ITERATIVE SOLVERS FOR THE STOCHASTIC
FINITE ELEMENT METHOD∗
EVELINE ROSSEEL† AND STEFAN VANDEWALLE†

Abstract. This paper presents an overview and comparison of iterative solvers for linear stochastic
partial differential equations (PDEs). A stochastic Galerkin finite element discretization is applied
to transform the PDE into a coupled set of deterministic PDEs. Specialized solvers are required to
solve the very high-dimensional systems that result after a finite element discretization of the resulting
set. This paper discusses one-level iterative methods, based on matrix splitting techniques; multigrid
methods, which apply a coarsening in the spatial dimension; and multilevel methods, which make
use of the hierarchical structure of the stochastic discretization. Also Krylov solvers with suitable
preconditioning are addressed. A local Fourier analysis provides quantitative convergence properties.
The efficiency and robustness of the methods are illustrated on two nontrivial numerical problems.
The multigrid solver with block smoother yields the most robust convergence properties, though a
cheaper point smoother performs as well in most cases. Multilevel methods based on coarsening
the stochastic dimension perform in general poorly due to a large computational cost per iteration.
Moderate size problems can be solved very quickly by a Krylov method with a mean-based precon-
ditioner. For larger spatial and stochastic discretizations, however, this approach suffers from its
nonoptimal convergence properties.

Key words. stochastic finite element method, polynomial chaos, multigrid, preconditioning

AMS subject classifications. 65N22, 35R60, 65F10

DOI. 10.1137/080727026

1. Introduction. Mathematical models of real life engineering and scientific
processes typically depend on parameters which are known only approximately or
which are inherently variable and stochastic. These models often take the form of
a stochastic partial differential equation (PDE) with coefficients that are random
variables, fields, or processes. Monte Carlo techniques [24] are a standard approach
to solving stochastic PDEs. During the last decades, however, various alternatives
have been proposed, for example, perturbation methods [1] and Neumann expansion
methods [26]. Among these, the stochastic finite element method (SFEM) [14] has
received particular attention. This method retains the advantages of Monte Carlo
simulations; i.e., it enables one to compute the full statistical characteristics of the
solution, while reducing the simulation time.

Two prominent variants of the SFEM are the stochastic collocation technique [2,
16] and the stochastic Galerkin method [14, 3]. Both approaches transform the
stochastic PDE into a set of deterministic PDEs. The former samples the stochastic
PDE in a set of collocation points and yields a separate deterministic PDE for each
collocation point. One of its main advantages is that existing software for determin-
istic PDEs can readily be reused. The number of collocation points may, however,
grow prohibitively quickly when a high accuracy is required or when the problem is
described by a large number of random variables. The stochastic Galerkin method
applies spectral finite element theory to transform a stochastic PDE into a coupled set

∗Received by the editors June 12, 2008; accepted for publication (in revised form) June 4, 2009;
†Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, B-
3001 Leuven, Belgium (eveline.rosseel@cs.kuleuven.be, stefan.vandewalle@cs.kuleuven.be). The first
author is a research assistant of the Fund for Scientific Research - Flanders (Belgium).
of deterministic PDEs. This set is in general smaller than for the stochastic collocation method, but is more complicated to solve. As a consequence, specialized solution methods are required, and solvers for the original deterministic problem cannot be straightforwardly reused.

Various iterative solvers have been proposed for stochastic Galerkin finite element discretizations, e.g., [13, 18, 25, 23, 7, 8, 22]. However, no general study of iterative solvers has been performed. In this work, multigrid methods and Krylov preconditioners previously discussed in the literature are generalized, and new iterative methods that exploit the special Kronecker product structure of the discretized Galerkin systems are explored. We categorize the methods into one-level methods, based on matrix splitting approaches; multigrid methods, which perform a coarsening in the spatial dimension; and multilevel methods, which make use of the hierarchical structure of the stochastic discretization. The use of Krylov methods is also addressed. The different methods are compared according to their computational cost, difficulty of implementation, and convergence properties. Convergence is studied by local Fourier analysis. Numerical experiments based on two nontrivial problems confirm the analysis.

We will show that for moderate spatial domains and variances, one-level block splitting approaches and mean-based preconditioning methods are most appropriate. They require only a limited implementation effort and result in low computing times. In case of larger problems, however, these methods suffer from their nonoptimal convergence properties. Optimal methods, based on multigrid techniques, are then required.

The paper is organized as follows. Section 2 gives an overview of the stochastic Galerkin finite element discretization and derives some properties of the stochastic discretization matrices. Section 3 discusses iterative solvers. First, one-level methods based on matrix splitting approaches are considered. Next, a multigrid method is presented, with spatial coarsening and the previously discussed one-level methods as smoother. Next, a multilevel method based on a coarsening in the stochastic dimension is proposed. The section ends with a discussion on preconditioned Krylov subspace methods. In section 4, the convergence properties of the methods presented in section 3 are analyzed. A local Fourier analysis is applied to obtain quantitative results. Based on the results of this section, a number of iterative methods are selected, which we test on two nontrivial numerical problems in section 5. In section 6, the conclusions of the paper are formulated.

2. The stochastic Galerkin finite element method.

2.1. Stochastic finite element discretization. Consider a stochastic PDE, defined by a linear stochastic differential operator $\mathcal{L}$ on a $d$-dimensional spatial domain $D \subseteq \mathbb{R}^d$,

$$\mathcal{L}(x, \omega)u(x, \omega) = f(x), \quad x \in D, \, \omega \in \Omega.$$  

(2.1)

The PDE (2.1) is completed with suitable boundary conditions. In order to simplify notation, we shall assume that the boundary conditions and the forcing term, $f(x)$, are deterministic. The operator $\mathcal{L}$ may, for example, correspond to a diffusion operator with a random diffusion coefficient. The stochastic nature of the problem is expressed by the variable $\omega$. This variable represents an element of a sample space $\Omega$ which defines, together with a $\sigma$-algebra $\mathcal{F}$ and a probability measure $\mathcal{P}$, a complete probability space $(\Omega, \mathcal{F}, \mathcal{P})$. A random variable $\xi_i(\omega)$ is a function which maps elements from a sample space to real values, $\xi_i : \Omega \to \mathbb{R}$. Further on, we shall write $\xi_i$ instead of $\xi_i(\omega)$. The integral of a random variable with respect to

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.
are optimal in the case of Gaussian random variables. However, appropriate choice for uniformly distributed random variables. Hermite polynomials converge of expansion (2.4) [37, 27]. For example, Legendre polynomials are the random variables, different sets of orthogonal polynomials can be selected for optimal

\[ \alpha \text{ appearing in } \langle \xi \rangle \text{ follows.} \]

with Ψ = [Ψ₁(ξ), ..., Ψₚ(ξ)]ᵀ a set of orthogonal polynomials defined in the L random variables appearing in (2.3). Note that we shall assume that these random variables are independent. The number of polynomials Q in the expansion depends on the number of random variables and on the order P of the multivariate polynomials Ψₚ as follows: Q = (L + P)!/(L!P!) [18]. According to the probability distribution of the random variables, different sets of orthogonal polynomials can be selected for optimal convergence of expansion (2.4) [37, 27]. For example, Legendre polynomials are the appropriate choice for uniformly distributed random variables. Hermite polynomials are optimal in the case of Gaussian random variables.

Next, the stochastic Galerkin finite element discretization of (2.1) follows the methodology described in [14, 3]. Below, we summarize the discretization of the stochastic part of the problem. We start with the discretization of the random input, followed by a discussion on the discretization of the random solution. Next, the stochastic PDE is converted into a set of deterministic PDEs by applying the Galerkin condition. Standard spatial discretization techniques can then be applied.

