymbol{N}}[u_h]\,(\nu_h^i) \coloneqq \mathrm{E}^{\mathrm{MS}}_{M_i,N_i}\big[u(\nu_h^i)\big] \qquad \text{for } i = 1,\dots,|\mathcal{N}_h|\,.$$

This introduces three types of errors into the global approximation of u. The first is the *stochastic approximation error* originating from the Monte Carlo estimators. It can be controlled by means of the Central Limit Theorem (CLT) subject to the number of samples N_i for each nodal point. The second error arises from the *approximation of the diffusion process* and is controlled by the step widths Δt_m. The third error contribution results from the P_1 *interpolation* which is approximated based on the Monte Carlo estimators and determined by the interpolation mesh parameter h.

We shall consider two error representations. The first describes a *decomposition of the mean square error* into three parts resulting from the interpolation error, the discretization of the ordinary differential equation (8), and the sampling error in the Monte Carlo method. For the stochastic solution u_h^M of the discretized ordinary differential equation it holds $\mathrm{E}\big[u_h^M\big] = \mathrm{E}\big[\mathrm{E}^{\mathrm{MS}}_{\boldsymbol{M},\boldsymbol{N}}[u_h]\big]$, i.e., the estimator is unbiased. Then, for the pointwise mean square error in the approximation, we derive

$$\begin{aligned}
\mathrm{E}\Big[\big(\mathrm{E}^{\mathrm{MS}}_{\boldsymbol{M},\boldsymbol{N}}[u_h]-\mathrm{E}[u]\big)^2\Big] &= \mathrm{E}\Big[\mathrm{E}^{\mathrm{MS}}_{\boldsymbol{M},\boldsymbol{N}}[u_h]^2\Big] - 2\,\mathrm{E}\big[\big(\mathrm{E}^{\mathrm{MS}}_{\boldsymbol{M},\boldsymbol{N}}[u_h]\big)\big]\,\mathrm{E}[u] + \mathrm{E}[u]^2\\
&= \mathrm{E}\Big[\mathrm{E}^{\mathrm{MS}}_{\boldsymbol{M},\boldsymbol{N}}[u_h]^2\Big] - \mathrm{E}\big[\mathrm{E}^{\mathrm{MS}}_{\boldsymbol{M},\boldsymbol{N}}[u_h]\big]^2 + \big(\mathrm{E}\big[\mathrm{E}^{\mathrm{MS}}_{\boldsymbol{M},\boldsymbol{N}}[u_h]\big]-\mathrm{E}[u]\big)^2\\
&= \mathrm{Var}\big[\mathrm{E}^{\mathrm{MS}}_{\boldsymbol{M},\boldsymbol{N}}[u_h]\big] + \big(\mathrm{E}\big[u_h^M\big]-\mathrm{E}[u]\big)^2\\
&= \frac{1}{N}\,\mathrm{Var}\big[u_h^M\big] + \big(\mathrm{E}\big[u_h^M\big]-\mathrm{E}[u]\big)^2,
\end{aligned} \tag{10}$$

since $\mathrm{Var}\big[\mathrm{E}^{\mathrm{MS}}_{\boldsymbol{M},\boldsymbol{N}}[u_h]\big] = \mathrm{Var}\Big[\frac{1}{N}\sum_{i=1}^N u_h^M(\omega_i)\Big] = \frac{1}{N}\,\mathrm{Var}\big[u_h^M\big]$ for the uncorrelated samples of u_h^M.

The second decomposition seeks to represent the error *locally in the L^2 norm.* We assume some convex $D \subseteq \mathbb{R}^2$ for the sake of a simpler presentation and H^2 regularity of the solution u. Higher dimensions are possible with different inequalities for the norms. Consider the element $T \in \mathcal{T}_h$ and the pointwise P_1 interpolation operator $\mathcal{I}_h$ on the mesh $\mathcal{T}_h$. By a triangle inequality and interpolation error estimates, for the approximation error it holds that

$$\begin{aligned}
\big\|\mathrm{E}[u]-\mathrm{E}^{\mathrm{MS}}_{\boldsymbol{M},\boldsymbol{N}}[u_h]\big\|_{L^2(T)} &\le \|\mathrm{E}[u]-\mathcal{I}_h\,\mathrm{E}[u]\|_{L^2(T)} + \big\|\mathcal{I}_h\,\mathrm{E}[u]-\mathrm{E}^{\mathrm{MS}}_{\boldsymbol{M},\boldsymbol{N}}[u_h]\big\|_{L^2(T)}\\
&\le \|\mathrm{E}[u]-\mathrm{E}[u_h]\|_{L^2(T)} + \|\mathrm{E}[u_h]-\mathcal{I}_h\,\mathrm{E}[u]\|_{L^2(T)}\\
&\qquad + \big\|\mathcal{I}_h\,\mathrm{E}[u]-\mathrm{E}\big[u_h^M\big]\big\|_{L^2(T)} + \big\|\mathrm{E}\big[u_h^M\big]-\mathrm{E}^{\mathrm{MS}}_{\boldsymbol{M},\boldsymbol{N}}[u_h]\big\|_{L^2(T)}\\
&\lesssim h_T\eta_T + h_T^2\,\|\Delta u\|_{L^2(T)} + |T|\,\Delta t_0 + |T|\max_{K\in\mathcal{N}(T)}\Big\{\mathrm{Var}\big[u_h^M\big]^{1/2}\,N_K^{-1/2}\Big\}.
\end{aligned} \tag{11}$$

The element-wise error indicator η_T which controls the FE approximation error is described in Section 3. While the first two terms are both governed by properties of the mesh $\mathcal{T}_h$ and can be controlled through refinement as well as the adaptive algorithms described in Section 3, the third term represents the Monte Carlo estimation error. It solely depends on the number of samples used for each node in the element as the variance converges to the variance of the continuous solution $\mathrm{Var}[u]$ for $M \to \infty$ and $h \to 0$. The last term represents the error in the approximation of the diffusion process in the Euler scheme for the stochastic differential equation. Numerical experiments show that $\Delta t_0 \simeq h_{\min}$ is a reasonable choice in the two-dimensional case. The second term is of higher order with respect to h and is therefore omitted in the further calculations as it vanishes with a higher rate of convergence for $h \to 0$.