First, the stochastic part of \( L(x, \omega) \) is represented by a finite number of random variables. Most commonly, Karhunen–Loève (KL)-expansions [19] and generalized polynomial chaos (gPC)-expansions [36] are applied to stochastic coefficients \( \alpha(x, \omega) \) appearing in \( L(x, \omega) \). In general, a stochastic coefficient \( \alpha(x, \omega) \) can be discretized as

\[
\alpha(x, \omega) \approx \hat{\alpha}(x, \xi) = \sum_{i=1}^{S} \alpha_i(x) \varphi_i(\xi),
\]

with \( \xi \) a vector of \( L \) random variables and \( \varphi_i(\xi) \) a function of \( \xi \). The truncation error of the expansion is determined by \( S \). For example, in case of a KL-expansion, \( \varphi_i(\xi) = \xi_i \) and \( S = L + 1 \) with \( \xi_0 = 1 \); in case of a gPC-expansion, \( \varphi_i(\xi) \) corresponds to an orthogonal polynomial in \( \xi_i \), and \( S \) equals the total number of orthogonal, \( L \)-dimensional polynomials of a certain order \( P_\alpha \). Applying this procedure to random coefficients in \( L(x, \omega) \), we can approximate the stochastic PDE (2.1) by

\[
\sum_{i=1}^{S} \hat{L}_i(x) \varphi_i(\xi) u(x, \omega) = f(x),
\]

with \( \hat{L}_i(x) \) deterministic differential operators. For example, if \( L(x, \omega) \) is a diffusion operator with stochastic diffusion coefficient \( \alpha(x, \omega) \), then \( \hat{L}_i(x) \) is a diffusion operator with coefficient \( \alpha_i(x) \).

The solution \( u(x, \omega) \) is approximated by a gPC-expansion,

\[
u(x, \omega) \approx \hat{u}(x, \xi) = \sum_{q=1}^{Q} u_q(x) \Psi_q(\xi) = \Psi^T u(x),
\]

with \( \Psi = [\Psi_1(\xi), \ldots, \Psi_Q(\xi)]^T \) a set of orthogonal polynomials defined in the \( L \) random variables appearing in (2.3). Note that we shall assume that these random variables are independent. The number of polynomials \( Q \) in the expansion depends on the number of random variables and on the order \( P \) of the multivariate polynomials \( \Psi_q \) as follows: \( Q = (L + P)!/(L!P!) \) [18]. According to the probability distribution of the random variables, different sets of orthogonal polynomials can be selected for optimal convergence of expansion (2.4) [37, 27]. For example, Legendre polynomials are the appropriate choice for uniformly distributed random variables. Hermite polynomials are optimal in the case of Gaussian random variables.

Next, the stochastic PDE is converted into a set of deterministic PDEs. The stochastic Galerkin condition is applied to (2.3), with \( u(x, \omega) \) replaced by (2.4). That
is, orthogonality of the residual with respect to the polynomial chaos is imposed:
\[
\sum_{i=1}^S \sum_{q=1}^Q \hat{L}_i(x) u_q(x) \int_{\Gamma} \varphi_i(y) \Psi_k(y) \Psi_q(y) p(y) dy = f(x) \int_{\Gamma} \Psi_k(y) p(y) dy, \quad k = 1, \ldots, Q,
\]
with \(p(y)\) the joint probability density function of \(\xi\). We can rewrite (2.5) as
\[
\sum_{i=1}^S C_i \hat{L}_i(x) \begin{bmatrix} u_1(x) \\ \vdots \\ u_Q(x) \end{bmatrix} = f(x) c \quad \text{with} \quad C_i = \begin{bmatrix} \langle \varphi_1 \Psi_1 \Psi_1 \rangle & \cdots & \langle \varphi_1 \Psi_1 \Psi_Q \rangle \\ \vdots & \ddots & \vdots \\ \langle \varphi_1 \Psi_Q \Psi_1 \rangle & \cdots & \langle \varphi_1 \Psi_Q \Psi_Q \rangle \end{bmatrix}
\]
and \(c = [(\Psi_1) \ldots (\Psi_Q)]^T\). In (2.6), the dependence of \(\Psi_q\) and \(\varphi_i\) on \(\xi\) is dropped. Equation (2.6) corresponds to a coupled set of \(Q\) deterministic PDEs.

The spatial part of (2.6) is discretized by the standard FEM. Denote by \(K_i \in \mathbb{R}^{N \times N}\) the discretization matrix corresponding to operator \(\hat{L}_i(x)\), with \(N\) the number of spatial degrees of freedom. A system of algebraic equations of the following form results:
\[
\sum_{i=1}^S C_i \otimes K_i x = b \quad \text{with} \quad x, b \in \mathbb{R}^{NQ \times 1}.
\]
Vectors \(x\) and \(b\) are the discretized solution and right-hand side, respectively.

**2.2. Properties of the discretization matrices \(C_i\).** The matrices \(C_i\) in (2.6) fully characterize the discretization of the stochastic part of (2.1). For example, in the case of \(L\) random variables and the use of a truncated KL-expansion, we have
\[
C_i = \langle \xi_{i-1} \Psi \Psi^T \rangle \quad \text{with} \quad i = 1, \ldots, L + 1.
\]
When the stochastic input parameters (2.2) are discretized by a gPC-expansion of order \(P\), defined on \(L\) random variables, the matrices \(C_i\) correspond to
\[
C_i = \langle \Psi_i \Psi \Psi^T \rangle \quad \text{with} \quad i = 1, \ldots, Q_\alpha = \frac{(L + P\alpha)!}{L! P\alpha!}.
\]

**Remark 2.1.** The matrices \(C_i\) are generally not diagonal. However, when they are defined by (2.8), a special set of biorthogonal polynomials can be constructed so that \(C_i\), for \(i = 1, \ldots, L + 1\), is a diagonal matrix \([25, 3]\). These polynomials require \(Q = 2^{P+1}\) terms to represent the solution by expansion (2.4). As pointed out in [25], the number of deterministic PDEs in (2.6) then grows prohibitively quickly, and therefore we do not consider this approach further. Also note that a stochastic Galerkin discretization with biorthogonal polynomials coincides with a stochastic collocation method [2], so a discussion in the context of the collocation approach is more appropriate.

**2.2.1. Matrix elements.** Explicit formulas for the matrix elements of (2.8) can be constructed based on properties of the polynomials \(\Psi_q\). Consider a gPC of order \(P\), defined by \(L\) random variables \(\xi\) and composed of \(Q\) polynomials \(\Psi_q\). The polynomials \(\Psi_q\) are orthonormal with respect to the probability measure of \(\xi\), i.e.,
\[
\langle \Psi_q \Psi_k \rangle = \delta_{q,k},
\]
with $\delta_{q,k}$ the Kronecker delta. Each multivariate polynomial $\Psi_q$ corresponds to a product of one-dimensional orthogonal polynomials $\psi_j$,

$$
\Psi_q = \prod_{i=1}^{L} \psi_{\eta_{q,i}}(\xi_i).
$$

The multi-index $\eta_q = [\eta_{q,1} \ldots \eta_{q,L}]$ collects the degrees of the one-dimensional factors and satisfies $\sum_{i=1}^{L} \eta_{q,i} \leq P$. The polynomials $\psi_j(z)$ are characterized by a three-term recurrence relation [30],

$$
\psi_{j+1}(z) = (a_j z + b_j) \psi_j(z) - c_j \psi_{j-1}(z) \quad \forall j \geq 1,
$$

with $a_j, b_j,$ and $c_j$ constants and $a_j, c_j > 0$. For completeness, we shall also define $a_0$ and $b_0$ such that $\psi_1(z) = (a_0 z + b_0) \psi_0$, and $c_0 = 0$.

**Theorem 2.2.** Consider matrices $C_i$, defined by (2.8) and based on polynomials $\Psi_q$ (2.10)–(2.12). The diagonal elements of $C_i$ equal

$$
C_i(j,j) = \frac{-b_{q,j,i-1}}{a_{q,j,i-1}} \quad \text{for } j = 1, \ldots, Q \quad \text{and} \quad i = 2, \ldots, L + 1.
$$

The off-diagonal elements, $C_i(j,k)$ for $j, k = 1, \ldots, Q$ with $j \neq k$ and $i = 2, \ldots, L + 1$, satisfy

$$
C_i(j,k) = \frac{1}{a_{q,j,i-1}} \left( \delta_{q,j,i-1+1,q,k,i-1+1} + c_{q,j,i-1} \delta_{q,j,i-1,q,k,i-1+1} \right) \prod_{m=1,m \neq i-1}^{L} \delta_{q,j,m,q,k,m}.
$$

**Proof.** The $(j,k)$th element of $C_i$ is defined as $\langle \xi_{i-1} \Psi_j \Psi_k \rangle$. For notational convenience, we replace index “$i-1$” by “$r$” in the remainder of the proof. Based on (2.10)–(2.12), we find

$$
\langle \xi_{r} \Psi_{j} \Psi_{k} \rangle = \prod_{m=1}^{L} \langle \xi_{r} \psi_{j,m} (\xi_{m}) \psi_{k,m} (\xi_{m}) \rangle = \prod_{m=1,m \neq r}^{L} \delta_{q,j,m,q,k,m} \left( a_{q,j,r} \right)^{-1} \left( b_{q,j,r} \psi_{j,r} + c_{q,j,r} \psi_{k,r} - \psi_{j,r+1} - \psi_{k,r+1} \right) \prod_{m=1,m \neq r}^{L} \delta_{q,j,m,q,k,m}.
$$

For $j = k$, (2.13) follows immediately. For $j \neq k$, one has $b_{q,j,r} \delta_{q,j,r,q,k,r} \prod_{m \neq r} \delta_{q,j,m,q,k,m} = 0$, and hence (2.14).

**Example 2.3.** For normalized Legendre polynomials, the recurrence (2.12) is given by

$$
\frac{n+1}{\sqrt{2n+3}} \psi_{n+1}(z) = \frac{2n+1}{\sqrt{2n-1}} \psi_{n}(z) - \frac{n}{\sqrt{2n-1}} \psi_{n-1}(z),
$$

with $\psi_0(z) = 1$ and $\psi_1(z) = \sqrt{3}$. Evaluating (2.14), we find

$$
\langle \xi_{i-1} \Psi_{j} \Psi_{k} \rangle = \frac{n_i \delta_{q,j,i-1+1,q,k,i-1+1} + \delta_{q,j,i-1,q,k,i-1+1}}{\sqrt{(2n_{j,i-1} + 1)(2n_{k,i-1} + 1)}} \prod_{m=1,m \neq (i-1)}^{L} \delta_{q,j,m,q,k,m}.
$$
Normalized Hermite polynomials satisfy \( \sqrt{n+1} \psi_{n+1}(z) = z \psi_n(z) - \sqrt{n} \psi_{n-1}(z) \), with \( \psi_0(z) = 1 \) and \( \psi_1(z) = z \). Theorem 2.2 produces the same result for \( \langle \xi_{i-1} \Psi \Psi^T \rangle \) in the case of a Hermite chaos as reported in [29, 22].

**Corollary 2.4.** The diagonal elements of matrices \( C_i \) defined by (2.8), with \( i = 2, \ldots, L+1 \) and a \( P \)-th-order chaos, are zero if the coefficients \( b_j \) in (2.12) equal zero for \( j = 0, \ldots, P \).

**Example 2.5.** From Corollary 2.4 it follows that \( \text{diag}(C_i) \equiv 0 \), for \( C_i \) defined by (2.8) and \( i = 2, \ldots, L+1 \), in case of a Legendre, Hermite, or Chebyshev chaos, but not in case of a Laguerre or Jacobi chaos. This can straightforwardly be deduced from the recurrence relations for the respective one-dimensional orthogonal polynomials; see Example 2.3 and [11].

**2.2.2. Hierarchical structure.** The matrices \( C_i \) possess a hierarchical structure [13, 21]. Given a \( P \)-th-order chaos, each \( C_i \in \mathbb{R}^{Q_i \times Q_i} \) can be written in block form as

\[
C_i = \begin{bmatrix}
\hat{C}_i & F_i^T \\
F_i & D_i
\end{bmatrix}, \quad \hat{C}_i \in \mathbb{R}^{Q_i \times Q_i}, \quad F_i \in \mathbb{R}^{Q_h \times Q_i}, \quad D_i \in \mathbb{R}^{Q_h \times Q_h},
\]

with \( Q = Q_i + Q_h = \binom{L+P+1}{L+P} \) and \( Q_i = \binom{L+P-1}{L+P-1} \). Submatrix \( \hat{C}_i \) is defined similarly as \( C_i \), and corresponds to a chaos of order \( P-1 \). By recursion, \( \hat{C}_i \) has a structure similar to that of (2.16). The recursion terminates with \( \hat{C}_i \in \mathbb{R}^{1 \times 1} \), corresponding to a zeroth-order polynomial chaos. This hierarchical structure can be exploited in the construction of iterative solvers; see sections 3.1.3 and 3.3. The following corollary enables one to reduce the computational work.

**Corollary 2.6.** Let matrix \( C_i \) be defined by (2.8) and based on a \( P \)-th-order chaos with \( L \) random variables. Then all off-diagonal elements of subblock \( D_i \) (2.16) equal zero.

**Proof.** The off-diagonal elements of \( D_i \) are defined by (2.14), with \( j \neq k \) and

\[
\sum_{i=1}^{L} \eta_{j,i} = \sum_{i=1}^{L} \eta_{k,i} = P.
\]

Consider the product term \( \prod_{m=1, m \neq (i-1)}^{L} \delta_{\eta_{j,m}=\eta_{k,m}} \) in (2.14). This term is either zero or one. The latter case implies that \( \eta_{j,m} = \eta_{k,m} \), for all \( m = 1, \ldots, L \), with \( m \neq i-1 \). From (2.17) it then follows that \( \eta_{j,i-1} = \eta_{k,i-1} \). However, this corresponds to \( \eta_{j} = \eta_{k} \) and thus \( j = k \), which is not possible for an off-diagonal element. Thus, the product term must equal zero. \( \square \)

**3. A classification of iterative solvers for the stochastic finite element method.**

**3.1. One-level methods based on matrix splitting.** The tensor product system (2.7) can be solved iteratively, based on matrix splitting techniques. Applied to a system \( Ax = b \), a matrix splitting method uses \( A = A^+ + A^- \) to compute an updated solution \( x^{\text{new}} \) from \( x^{\text{old}} \) as

\[
A^+ x^{\text{new}} = b - A^- x^{\text{old}}.
\]

The convergence rate of iteration (3.1) is given by \( -\log_{10}(\rho) \), with \( \rho \) the convergence factor defined as the spectral radius of the iteration operator \( -(A^+)^{-1}A^- \) [34].
The system matrix (2.7) can be split globally or according to its block structure. Global splitting methods are presented in section 3.1.1. Based on the block structure of (2.7), we consider a splitting of the stiffness matrices $K_i$, $K_i = K_i^+ + K_i^-$, in section 3.1.2. A second approach is based on splitting the $C_i$-matrices, $C_i = C_i^+ + C_i^-$, and is dealt with in section 3.1.3.