3. Adaptivity

To achieve convergence of the method presented in Section 2 by means of an adaptive algorithm, all error components in the decomposition (11) have to converge separately. For optimal convergence, for all contributing parts the same rate should be achieved with respect to the computational effort that has to be invested to gain the error reduction. The main idea of an adaptive algorithm is therefore to base the parameter choices in some way on the underlying mesh. Hence, in the following it is the goal to define some optimal sequence of meshes $\mathcal{T}_0 \subset \ldots \subset \mathcal{T}_L$ for the interpolation and then to choose appropriate values for the other discretization parameters.

We use the convenience notation $a \lesssim b$ (or $a \approx b$) to state that $a \leq Cb$ (or $C^{-1}b = a = Cb$) with a constant $C > 0$ independent of the discretization parameters.

Our starting point is an initial quasi-uniform triangulation $\mathcal{T}_0$ of the spatial domain D. The first step is to calculate a discrete solution $\mathrm{E}^{\mathrm{MS}}_{\boldsymbol{M_0},\boldsymbol{N_0}}[u_0]$. The parameters $\boldsymbol{M_0}$ and $\boldsymbol{N_0}$ have to be guessed as not enough information is available on the initial level[3]. The next step involves the calculation of some finite element error indicator η_0 which we briefly describe.

One of the most common and simple a posteriori FE error estimation techniques is given by the residual error estimator. We define the local error indicator $\eta_{h,T}$ on some element $T \in \mathcal{T}_h$ of size h_T subject to an approximation u_h of the true solution u by

$$\eta_{h,T}^2 = h_T^2 \left\| f \right\|_{L^2(T)}^2 + \sum_{E \in \mathcal{E}(T)} h_T \left\| \nabla u_h \cdot n_E \right\|_{L^2(E)}^2 . \tag{12}$$

Recall that the set of edges of $\mathcal{T}_h$ is denoted $\mathcal{E}_h$ and the edges of some element T are denoted $\mathcal{E}(T)$. More details and a proper derivation can e.g. be found in [24] and the references therein.

Once a local error indicator $\eta_{h,T}$ is readily available, a mesh refinement strategy can be chosen which selects a set of elements $\mathcal{M}_h \subset \mathcal{T}_h$ such that an error reduction is achieved for the approximate solution on the finer mesh. A common choice is the so-called bulk or Dörfler marking defined for some fraction parameter $0 < \theta \leq 1$ by

$$\sum_{T \in \mathcal{M}_h} \mathrm{E}[\eta_{h,T}]^2 \geq \theta \sum_{T \in \mathcal{T}_h} \mathrm{E}[\eta_{h,T}]^2 . \tag{13}$$

We also employ this marking and refer to [25] for further details, in particular with regard to error reduction properties.

[3] the subscript denotes the refinement level for which the parameters or functions are considered

As can be seen in (11), the estimator only covers the first term $\|\mathrm{E}[u]-\mathrm{E}[u_0]\|_{L^2(T)}$ and thus the parameters $\boldsymbol{M_0}$ and $\boldsymbol{N_0}$ have to be chosen such that

$$h_T\eta_{0,T} \gg \left\|\mathcal{I}_h\,\mathrm{E}[u]-\mathrm{E}\big[u_h^M\big]\right\|_{L^2(T)} + |T| \max_{K\in\mathcal{N}(T)} \left\{\mathrm{Var}\big[u_h^M\big]^{1/2} N_K^{-1/2}\right\}.$$

The reason for this is that the finite element solution is bounded by the interpolator but the stochastic properties of the discrete solutions introduce oscillations of length h and amplitudes as seen above. These can be unbounded in principle and artificially increase the finite element error estimator. As refinement would not reduce these oscillations, we have to limit them by means of the central limit theorem such that we can get reliable mesh refinements. In fact, the stochastic error term from (11) for each element has to be smaller than the smallest error estimator $\eta_{\ell,T}$ chosen for the refinement set $\mathcal{M}_\ell$ (on refinement levels $\ell=1,\dots,L$) given by

$$\min_{T\in\mathcal{M}_\ell} \eta_{0,T} > |T| \max_{K\in\mathcal{N}_\ell(T)} \left\{\mathrm{Var}\big[u_h^M\big]^{1/2} N_K^{-1/2}\right\} + \left\|\mathcal{I}_h\,\mathrm{E}[u]-\mathrm{E}\big[u_h^M\big]\right\|_{L^2(T)}. \tag{14}$$

This constraint also includes the error of the approximation in the Euler scheme but it is deterministic, smooth, and global with respect to the spatial domain D. As a result, it alters the error estimator only to some minor extent.

The derived refinement $\mathcal{T}_1$ of the initial mesh $\mathcal{T}_0$ is the basis for the next level and the process is repeated. Heuristics based on educated guesses of the error components allow to balance the parameters in actual computations. This however is only possible if there are at least three meshes, that is $L\geq 3$. Section 3.1 covers some approaches to this topic in more detail.

3.1. **A practical parameter selection strategy.** The error decomposition in (11) results in three components that need to be balanced for guaranteed and optimal convergence. We hence aim at finding good estimates for the convergence rates with respect to the relevant parameters and then extrapolate the error estimates to the next level. With these, we can approximate parameters that fulfill the balancing requirements. The first task is the approximation of the convergence rate of the interpolation error subject to the current mesh and controlled through the parameter α. Let h be the minimal inradius over all the triangles of the triangulation $\mathcal{T}_h$ and suppose we have some $\alpha>0$ such that

$$\|\mathrm{E}[u]-\mathrm{E}[\mathcal{I}_h u]\|_{L^2(D)} \lesssim h^\alpha.$$

Standard finite element theory gives

$$\|\mathrm{E}[u]-\mathrm{E}[\mathcal{I}_h u]\|_{L^2(D)} \sim \|\mathrm{E}[u]-\mathrm{E}[u_h]\|_{L^2(D)}.$$

Thus, with some efficient and reliable error estimate $h\eta_h \sim \|\mathrm{E}[u]-\mathrm{E}[u_h]\|_{L^2(D)}$, we get $h\eta_h \sim h^\alpha$. We exploit this property to gauge the parameter α as error estimators have smoothing properties and thus in practice exhibit a behavior closer to a monotonic convergence. For the estimation α_L of α, we carry out a linear regression over data points $(\log(h_\ell),\log(h\eta_\ell))_{\ell=1,\dots,L}$.