Remark 3.1. Note that the tensor product system matrix (2.7) does not need to be stored in memory explicitly when an iterative solver is used. It suffices to store the matrices $C_i$ and $K_i$ separately, as explained in [23]. The storage is proportional to $N$ for each $K_i$-matrix, and to $Q$ for each $C_i$-matrix. The latter holds for $C_i$ defined by (2.8), as follows from Theorem 2.2.

3.1.1. Basic splitting methods. By decomposing matrix $A$ into its strictly lower triangular part $A_L$, strictly upper triangular part $A_U$, and diagonal $A_D$, a variety of basic iterative methods can be constructed. We shall consider the lexicographic Gauss–Seidel (GSLEX) splitting in more detail. Applied to (2.7) and using the notation of (3.1), this corresponds to

$$A^+ = \sum_{i=1}^{S} (C_i^L + C_i^D) \otimes (K_i^L + K_i^D) + \sum_{i=1}^{S} C_i^L \otimes K_i^U,$$

with $C_i = C_i^L + C_i^D + C_i^U$ and $K_i$ decomposed analogously. Taking Remark 3.1 into account, one can see that a GSLEX-iteration requires a number of multiplications and additions that is proportional to $NQ$. The latter holds for $C_i$ defined by (2.8), as follows from Theorem 2.2.

3.1.2. Splitting of the $K_i$-matrices. When only the $K_i$-matrices in (2.7) are decomposed, the following iteration results:

$$\sum_{i=1}^{S} C_i \otimes K_i^+ x^{\text{new}} = b - \sum_{i=1}^{S} C_i \otimes K_i^- x^{\text{old}}.$$

Different $K_i$-splittings are discussed in [25]. When $K_i^+$ corresponds to the lower triangular part of $K_i$, iteration (3.3) is a collective Gauss–Seidel method. In that case, one iteration requires the solution of $N$ systems of size $Q \times Q$. In [23], it is shown that the system matrices, $\sum_{i=1}^{S} K_i(r,r)C_i$, $r = 1, \ldots, N$, are in general well conditioned. Hence a limited number of conjugate gradient (CG) steps suffices to solve each block system. The cost of one collective Gauss–Seidel iteration is then proportional to $NQs_{\text{CG}}$, with $s_{\text{CG}}$ the required number of CG iterations.

3.1.3. Splitting of the $C_i$-matrices. A $C_i$-splitting leads to the iteration

$$\sum_{i=1}^{S} C_i^+ \otimes K_i x^{\text{new}} = b - \sum_{i=1}^{S} C_i^- \otimes K_i x^{\text{old}}.$$

Figure 3.1 shows five different constructions of $C_i^+$. In case the conditions of Corollary 2.4 hold, iteration (3.4) with a Jacobi-type $C_i$-splitting simplifies to

$$I_Q \otimes K_1 x^{\text{new}} = b - \sum_{i=2}^{S} C_i \otimes K_i x^{\text{old}}.$$

The block Jacobi and block Gauss–Seidel splittings, illustrated in Figure 3.1(e)–(d), are based on the hierarchical block structure (2.16) of the $C_i$-matrices. Note that in

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.
case Corollary 2.6 holds, the off-diagonal elements of the $D_i$-subblock of $C_i$ (2.16) are all zero. The block Gauss–Seidel and block Jacobi $C_i$-splittings coincide then with the simple Gauss–Seidel and Jacobi splittings, respectively, shown in Figures 3.1(b) and 3.1(a). In the circulant $C_i$-splitting approach, represented in Figure 3.1(e), $C_i^+$ is constructed as the circulant matrix that minimizes the Frobenius norm $\|C_i^+ - C_i\|_F$ [6, 33].

The Gauss–Seidel $C_i$-splitting (Figure 3.1(b)) requires in every iteration the solution of $Q$ systems of size $N \times N$. In case of a diffusion operator $L(x, \omega)$, these systems can be solved by CG, preconditioned with a standard multigrid cycle [32]. The block Gauss–Seidel $C_i$-splitting (Figure 3.1(d)) requires the solution of $P + 1$ smaller tensor product systems, with $P$ the chaos order. In case of a diffusion problem, these systems can be solved by CG, preconditioned with the multigrid method discussed in section 3.2. The circulant splitting approach can be implemented efficiently by using FFT techniques [5]. Applying an FFT transformation decouples (3.4) into $Q$ independent systems of size $N \times N$. Hence the total cost becomes similar to the cost of a Gauss–Seidel $C_i$-splitting iteration. The construction of the circulant matrices can be done during a set-up phase, at a negligible cost.

3.1.4. Some remarks on splitting methods. The implementation of Gauss–Seidel-type splitting methods is almost trivial once all matrix elements of $C_i$ and $K_i$ are known. Also block splitting methods do not pose problems as the blocks are based on the Kronecker product structure of (2.7). When simulation code is available for similar deterministic PDEs, block splitting methods based on $C_i$-splittings, for example, method (3.5), can reuse this code and the $K_i$-matrices need not be stored explicitly, given a matrix-vector product routine. As in the case of the stochastic collocation method, nonintrusive code could be developed for the stochastic Galerkin method by using a $C_i$-splitting solution method. The other splitting approaches do not possess this property.

A block splitting iteration results in a larger computational cost than a basic splitting method because of the repeated solving of smaller systems. The cost of a $K_i$-splitting, however, becomes proportional to the cost of a basic splitting iteration when only a limited number of CG steps are required for solving the $C_i$-blocks, as mentioned in section 3.1.2. With respect to $C_i$-splitting methods, $Q$ deterministic simulations are required for one Gauss–Seidel $C_i$-splitting iteration. Thus when many iterations are required, this approach will rapidly lead to a larger number of simulations, than, for example, a stochastic collocation approach. However, by solving the $N \times N$-blocks inexactly, as is done in some of the following methods, this computational cost problem is no longer present.
3.2. Multigrid methods in the spatial dimension. Multigrid algorithms lead to very efficient and robust solvers for many deterministic problems [32] and can be extended to the system (2.7). By using the iterative methods discussed in section 3.1 as smoothers in a standard multigrid framework, a very robust solver can be built; see [18, 8, 25, 23]. The multigrid coarsening is performed on the spatial discretization; that is, prolongation operators are constructed as $I_Q \otimes P_{K_1}$, with $P_{K_1}$ a prolongation operator based on $K_1$ (2.7).

To quantify the computational cost, we shall consider geometric multigrid. We assume that the number of gridpoints on level $l$ corresponds to $1/2^d$ times the number of points on level $l-1$, with $l = 0$ the finest level and $d$ the number of spatial dimensions. The total cost of a $V(1,1)$-cycle is then well approximated by $2^{d-2} c_s$ [4], with $c_s$ the cost of one smoothing iteration.