Next, we estimate the expected spatial error on the next level $L+1$. For this, the triangle inequality for $\ell=1,\dots,L$ yields

$$\|\mathrm{E}[u]-\mathrm{E}[u_\ell]\|_{L^2(D)} \leq \|\mathrm{E}[u]-\mathrm{E}[u_L]\|_{L^2(D)} + \|\mathrm{E}[u_L]-\mathrm{E}[u_\ell]\|_{L^2(D)}.$$

The last term on the right-hand side is computable and it remains to estimate the error on level L. We assume $\|\mathrm{E}[u]-\mathrm{E}[u_L]\|_{L^2(D)} \ll \|\mathrm{E}[u_L]-\mathrm{E}[u_\ell]\|_{L^2(D)}$ for the coarser levels $\ell<L$ since, for a sufficiently high convergence rate α and a fine mesh $\mathcal{T}_L$, it holds $h_\ell^\alpha/h_L^\alpha \gg 1$ and we can conclude that

$$\|\mathrm{E}[u]-\mathrm{E}[u_\ell]\|_{L^2(D)} \approx \|\mathrm{E}[u_L]-\mathrm{E}[u_\ell]\|_{L^2(D)}$$

for each level $\ell=1,\dots,L-1$.

As we already have a good approximation of the expected convergence rate α_L, for brevity of notation denoted hereafter by α, we can now find approximations of the error $\|\mathrm{E}[u]-\mathrm{E}[u_L]\|_{L^2(D)}$ with the help of the errors on the coarser levels by

$$\|\mathrm{E}[u]-\mathrm{E}[u_\ell]\|_{L^2(D)} \sim h_\ell^\alpha \qquad \text{and} \qquad \|\mathrm{E}[u]-\mathrm{E}[u_L]\|_{L^2(D)} \sim h_L^\alpha.$$

To improve the readability we write α instead of α_L for the numerical approximations hereafter. For each $\ell = 1,\dots,L-1$, this asymptotically yields the identity

$$\frac{\|\mathrm{E}[u]-\mathrm{E}[u_L]\|_{L^2(D)}}{\|\mathrm{E}[u]-\mathrm{E}[u_\ell]\|_{L^2(D)}} = \frac{h_L^\alpha}{h_\ell^\alpha}.$$

Hence, we can construct the approximation

$$\|\mathrm{E}[u]-\mathrm{E}[u_L]\|_{L^2(D)} \approx \frac{h_L^\alpha}{h_\ell^\alpha}\|\mathrm{E}[u_L]-\mathrm{E}[u_\ell]\|_{L^2(D)}.$$

We subsequently define the estimate $\tilde{e}_L$ for the error $\|\mathrm{E}[u]-\mathrm{E}[u_L]\|_{L^2(D)}$ on level L as the arithmetic mean of the different extrapolations from the coarser levels $\ell = 1,\dots,L-1$, by

$$\tilde{e}_L \coloneqq \frac{1}{L-1}\sum_{\ell=1}^{L-1}\frac{h_L^\alpha}{h_\ell^\alpha}\|\mathrm{E}[u_L]-\mathrm{E}[u_\ell]\|_{L^2(D)}. \tag{15}$$

The same technique is used to gauge the expected error on level $L+1$. The finer mesh $\mathcal{T}_{L+1}$ on level $\ell = L+1$ is obtained from a local refinement (13). Thus, the parameter h_{L+1} can be used to generate the extrapolation $\hat{e}_{L+1}$ with the same levels $L = 1,\dots,L-1$,

$$\hat{e}_{L+1} \coloneqq \frac{1}{L-1}\sum_{\ell=1}^{L-1}\frac{h_{L+1}^\alpha}{h_\ell^\alpha}\|\mathrm{E}[u_L]-\mathrm{E}[u_\ell]\|_{L^2(D)}.$$

Remark 3.1. The derived estimate $\tilde{e}_L$ for the error in the $L^2(D)$ norm is also suitable as a stopping criterion e_{TOL} to determine the convergence of the adaptive method. This estimate is with respect to the last level L for which a numerical solution has been computed. If it is smaller than the criterion e_{TOL}, the computation is stopped.

The next step is to balance the expected Monte Carlo error with the extrapolated spatial error. For the refinement scheme (13), it is crucial to keep the Monte Carlo error well below the spatial error as otherwise this error gets picked up by the estimator which results in wrong refinement and thus unstable behavior and suboptimal convergence or even in no convergence at all. Hence, we introduce the balancing factor δ which describes the desired relation between the two errors in (10) by

$$\delta^2 = \frac{\left(\mathrm{E}[u]-\mathrm{E}\big[u_h^M\big]\right)^2}{N^{-1}\operatorname{Var}\big[u_h^M\big]}. \tag{16}$$

A choice of $\delta = 1$ would lead to equality. For some constant c_N^ℓ for each level $\ell = 1,\dots,L$, we set the numbers of samples in the vertices $\mathcal{N}_\ell = (\nu_\ell^i)_{i=1}^{|\mathcal{N}_\ell|}$ as

$$N_\ell^i \coloneqq c_N^\ell \operatorname{Var}\big[u_\ell^M(\nu_\ell^i)\big] \qquad \text{for } i = 1,\dots,|\mathcal{N}_\ell|. \tag{17}$$

The next task is to choose c_N^{L+1} appropriately so that (16) will be fulfilled for $h = h_{L+1}$. In fact, applying (17) to (16) results in

$$\mathrm{E}[u]-\mathrm{E}\big[u_L^M\big] = \delta\left(c_N^L \operatorname{Var}\big[u_L^M\big]\right)^{-1/2}\operatorname{Var}\big[u_L^M\big]^{1/2} = \delta(c_N^L)^{-1/2}.$$