3.3. Multilevel methods in the stochastic dimension. In case a high-order polynomial chaos is used, the random dimension $Q$ may become very large. A coarsening in random space may then be appropriate. Based on the hierarchical structure of the matrices $C_i$ (2.16), the random dimension can be coarsened [17]. A set of $P + 1$ levels can be constructed, where $P$ is the chaos order. At the coarsest level, corresponding to $P = 0$, the system matrix reduces to $K_1$. To transfer a vector at level $p$ to the next coarse level $p-1$, injection can be used. This corresponds to the following restriction and prolongation operators:

$$
P^{-1}_{p-1} = [I_{Q_i}, 0] \otimes I_N \quad \text{and} \quad P_{p-1}^p = \begin{bmatrix} I_{Q_i} & 0 \\ 0 & I_N \end{bmatrix} \otimes I_N.
$$

The multilevel method is summarized in Algorithm 1. The parameter $\gamma$ determines the recursion scheme; for example, the case $\gamma = 1$ is called a $V$-cycle, and the case $\gamma = 2$ a $W$-cycle. In Algorithm 1, we apply at every level a few steps of $C_i$-splitting iteration (3.4). When $\nu_1 = \nu_2 = 1$ and $\gamma = 1$, i.e., in case of a $V(1,1)$-cycle, each iteration requires solving $2Q - 1$ $(N \times N)$-systems, with $Q = (L+1+P)/((L+1)!)$.

Other iterative methods can also be applied instead of the $C_i$-splitting smoother. Various approaches will be compared in section 4.3. A combined coarsening in the spatial and stochastic dimensions can be achieved by using one multigrid cycle as smoothing step in Algorithm 1. This will be demonstrated by the numerical experiments in section 5.

Remark 3.2. Algorithm 1 is related to $p$-multigrid methods for $p$-finite element problems. These, however, would use a $K_i$-splitting iteration at the first and last steps of Algorithm 1 [9].

3.4. Krylov subspace methods. In practice, the methods presented above will typically be used as preconditioners for a Krylov method. When the system (2.7) is symmetric and positive definite, CG can be used. This is the case, for example, for the discretization of a diffusion operator with a random diffusion coefficient that is bounded and uniformly coercive [3].

The basic splitting method (3.2) and the $K_i$-splitting iteration (3.3) perform poorly as preconditioners. Their spectrum does not approximate the spectrum of (2.7) well, as will be verified by the local Fourier analysis in section 4.

The $C_i$-splitting methods are appropriate as preconditioners, as long as the Hermite polynomial chaos order remains low. These preconditioners can be approximated by solving the $Q$ $(N \times N)$-block systems in exactly. In the numerical experiments in section 5, we shall use a standard $V(1,1)$-multigrid cycle to approximately solve these systems.
**Algorithm 1. Multilevel Solution Approach.**

\[ x^{\text{new}} = \text{multilevel}(x^{\text{old}}, b, [C_1, \ldots, C_S], p) \]

1. apply \( \nu_1 \) steps of the \( C_i \)-splitting iteration (3.4) starting from \( x^{\text{old}} \) to compute \( \bar{x} \),
2. restrict residual: \( \hat{b} = P_{p-1}^{-1}(b - \sum_{i=1}^{S} C_i \otimes K_i \bar{x}) \),
3. coarse level solve: solve \( \sum_{i=1}^{S} \hat{C}_i \otimes K_i \hat{v} = \hat{b} \)
   - if \( p = 1 \), \( \hat{v} = K_1^{-1} \hat{b} \),
   - if \( p > 1 \), \( \hat{v} = \text{multilevel}([0 \ldots 0]^T, \hat{b}, [\hat{C}_1, \ldots, \hat{C}_S], p-1) \),
4. prolongate correction and update solution: \( \tilde{x} = x^{\text{old}} + P_{p-1} \hat{v} \),
5. apply \( \nu_2 \) steps of the \( C_i \)-splitting iteration (3.4) starting from \( \tilde{x} \) to compute \( x^{\text{new}} \).

A mean-based preconditioner is constructed as \( I_Q \times K_1 \) and occurs as a special case of Jacobi \( C_i \)-splittings (3.5). In [22] the spectrum of this preconditioner is analyzed.

Krylov acceleration is also required when multigrid is used in combination with an unstructured finite element spatial discretization. The multigrid method of section 3.2 is then of algebraic multigrid type [28, 23].

4. Convergence analysis. To analyze the convergence properties of the iterative methods presented in section 3, we conduct a local Fourier analysis (LFA) [32]. LFA considers linear discrete operators with constant coefficients defined on an infinite grid. A general discrete operator with nonconstant coefficients can be analyzed through local linearization and replacement by an operator with constant coefficients.

We shall apply our analysis to the following two-dimensional model problem:

\[
\frac{\partial^2 u(x, \omega)}{\partial x^2} + \alpha(x, \omega) \frac{\partial^2 u(x, \omega)}{\partial y^2} = f(x).
\]

The effect of boundary conditions is neglected due to the infinite grid assumption. We shall apply the SFEM described in section 2 to (4.1). The spatial discretization uses a standard five-point finite difference scheme on a rectangular grid \( G_h = \{(jh, kh)\}_{j,k \in \mathbb{Z}} \) with grid spacing \( h \) in \( x \)- and \( y \)-directions. This results in a set of discretized equations,

\[
\begin{align*}
(u_{j-1,k} - 2u_{j,k} + u_{j+1,k}) + M_{j,k}(u_{j,k-1} - 2u_{j,k} + u_{j,k+1}) &= h^2 f_{j,k}(\Psi),
\end{align*}
\]

where \( u_{j,k} \) denotes the \( Q \)-vector of unknown gPC coefficients (2.4) at grid point \((jh, kh)\) and \( M_{j,k} \) is the \( Q \times Q \) matrix defined as

\[
M_{j,k} = \sum_{i=1}^{S} \alpha_i(jh, kh) \langle \varphi_i(\xi) \Psi \Psi^T \rangle = \sum_{i=1}^{S} \alpha_i(jh, kh) C_i.
\]

Equation (4.2) can be represented by a linear discrete operator \( L_h \). This operator is invariant to Fourier grid mode components of the form

\[
\phi_{j,k}(\theta, z) = \exp(i(j\theta_x + k\theta_y)) z,
\]
with \( z = [1 \ldots 1]^T \in \mathbb{R}^Q, \theta = (\theta_x, \theta_y) \in [-\pi, \pi)^2 \), and \( i \) the imaginary unit. This gives \( L_h \phi_{j,k}(\theta) = \tilde{L}^{j,k}_h(\theta) \phi_{j,k}(\theta) \) with \( \tilde{L}^{j,k}_h(\theta) \) called the symbol of \( L_h \),

\[
\tilde{L}^{j,k}_h(\theta) = \frac{1}{h^2} \left( I_Q(\exp(-i\theta_x) - 2 + \exp(i\theta_x)) + M_{j,k}(\exp(-i\theta_y) - 2 + \exp(i\theta_y)) \right)
\]

(4.5)

To numerically verify the accuracy of our analysis, we shall consider three configurations for \( \alpha \): \( \alpha_g(x, \omega) \), \( \alpha_u(x, \omega) \), and \( \alpha_{\text{log}}(x, \omega) \). The random field \( \alpha_g(x, \omega) \) models a Gaussian field that is discretized by a KL-expansion based on an exponential covariance function (5.5) with variance \( \sigma^2 \) and correlation length \( L_c \). The random field \( \alpha_u(x, \omega) \) is constructed by a similar KL-expansion, but with uniformly distributed random variables. The solution \( u(x, \omega) \) is then discretized by a Legendre chaos, while \( \alpha_{\text{log}}(x, \omega) \) leads to a Hermite chaos expansion of the solution. The lognormal random field \( \alpha_{\text{log}}(x, \omega) \), defined as \( \exp(\alpha_g(x, \omega)) \), is discretized by a Hermite chaos expansion as described in [12]. For the problem to be well-defined [20], the chaos expansion order must equal twice the order of the Hermite chaos expansion for the solution.

The numerical results will report the chaos order \( P \) of the solution and the statistics, i.e., variance and correlation length, of the underlying Gaussian field \( \alpha_g(x, \omega) \). The corresponding mean \( \mu_l \) and variance \( \sigma^2 \) of the lognormal field \( \alpha_{\text{log}}(x, \omega) \) can be calculated, respectively, as \( \exp(\mu_g + \sigma^2/2) \) and \( \mu_l^2(\exp(\sigma^2) - 1) \), with \( \mu_g \) and \( \sigma^2 \) the mean and variance of \( \alpha_g(x, \omega) \).

Remark 4.1: When using the Gaussian field \( \alpha_g \) in the LFA and the numerical experiments of section 5, the positive definiteness of the linear systems (2.7) and (4.2) is no longer guaranteed since \( \alpha(x, \omega) \) may become unbounded. We shall, however, make the generally used assumption [10, 14, 37] that by using a sufficiently small variance, sensible solutions can be obtained. In practice, we indeed observe positive definiteness of the systems (2.7) and (4.2) when the variance is small enough. For large variances, problems can occur; see Tables 4.1, 4.3, and 4.6.

4.1. One-level methods. To compute asymptotic convergence factors for the methods of section 3.1, the action of iteration operators onto the Fourier modes is investigated. Denote an iteration operator by \( S_h \) and its symbol by \( \tilde{S}^{j,k}_h(\theta) \). When the variation of the coefficients \( \alpha_i \) is sufficiently smooth [15, section 8.2.2], the asymptotic convergence factor is defined as

\[
\rho = \max_{x = (j,h,k) \in G_h} \max_{\theta \in [-\pi, \pi)^2} \rho(\tilde{S}^{j,k}_h(\theta)),
\]

(4.6)

with \( \rho(\tilde{S}^{j,k}_h(\theta)) \) the spectral radius of \( \tilde{S}^{j,k}_h(\theta) \). For practical computations at finite grids with Dirichlet boundary conditions, accurate estimates for the convergence factor can be obtained from (4.6) by sampling \( \theta \)-values from the finite grid \( \Theta^D := \{ \theta \mid \theta = (\theta_x, \theta_y) \) with \( \theta_d = \pi n_d / n_d, 1 \leq n_d \leq n_d - 1, n_d = 1/h \} \) [35].

The smoothing factor is another important measure that can obtained by LFA. It is used in a multigrid context to assess the quality of a smoothing method and quantifies the reduction rate of the high frequency Fourier modes [32]. It is defined as

\[
s = \max_{x = (j,h,k) \in G_h} \max_{\theta \in [-\pi, \pi)^2} \rho(\tilde{S}^{j,k}_h(\theta)) .
\]
4.1.1. Basic splitting methods. The Gauss–Seidel splitting iteration (3.2) is characterized by the iteration operator \( S_{GS,h}^{j,k} = -(L_{GS,h}^+)^{-1}L_{GS,h}^- \), with corresponding symbols

\[
\tilde{L}_{GS,h}^{(j,k)+}(\theta) = (\exp(-i\theta_x) - 2)IQ + (\exp(-i\theta_y) - 2)(M_{j,k}^I + M_{j,k}^D) + \exp(i\theta_y)M_{j,k}^L,
\]

\[
\tilde{L}_{GS,h}^{(j,k)-}(\theta) = \exp(i\theta_x)IQ + \exp(i\theta_y)M_{j,k}^D + (\exp(-i\theta_y) - 2 + \exp(i\theta_y))M_{j,k}^U,
\]

where \( M_{j,k} = M_{j,k}^I + M_{j,k}^D + M_{j,k}^U \) is decomposed into its strictly lower triangular part, diagonal part, and strictly upper triangular part. The asymptotic convergence factor of GSLEX can be computed from (4.6), with \( \tilde{S}_{GS,h}^{j,k}(\theta) = -(\tilde{L}_{GS,h}^{(j,k)+}(\theta))^{-1}\tilde{L}_{GS,h}^{(j,k)-}(\theta) \).

The largest eigenvalue in modulus is obtained at \((\theta_x, \theta_y) = (\pi h, \pi h)\).

A similar convergence analysis can be applied for other splittings. For example, in case of a damped Jacobi-splitting with damping factor \( \omega_{JAC} \), the operator \( S_{JAC,h}^{j,k} = -(L_{JAC,h}^+)^{-1}L_{JAC,h}^- \) is characterized by the symbols

\[
\tilde{L}_{JAC,h}^{(j,k)+}(\theta) = -\frac{2}{\omega_{JAC}}(IQ + M_{j,k}^D),
\]

\[
\tilde{L}_{JAC,h}^{(j,k)-}(\theta) = 2\cos(\theta_x)IQ + 2\left(\frac{1}{\omega_{JAC}}M_{j,k}^D - M_{j,k}^I + \frac{1}{\omega_{JAC}} - 1\right)IQ + 2\cos(\theta_y)M_{j,k}^L.
\]

The largest eigenvalue in modulus of \( \tilde{S}_{JAC,h}^{j,k}(\theta) \) is again obtained at \((\pi h, \pi h)\).

The GSLEX convergence properties are illustrated in Table 4.1. The table shows the theoretical and numerically observed convergence factors when solving (4.1) with (3.2) in case of a Gaussian field and a lognormal field.

**Remark 4.2.** The results for the lognormal field, shown between brackets, behave similarly to those for the Gaussian field. Hence, the convergence of GSLEX is not influenced by the definition of \( C_i \), (2.8), or (2.9). Note that in case of a lognormal field, there is no problem when the variance increases, as the positivity of \( \alpha(x, \omega) \) is guaranteed.

**Remark 4.3.** From \( M_{j,k} \) appearing in (4.7) and (4.8), a dependency of the convergence on the stochastic discretization is expected. Table 4.1 shows, however, that this effect is very limited.

**Remark 4.4.** The convergence of GSLEX is strongly dependent on \( h \): this prevents a direct use of GSLEX in practical computations.

Table 4.2 displays convergence and smoothing factors for GSLEX and damped Jacobi applied to an isotropic and an anisotropic problem. The damping factor \( \omega_{JAC} \) was chosen equal to 2/3. The convergence of both methods slightly improves for the anisotropic problem.

4.1.2. Splitting of the \( K_i \)-matrices. The convergence of \( K_i \)-splitting iterations (3.3) applied to problem (4.1) has been analyzed in [25]. In the collective Gauss–Seidel case, the analysis shows that the asymptotic convergence factor is independent of the stochastic discretization,

\[ \rho_{GS-K_i} = 1 - \pi^2 h^2 + O(h^4). \]

This method is not useful as a stand-alone method due to the strong dependence of the convergence factor on the grid spacing \( h \). In [25], it was also shown that the smoothing factor of (3.3) is independent of \( h \). It does, however, depend on the eigenvalues of \( M_{j,k} \) (4.3).
Theoretical ($\rho_{\text{theo}}$) and numerically observed convergence factors ($\rho_{\text{num}}$) for lexicographic Gauss–Seidel (3.2) applied to (4.1), with $\alpha(x,\omega)$ a Gaussian field $\alpha_{\omega}(x,\omega)$. Between brackets, the results are shown for the case of a lognormal field $\alpha_{\text{log}}(x,\omega)$. The discretization uses by default $h = 2^{-5}$, $L = 4$, $P = 2$, $\sigma = 0.2$, $L_c = 1$. In each row of the table, one of these parameters is changed.