We assume, with $u^M = \lim_{h\to 0} u_h^M$, that $\operatorname{Var}\bigl[u_\ell^M\bigr] \approx \operatorname{Var}\bigl[u^M\bigr]$ for $\ell = 1,\dots,L$ is a sufficient approximation since only a rough estimate of the variance is needed. Taking the L^2 norm of the last equation and applying the variance approximation yields

$$c_N^L = \delta^2 \left|D\right|^2 \left\|\mathrm{E}\bigl[u_L^M\bigr] - \mathrm{E}[u]\right\|_{L^2(D)}^{-2}.$$

With the extrapolated estimate $\hat{e}_{L+1} \approx \left\|\mathrm{E}[u] - \mathrm{E}\bigl[u_{L+1}^M\bigr]\right\|$, we get an estimate for the constant c_N^{L+1} as $\hat{c}_N^{L+1} = \frac{|D|^2}{\delta^2}\hat{e}_{L+1}^{-2}$. The numbers of samples for level $L+1$ are now set according to (17) by

$$N_{L+1}^i \coloneqq \hat{c}_N^{L+1} \operatorname{Var}\bigl[u_L^M(\nu_{L+1}^i)\bigr] \qquad \text{for } i = 1,\dots,\left|\mathcal{N}_{L+1}\right|. \tag{18}$$

Remark 3.2. It is imperative to ensure a minimum number of samples for each N_L^i in (18) since on each level a sufficient approximation of the variance $\operatorname{Var}\bigl[u_\ell^M\bigr]$ has to be available. This is important since otherwise the algorithm might become unstable due to severe undersampling in single points. This would result in a bad spatial error estimation in (14) and thus suboptimal mesh refinement with reduced convergence rate or even lack of convergence.

To alleviate this issue, a simple solution is to choose some $N_{\min}$ independent of all parameters and especially independent of the level ℓ. The practical application requires some rough estimate of the variance which can be computed alongside the expected value and thus set the numbers of samples on level $L+1$ as

$$\hat{N}_{L+1}^i \coloneqq \max\left\{\hat{c}_N^{L+1} \operatorname{Var}_{\boldsymbol{M},\boldsymbol{N}}^{\mathrm{MC}}\bigl[u_L^M(\nu_{L+1}^i)\bigr], N_{\min}\right\} \quad \text{for } i = 1,\dots,\left|\mathcal{N}_{L+1}\right|. \tag{19}$$

Finally, it remains to choose the parameter Δt on each level. The influence of this factor on the error is given in the last term of (11). Since we assume linear pointwise convergence with respect to Δt, we choose the relation $h \sim \Delta t$. In Algorithm 2, the overall adaptive algorithm is sketched. The computation of the employed pointwise solution estimator $\mathrm{E}_{\boldsymbol{M},\boldsymbol{N}}^{\mathrm{MS}}[u_\ell]$ is depicted in Algorithm 1.

Note that instead of the variance based adaptive local number of samples for each vertex, one could also choose a common number of samples based on the variance $\operatorname{Var}\bigl[u_L^M\bigr]$.

Remark 3.3. In the numerical calculations one has to impose (14) well enough, such that the algorithm becomes stable. This can be achieved by choosing $\delta < 1$. In the experiments in Section 4, we choose $\delta = 0.2$. This allows for a stable algorithm and only imposes a slight impact on performance.

3.2. **Adaptation to a goal functional.** In addition to the computation of the global solution u, we also consider a derived *quantity of interest* Q depending on the solution of the form

$$Q(u) = \mathrm{E}\left[\int_D qu \,\mathrm{d}x\right], \tag{20}$$

with a non-negative weight function $q \in L^2(D)$, i.e. $q \geq 0$ on D. For such functionals, an adaptive procedure can result in different refinements than what one gets when solving for the global solution. This is particularly true if q has a smaller support than D. In the case of goal-oriented adaptive FE algorithms [26], an adaptive error indicator usually is based on the solution of some dual problem in order to identify the area of influence of the goal. However, the localized nature of the presented approach allows to quantify the error contribution directly, i.e. without any global dual solutions. Therefore, we quantify the error in the goal evaluation with the help of the weighted norm

$$\|v\|_q := \left(\int_D (vq)^2 \, \mathrm{d}x \right)^{1/2} .$$

In this norm, we can perform the same residual error estimation approach as in the global case described in the preceding section. With the local norms $\|v\|_{q,T}^2 := \int_T (vq)^2 \, \mathrm{d}x$ on $T \in \mathcal{T}$ and $\|v\|_{q,E}^2 := \int_E (vq)^2 \, \mathrm{d}s$ on $E \in \mathcal{E}$, we define the local error contributions estimator by

$$\eta_{h,t,g}^2 := h_T^2 \|f\|_{q,T}^2 + \sum_{E \in \mathcal{E}(T)} h_T \|\nabla u_h \cdot n_E\|_{q,E}^2$$

for the local mesh refinement indicator. A similar approach is used for the local number of samples such that instead of $\hat{N}_{L+1}^i$ in (19) we apply

$$\hat{N}_{L+1}^i := \max \left\{ q(x_i) \hat{q}^{-1} \hat{c}_N^{L+1} \operatorname{Var}_{\boldsymbol{M},\boldsymbol{N}}^{\mathrm{MC}} \left[u_L^M (\nu_{L+1}^i) \right], N_{\min} \right\} \quad \text{for } i = 1, \dots, |\mathcal{N}_{L+1}| ,$$

where $\hat{q}$ is the maximum of q in the domain D and x_i is the location of vertex i.