<table>
<thead>
<tr>
<th>Grid spacing $h$</th>
<th>$2^{-4}$</th>
<th>$2^{-5}$</th>
<th>$2^{-6}$</th>
<th>$2^{-7}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1 - \rho_{\text{theo}} = 10^{-3} \times x$</td>
<td>30.65 (28.87)</td>
<td>7.988 (7.470)</td>
<td>2.019 (1.887)</td>
<td>0.5060 (0.4731)</td>
</tr>
<tr>
<td>$1 - \rho_{\text{num}} = 10^{-3} \times x$</td>
<td>32.74</td>
<td>8.247</td>
<td>2.347</td>
<td>0.6329</td>
</tr>
</tbody>
</table>

Random variables $L$

| $1 - \rho_{\text{theo}} = 10^{-3} \times x$ | 8.102 (7.545) | 8.016 (7.514) | 7.988 (7.470) | 7.967 (7.426) |
| $1 - \rho_{\text{num}} = 10^{-3} \times x$ | 8.276 | 8.262 | 8.247 | 8.230 |

Order PC-basis $P$

| $1 - \rho_{\text{theo}} = 10^{-3} \times x$ | 8.625 (8.270) | 7.464 (6.867) | 7.009 (6.380) | 6.602 (5.971) |
| $1 - \rho_{\text{num}} = 10^{-3} \times x$ | 9.096 | 7.827 | 7.375 | 7.052 |

Standard deviation $\sigma$

| $1 - \rho_{\text{theo}} = 10^{-3} \times x$ | 8.741 (8.439) | 6.481 (5.757) | 4.971 (4.336) | $\rho_{\text{num}} > 1$ (3.738) |
| $1 - \rho_{\text{num}} = 10^{-3} \times x$ | 9.093 | 6.847 | 5.395 | $\rho_{\text{num}} > 1$ |

Correlation length $L_c$

| $1 - \rho_{\text{theo}} = 10^{-3} \times x$ | 8.336 (7.955) | 8.124 (7.665) | 7.907 (7.330) | 7.881 (7.277) |
| $1 - \rho_{\text{num}} = 10^{-3} \times x$ | 8.896 | 8.476 | 8.079 | 8.017 |

Table 4.2: Theoretical convergence factors ($\rho_{\text{theo}}$), numerically observed convergence factors ($\rho_{\text{num}}$), and smoothing factors ($s_{\text{theo}}$) for $\omega$-Jacobi and lexicographic Gauss–Seidel (3.2) applied to (4.1), with $\alpha(x,\omega)$ a Gaussian or lognormal random field ($\omega = 2/3$, $h = 2^{-5}$, $L = 4$, $L_c = 1$).

<table>
<thead>
<tr>
<th>Gaussian field</th>
<th>1 - $\rho_{\text{theo}}$</th>
<th>1 - $\rho_{\text{num}}$</th>
<th>$s_{\text{theo}}$</th>
<th>1 - $\rho_{\text{theo}}$</th>
<th>1 - $\rho_{\text{num}}$</th>
<th>$s_{\text{theo}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\omega$-Jacobi</td>
<td>$\omega$-Jacobi</td>
<td>$\omega$-Jacobi</td>
<td>$\omega$-Jacobi</td>
<td>$\omega$-Jacobi</td>
<td>$\omega$-Jacobi</td>
<td>$\omega$-Jacobi</td>
</tr>
<tr>
<td>$\omega$-Jacobi</td>
<td>$\omega$-Jacobi</td>
<td>$\omega$-Jacobi</td>
<td>$\omega$-Jacobi</td>
<td>$\omega$-Jacobi</td>
<td>$\omega$-Jacobi</td>
<td>$\omega$-Jacobi</td>
</tr>
</tbody>
</table>

4.1.3. Splitting of the $C_i$-matrices. The $C_i$-splitting (3.4) applied to (4.2) yields the following iteration for the error, $e_{j,k}^{\text{new}} = u_{j,k} - u_{j,k}^{\text{new}}$.

$$
\begin{align*}
(e_{j,k}^{\text{new}} - 2e_{j,k}^{\text{old}} + e_{j+1,k}^{\text{old}} + M_{j,k}^+ (e_{j,k-1}^{\text{new}} - 2e_{j,k}^{\text{new}} + e_{j,k+1}^{\text{new}}))
&= -M_{j,k}^+ (e_{j,k-1}^{\text{old}} - 2e_{j,k}^{\text{old}} + e_{j,k+1}^{\text{old}}).
\end{align*}
$$

This corresponds to an iteration operator $S_{C_i}$ with symbol $\tilde{S}_{C_i,k}^j(\theta)$ given by

$$
\tilde{S}_{C_i,k}^j(\theta) = -\left(\gamma_i I + M_{j,k}^+\right)^{-1} M_{j,k}^- \quad \text{and} \quad \gamma_d = -4 \sin^2 \left(\frac{\theta_d}{2}\right), \quad d = x \text{ or } y.
$$

The largest eigenvalue in modulus of $\tilde{S}_{C_i,k}^j(\theta)$ occurs at $(\theta_x, \theta_y) = (\pi h, \pi (1 - h))$. Using a Taylor expansion around $h = 0$, this yields

$$
\frac{\gamma_x}{\gamma_y} \approx \frac{\pi^2 h^2}{4 - \pi^2 h^2}.
$$

Thus, the convergence is asymptotically independent of $h$ but depends on the stochastic discretization via $M_{j,k}$. Note that in this case, the smoothing and convergence factors coincide.
Table 4.3
Effect of the type of polynomials on the theoretical ($\rho_{\text{theo}}$) convergence factors for the Gauss–Seidel $C_i$-splitting iteration (3.4) applied to (4.2). The discretization uses $h = 2^{-5}$, $L = 4$, $P = 2$, $\sigma = 0.2$, $L_c = 1$. In each row, one of these parameters is changed.

<table>
<thead>
<tr>
<th>Grid spacing $h$</th>
<th>$\alpha_{(x,\omega)}$—Hermite chaos</th>
<th>$\alpha_{(x,\omega)}$—Legendre chaos</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2^{-6}$</td>
<td>$2^{-5}$</td>
<td>$2^{-6}$</td>
</tr>
<tr>
<td>$\rho_{\text{theo}} = 10^{-2}x$</td>
<td>9.930</td>
<td>10.09</td>
</tr>
<tr>
<td>Random variables $L$</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>$\rho_{\text{theo}} = 10^{-2}x$</td>
<td>8.624</td>
<td>10.38</td>
</tr>
<tr>
<td>Order PC-basis $P$</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>$\rho_{\text{theo}} = 10^{-2}x$</td>
<td>3.363</td>
<td>18.33</td>
</tr>
<tr>
<td>Standard deviation $\sigma$</td>
<td>0.1</td>
<td>0.4</td>
</tr>
<tr>
<td>$\rho_{\text{theo}} = 10^{-2}x$</td>
<td>2.522</td>
<td>40.36</td>
</tr>
<tr>
<td>Correlation length $L_c$</td>
<td>0.25</td>
<td>0.5</td>
</tr>
<tr>
<td>$\rho_{\text{theo}} = 10^{-2}x$</td>
<td>5.966</td>
<td>8.350</td>
</tr>
</tbody>
</table>