Remark 3.4. The proposed refinement with respect to Q is merely a heuristical approach without a proper analytical derivation. A common case for a goal functional is the point evaluation of the solution u. In the finite element approach, this requires an approximation by some mollifier with a small support since evaluations on a null set are not well-defined for H^1 functions, see [26]. With our approach, this can be computed directly by the solution of the SDE at the single point instead of the global evaluation in (20).

```
In  : $\mathcal{T}_0$, $\boldsymbol{N}_{init}$, $\Delta t_0$, $e_{\mathrm{TOL}}$
Out: solution $\mathrm{E}_{\boldsymbol{M},\boldsymbol{N}}^{\mathrm{MS}}[u_L]$

    for $\ell = 1, 2, \dots$ do
        if $\ell \geq 2$ then
            compute $\boldsymbol{N}_\ell$ according to (19)
        else
            set $\boldsymbol{N}_\ell$ to $\boldsymbol{N}_0$
        set $\Delta t$ according to $h$
        compute $\mathrm{E}_{\boldsymbol{M},\boldsymbol{N_\ell}}^{\mathrm{MS}}[u_\ell]$ and $\operatorname{Var}_{\boldsymbol{M},\boldsymbol{N_\ell}}^{\mathrm{MS}}[u_\ell]$ at $x \in \mathcal{N}_\ell$ with Algorithm 1
        if $\tilde{e}_L \leq e_{TOL}$ then
            break
        compute $\eta_\ell$
        refine $\mathcal{T}_\ell$ with $\eta_\ell$ to get $\mathcal{T}_{\ell+1}$
    return $\mathrm{E}_{\boldsymbol{M},\boldsymbol{N}}^{\mathrm{MS}}[u_L]$
```

Algorithm 2. Adaptive algorithm for the stochastic representation with stopping criterion based on $\tilde{e}_L$ from (15).

4. Numerical Experiments

This section is concerned with the illustration of the presented adaptive sampling method based on some numerical benchmark problems. The visualizations show properties of the unique feature of this approach, namely its proper separation of the spatial and stochastic domain. This allows to choose the number of samples locally based on the variance in any part of the spatial domain as well as to exploit sparsity

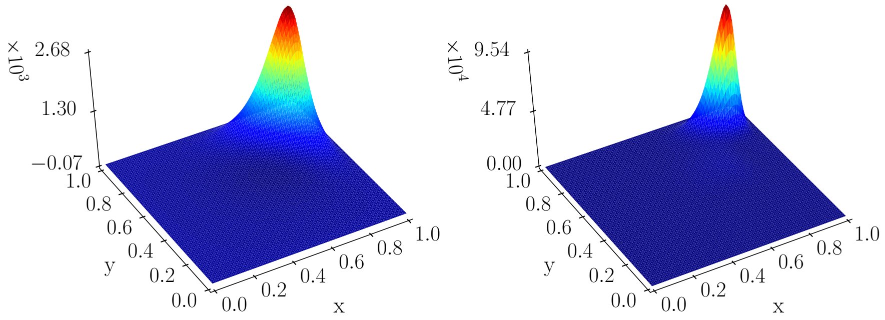


Figure 2. The mean (left) and the variance (right) of the solution for the numerical example.

with respect to the sampling locations. For comparison purposes we compute the solution for the experiments with a fixed number of samples for all the nodal points in the domain using uniform and adaptive meshes and we use different numbers of samples for the same meshes.

Consider the input data for the problem in (1) with $f = \Delta u^*$ and

$$u^* = 5x_1^2(1-x_1)^2(e^{10x_1^2}-1)x_2^2(1-x_2)^2(e^{10x_2^2}-1),$$

together with homogeneous Dirichlet boundary data $u \equiv 0$ on ∂D.

A common representation of (Gaussian) random fields is the Karhunen-Loeève expansion, see e.g. [14, 3]. In general, given the covariance function of a random field, it is an L^2 converging expansion in the eigenfunctions of the covariance integral operator weighted by the square roots of the eigenvalues of this operator. For the experiments, we assume an expansion of this type[4], which then reads as

$$\kappa(x,\varphi) := \frac{c_a}{\alpha_{\min}} \left(\sum_{m=1}^{t_a} a_m(x)\varphi_m + \alpha_{\min} \right) + \varepsilon_a.$$

Here, the φ_m are standard uniformly distributed uncorrelated random variables in [-1,1]. The parameters $c_a, \varepsilon_a > 0$ and the truncation length $t_a \in \mathbb{N}$ control the properties of the random field. The coefficients a_m are defined for $m = 1, 2, ...$ for the parameters $\sigma_\alpha > 1$ and $0 < A_\alpha < 1/\zeta(\sigma_\alpha)$ with the Riemann zeta function ζ by

$$\begin{aligned} a_m(x) &= \alpha_m \cos(2\pi\beta_1(m)x_1)\cos(2\pi\beta_2(m)x_2), \quad \alpha_m = A_\alpha m^{-\sigma_\alpha}, \\ \beta_1(m) &= m - k(m)(k(m)+1)/2, \quad \beta_2(m) = k(m) - \beta_1(m), \\ k(m) &= \lfloor -1/2 + \sqrt{1/4 + 2m} \rfloor. \end{aligned}$$

In our experiment we choose the setting $\sigma_\alpha = 2, A_\alpha = 0.6, c_\kappa = 1, \varepsilon_\alpha = 0.5\cdot 10^{-3}$ and $t_\alpha = 5$, which ensures uniform ellipticity of the PDE. Together with the prescribed right-hand side f this results in strong oscillations in the corner $(1,1)$ of the unit square together with a high variance of the solution in the same region. The resulting mean $\mathrm{E}[u]$ and variance $\mathrm{Var}[u]$ are depicted in Figure 2 as computed by a standard Monte Carlo finite element method.

The algorithm automatically chooses the number of samples locally and applies mesh adaptivity at the same time. Plotting the numbers of samples for each vertex of the mesh as a P_1 projection thus gives a false impression of the distribution of the computational effort in the spatial domain. Hence, we define the additional

[4] Note that any other approach to generate random field realizations could be employed similarly.

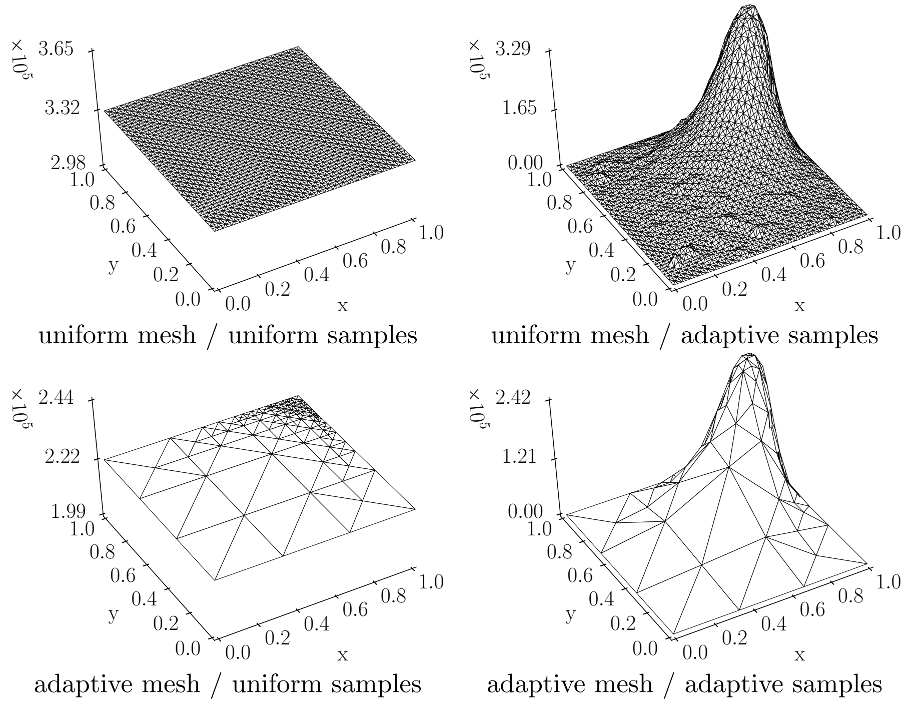
quantity $\mathcal{C}_T$ piecewise on each element $T \in \mathcal{T}$ as the mean over the elements three nodes as

$$\mathcal{C}_T = \frac{1}{3\,|T|} \sum_{i=1}^{3} N_i \tag{21}$$

which gives the average number of samples used for each element weighted by the size of that element as a measure for the computational effort per area.

In the experiments we set the minimum number of samples $N_{\min}$ for each vertex as 100. We continue the algorithm until our error heuristic drops below the prescribed value $\left\|\mathsf{e}^{\mathrm{MS}}\right\|_{H^1(D)} = \left\|\mathrm{E}[u] - \mathrm{E}^{\mathrm{MS}}_{\boldsymbol{M},\boldsymbol{N}}[u_h]\right\|_{H^1(D)} \leq 2 \cdot 10^3$. We consider four experiments. The first uses a fixed number of samples for each level and uniform meshes, the second one applies adaptive meshes, the third experiment combines uniform meshes with locally chosen numbers of samples, and the fourths experiment uses both adaptive meshes and local sample adaptivity.

All computations have been carried out in parallel on fourteen core Intel Xeon E5-2697 v3 shared memory systems with 2.6 GHz clock frequencies running OpenSUSE using a combination of C++ programs for the solution of the SDE and Python with the FEniCS toolbox for the finite element related computations and the overall control.



uniform mesh / uniform samples

uniform mesh / adaptive samples

adaptive mesh / uniform samples

adaptive mesh / adaptive samples

Figure 3. The number of samples for each vertex for different methods.

The resulting numbers of samples are plotted in Figure 3. The differences in the number of samples between two uniform samples approaches is due to the variance in the approximation of the error contributions in the heuristics. The weighting parameter δ in (16) has to be chosen such that this estimate is sufficiently stable. Note that already a computational advantage is visible for the adaptive sampling variants whereas both uniform sampling experiments falsely appear to be similar in effort. This changes if one considers the plots of $\mathcal{C}_T$ shown in Figure 4. The adaptive mesh reduces the amount of vertices in the smooth regions of the solution whereas the local sample number reduces the amount of samples for vertices with a small variance. In the optimal case of the combined method, the adaptivity eliminates a lot of expensive vertices with a high number of samples and the local sampling reduces the amount of samples needed in areas with a highly oscillating mean.

The maximum computational effort is in the same order of magnitude for all four experiments but the computational effort $\mathcal{C}_T$ drops dramatically away from the $(1, 1)$-corner in all but the first experiment. This effect is more pronounced for the two experiments with mesh adaptivity. The last experiment improves this even further as fewer samples are used for the small triangles which are closer to $(0.5, 1)$ and $(1, 0.5)$.

The above observations correspond with the convergence in the H^1 norm with respect to the measured processor time as presented in Figure 5. The slowest method uses uniform meshes and applies a constant number of samples for all vertices in the domain. Local numbers of samples reduce the time almost nine-fold whereas the mesh adaptivity gives an improvement factor of 80. The combination of the two methods gives the fastest algorithm with an increase in speed of almost two orders of magnitude.

In Figure 6 the convergence of the error in the $L^2(D)$ norm is plotted with respect to the same computational effort. Here again, a uniform mesh with constant sample numbers results in the slowest method. Contrary to the last graph, all other methods perform very similarly to each other with an approximate improvement of 8-10 compared to the slowest approach.

In the final example we consider the weight for the goal functional in (20)

$$q = \begin{cases} C_g \varepsilon_g^{-2} \exp\left(\frac{-1}{1-\|x_0-x\|^2 \varepsilon_g^{-2}}\right) & \text{if } \|x_0 - x\|_{L^2(D)} \leq \varepsilon, \\ 0 & \text{else.} \end{cases}$$

We choose the location $x_0 = (0.6, 0.8)$, radius $\varepsilon_g = 0.1$, and weight $C_g = 2.14$. In Figure 7 the effect of the different adaptive methods is shown. The greatest benefit stems from the goal-weighted adaptive finite element method by comparing uniform methods with global adaptivity as in the previous example and the goal-weighted adaptive methods. Compared to uniform refinements it is up to two orders of magnitude faster in achieving the same error level. Local adaptive variance and its goal-weighted variant add upon this improvement. With global finite element adaptivity no convergence with respect to the goal is observed.

References

[1] Felix Anker, Christian Bayer, Martin Eigel, Marcel Ladkau, Johannes Neumann, and John Schoenmakers. SDE based regression for random PDEs. *WIAS preprint*, 2015.

[2] 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(1):322–352, 2013.

[3] Gabriel J Lord, Catherine E Powell, and Tony Shardlow. *An Introduction to Computational Stochastic PDEs.* Number 50. Cambridge University Press, 2014.

[4] Ralph C Smith. *Uncertainty Quantification: Theory, Implementation, and Applications*, volume 12. SIAM, 2013.

[5] Olivier P Le Maître and Omar M Knio. *Introduction: Uncertainty Quantification and Propagation.* Springer, 2010.

[6] Christoph Schwab and Claude Jeffrey Gittelson. Sparse tensor discretizations of high-dimensional parametric and stochastic PDEs. *Acta Numer.*, 20:291–467, 2011.

[7] 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(1):3–15, 2011.

[8] I. G. Graham, F. Y. Kuo, D. Nuyens, R. Scheichl, and I. H. Sloan. Quasi-Monte Carlo methods for elliptic PDEs with random coefficients and applications. *J. Comput. Phys.*, 230(10):3668–3694, 2011.

[9] Frances Y. Kuo, Christoph Schwab, and Ian H. Sloan. Quasi-Monte Carlo finite element methods for a class of elliptic partial differential equations with random coefficients. *SIAM J. Numer. Anal.*, 50(6):3351–3374, 2012.

[10] Andrea Barth, Annika Lang, and Christoph Schwab. Multilevel Monte Carlo method for parabolic stochastic partial differential equations. *BIT*, 53(1):3–27, 2013.

[11] Ivo Babuška, Fabio Nobile, and Raúl Tempone. A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM J. Numer. Anal.*, 45(3):1005–1034, 2007.

[12] Fabio Nobile, Raul Tempone, and Clayton G. Webster. An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM J. Numer. Anal.*, 46(5):2411–2442, 2008.

[13] Fabio Nobile, Raul Tempone, and Clayton G. Webster. A sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM J. Numer. Anal.*, 46(5):2309–2345, 2008.

[14] Roger G Ghanem and Pol D Spanos. *Stochastic finite elements: A spectral approach.* Courier Corporation, 2003.

[15] Hermann G. Matthies and Andreas Keese. Galerkin methods for linear and nonlinear elliptic stochastic partial differential equations. *Comput. Methods Appl. Mech. Engrg.*, 194(12-16):1295–1331, 2005.

[16] Ivo Babuška, Raúl Tempone, and Georgios E. Zouraris. Solving elliptic boundary value problems with uncertain coefficients by the finite element method: the stochastic formulation. *Comput. Methods Appl. Mech. Engrg.*, 194(12-16):1251–1294, 2005.

[17] Ivo Babuška, Raúl Tempone, and Georgios E. Zouraris. Galerkin finite element approximations of stochastic elliptic partial differential equations. *SIAM J. Numer. Anal.*, 42(2):800–825, 2004.

[18] Philipp Frauenfelder, Christoph Schwab, and Radu Alexandru Todor. Finite elements for elliptic problems with stochastic coefficients. *Comput. Methods Appl. Mech. Engrg.*, 194(2-5):205–228, 2005.

[19] Martin Eigel, Claude Jeffrey Gittelson, Christoph Schwab, and Elmar Zander. Adaptive stochastic Galerkin FEM. *Comput. Methods Appl. Mech. Engrg.*, 270:247–269, 2014.

[20] Martin Eigel, Christian Merdon, and Johannes Neumann. An adaptive multi level Monte–Carlo method with stochastic bounds for quantities of interest in groundwater flow with uncertain data. *WIAS Preprint*, 2060, 2014.

[21] G. N. Milstein and M. V. Tretyakov. *Stochastic numerics for mathematical physics.* Scientific Computation. Springer-Verlag, Berlin, 2004.

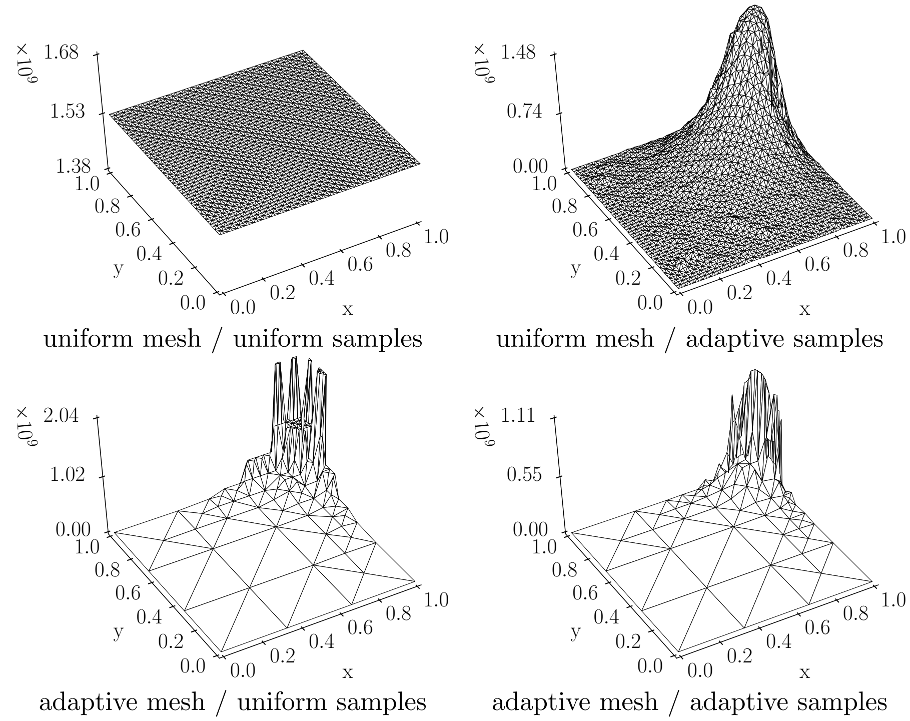
[22] G.N. Milstein and M.V. Tretyakov. Layer methods for stochastic Navier-Stokes equations using simplest characteristics. *Journal of Computational and Applied Mathematics*, 302:1 – 23, 2016.

[23] Michael B. Giles. Multilevel Monte Carlo path simulation. *Operations Research*, 56(3):607–617, 2008.

[24] Carsten Carstensen, Martin Eigel, Ronald H.W. Hoppe, and Caroline Löbhard. A review of unified a posteriori finite element error control. *Numer. Math. Theor. Meth. Appl.*, 5(4):509–558, 2012.

[25] Willy Dörfler. A convergent adaptive algorithm for Poisson's equation. *SIAM Journal on Numerical Analysis*, 33(3):1106–1124, 1996.

[26] Mark Ainsworth and J. Tinsley Oden. *A posteriori error estimation in finite element analysis.* Pure and Applied Mathematics (New York). Wiley-Interscience [John Wiley & Sons], New York, 2000.



uniform mesh / uniform samples

uniform mesh / adaptive samples

adaptive mesh / uniform samples

adaptive mesh / adaptive samples

Figure 4. The weighted computational effort $\mathcal{C}_T$ from (21) for each element $T \in \mathcal{T}$ for different methods.

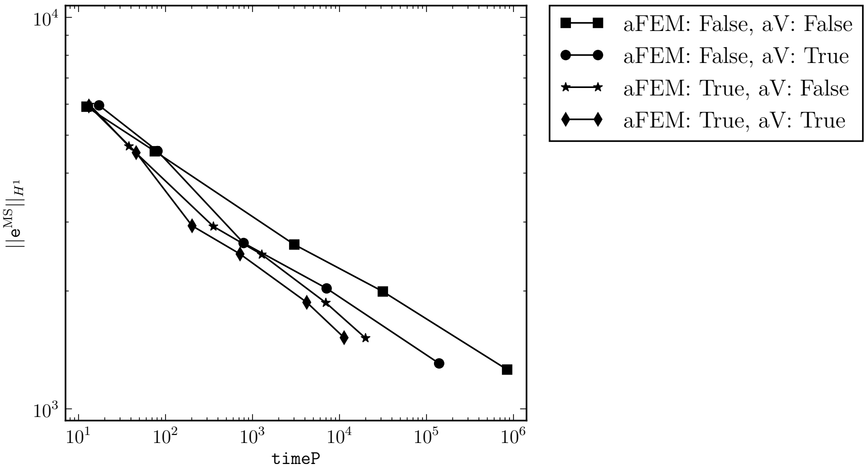


Figure 5. Convergence in the H^1 norm for the different methods. The abbreviations are `aFEM` for finite element adaptivity and `aV` for the localized adaptive number of samples.

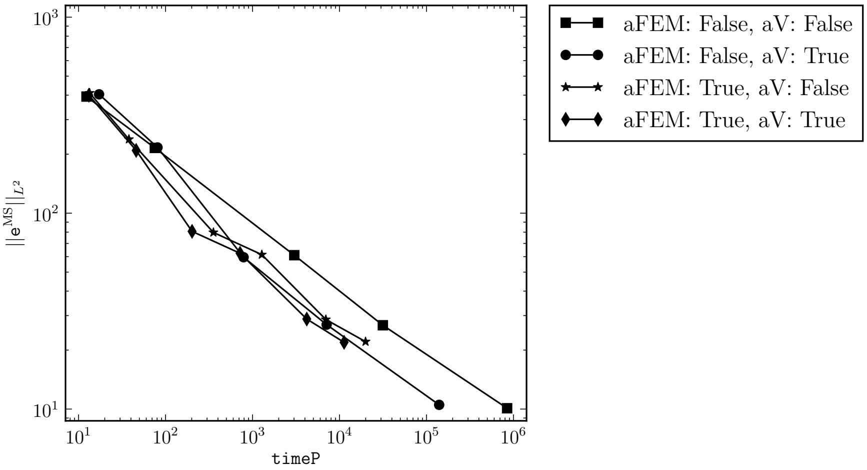


Figure 6. Convergence in the L^2 norm for the different methods. The abbreviations are `aFEM` for finite element adaptivity and `aV` for the localized adaptive number of samples.

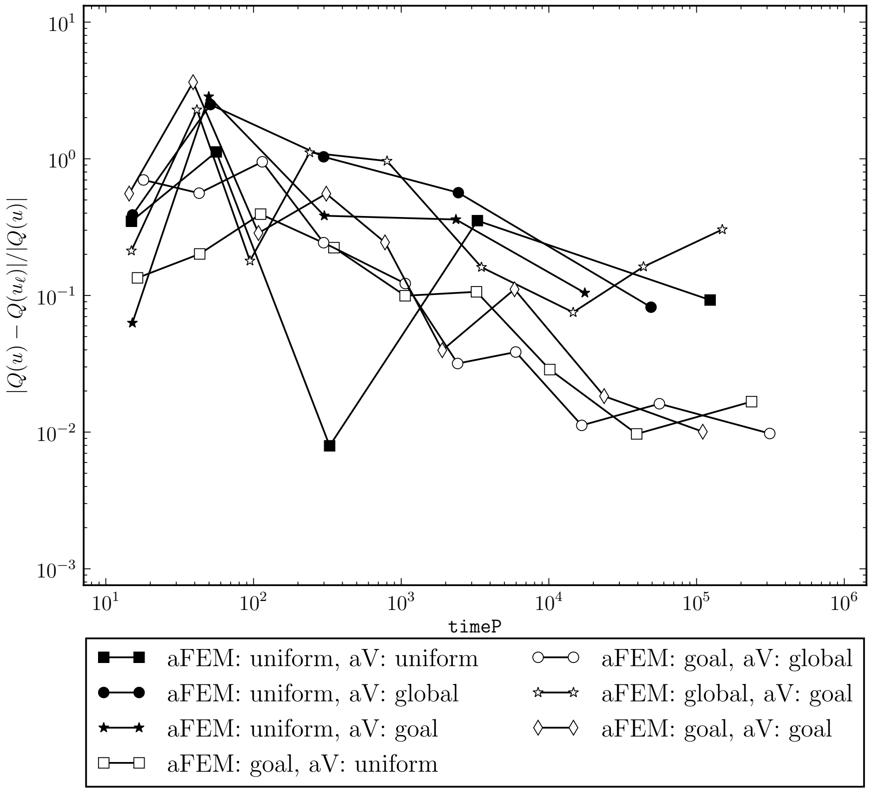


Figure 7. Convergence for the goal error for the different methods. The abbreviations are `aFEM` for finite element adaptivity, `aFEMg` for goal-weighted finite element adaptivity, `aV` for the localized adaptive number of samples and `aVg` for the goal-weighted localized adaptive number of samples.