The spectral radius (4.6), with $\tilde{S}_{h,k}$ given by (4.9), can be simplified in case of a Jacobi $C_i$-splitting (3.5). Consider $M_{j,k}^+ = \text{diag}(M_{j,k})$ and $\alpha(x,\omega)$ discretized by a truncated KL-expansion so that $\alpha_{j,k}(\omega) \approx \alpha_{j,k}^{(1)} + \sum_{i=2}^{L+1} \alpha_{j,k}^{(i)} \xi_{i-1}$. The matrices $M_{j,k}^+$ and $M_{j,k}^-$ then correspond, respectively, to $\alpha_{j,k}^{(1)} I_Q$ and $\sum_{i=2}^{L+1} \alpha_{j,k}^{(i)} C_i$, with $C_i$ defined by (2.8). Together with (4.9) and (4.6), this yields

$$\rho_{\text{Jac} - C_i}\left(\alpha_{(x,\omega)}\right) = \max_{x = (jh, kh) \in G_h} \max_{\theta \in [-\pi, \pi)^2} \rho \left( \frac{\gamma_y}{\gamma_x + \alpha_{j,k}^{(0)} \gamma_y} \sum_{i=2}^{L+1} \alpha_{j,k}^{(i-1)} C_i \right) \leq \max_{x = (jh, kh) \in G_h} \max_{\theta \in [-\pi, \pi)^2} \rho \left( \frac{\gamma_y}{\gamma_x + \alpha_{j,k}^{(0)} \gamma_y} \sum_{i=2}^{L+1} \alpha_{j,k}^{(i-1)} \rho(C_i) \right).$$

The spectral radius $\rho(C_i)$ can be bounded in terms of the roots of the univariate orthogonal polynomials $\psi_i$ in (2.11), as explained in [25, 23]. This shows that the bounds on $\rho(C_i)$ and $\rho_{\text{Jac} - C_i}$ depend asymptotically only on the polynomial chaos order $P$, and not on the number of random variables $L$. The latter follows from the monotone decay of KL-coefficients $\alpha_{j,k}^{(i)}$, $i = 1, \ldots, L+1$.

In Table 4.3, the effect of the type of polynomial chaos on the convergence is demonstrated: Legendre polynomials result in lower convergence factors overall than Hermite polynomials.

Remark 4.5. Table 4.3 shows the independence of the convergence on the grid spacing $h$ and the dependence on the number of random variables, the chaos order, and the variance. The dependence on the chaos order is worse in case of a Hermite chaos than in case of a Legendre chaos. This property is related to the infinite range of Hermite polynomials, compared to the boundedness of Legendre polynomials to the interval $[-1, 1]$, as explained in [23].

Table 4.4 shows convergence factors for Gauss–Seidel and block Gauss–Seidel $C_i$-splitting iterations. Numerical convergence factors are given to illustrate the accuracy of the analysis.

Remark 4.6. The more expensive block Gauss–Seidel $C_i$-splittings do not improve the convergence rate in comparison to Gauss–Seidel $C_i$-splitting iterations. The same result holds when comparing Jacobi $C_i$-splittings to block Jacobi $C_i$-splittings.

Table 4.5 shows that the circulant $C_i$-approximation leads to worse convergence factors than the Gauss–Seidel $C_i$-splitting.
4.2. Multigrid methods in the spatial dimension. The convergence properties of multigrid with a basic splitting (3.2) or a $K_z$-splitting smoother (3.3) have been analyzed in [8, 25]. The multigrid method has very favorable convergence properties: $h$- and $L$-independent convergence and, depending on the differential operator [23], also $P$-independent convergence. In [25] quantitative convergence results were given for the case of a $K_z$-splitting smoother. Below, we provide results for the basic splitting (3.2) and the $C_z$-splitting (3.4) smoother.

LFA considers the action of the two-grid operator on the Fourier grid modes (4.4). To this end, the Fourier space is divided into subspaces spanned by four harmonics, $\mathcal{H}(\theta, z) := \text{span}[\phi(\theta^1, z) \phi(\theta^2, z) \phi(\theta^3, z) \phi(\theta^4, z)]$, for a given $(\theta_x, \theta_y) \in \left[-\frac{\pi}{L}, \frac{\pi}{L}\right]$. The problems are invariant under the fine and coarse grid discrete differential operators, and under certain smoothing operators. The action of a smoothing operator

| Table 4.4 | Effect of using the Gauss–Seidel versus block Gauss–Seidel $C_z$-splitting strategy on the theoretical ($\rho_{\text{theo}}$) and numerical ($\rho_{\text{num}}$) convergence factors for problem (4.2), with lognormal field $\alpha_{\text{log}}(x, \omega)$. The discretization uses $h = 2^{-3}$, $L = 4$, $P = 2$, $\sigma = 0.2$, $L_c = 1$. In each row, one of these parameters is changed. |
|-------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| | Gauss–Seidel | Block GS |
| Grid spacing $h$ | $2^{-4}$ | $2^{-5}$ | $2^{-6}$ | $2^{-7}$ | $2^{-4}$ | $2^{-5}$ | $2^{-6}$ | $2^{-7}$ |
| $\rho_{\text{theo}} = 10^{-2}x$ | 7.070 | 7.112 | 7.123 | 7.125 | 8.205 | 8.261 | 8.272 | 8.276 |
| $\rho_{\text{num}} = 10^{-2}x$ | 6.695 | 6.656 | 6.734 | 6.763 | 7.793 | 7.853 | 7.885 | 7.885 |
| Random variables $L$ | 1 | 5 | 8 | 10 | 1 | 5 | 8 | 10 |
| $\rho_{\text{theo}} = 10^{-2}x$ | 7.128 | 7.058 | 7.058 | 7.007 | 7.128 | 8.479 | 8.688 | 8.854 |
| $\rho_{\text{num}} = 10^{-2}x$ | 6.749 | 6.448 | 6.475 | 6.591 | 6.734 | 8.089 | 8.426 | 8.575 |
| Order PC-basis $P$ | 1 | 3 | 4 | 5 | 1 | 3 | 4 | 5 |
| $\rho_{\text{theo}} = 10^{-2}x$ | 2.802 | 11.80 | 16.52 | 21.12 | 3.263 | 13.64 | 19.01 | 24.19 |
| $\rho_{\text{num}} = 10^{-2}x$ | 2.659 | 11.22 | 15.58 | 20.31 | 3.051 | 12.91 | 18.21 | 23.33 |
| Standard deviation $\sigma$ | 0.1 | 0.4 | 0.6 | 0.7 | 0.1 | 0.4 | 0.6 | 0.7 |
| $\rho_{\text{theo}} = 10^{-2}x$ | 1.855 | 24.32 | 43.81 | 52.64 | 2.170 | 27.59 | 48.33 | 57.30 |
| $\rho_{\text{num}} = 10^{-2}x$ | 1.681 | 23.49 | 43.03 | 51.71 | 2.015 | 26.36 | 47.03 | 56.33 |
| Correlation length $L_c$ | 0.25 | 0.5 | 2.5 | 5 | 0.25 | 0.5 | 2.5 | 5 |
| $\rho_{\text{theo}} = 10^{-2}x$ | 3.182 | 5.279 | 8.543 | 9.085 | 5.017 | 6.921 | 9.114 | 9.394 |
| $\rho_{\text{num}} = 10^{-2}x$ | 2.619 | 4.935 | 8.110 | 8.720 | 4.636 | 6.392 | 8.745 | 9.135 |

| Table 4.5 | Effect of using the circulant $C_z$-splitting iteration (3.4) on the theoretical ($\rho_{\text{theo}}$) convergence factors for problem (4.2). The random field $\alpha(x, \omega)$ corresponds to a lognormal field $\alpha_{\text{log}}(x, \omega)$ and is discretized by a Hermite chaos expansion. The discretization uses by default $h = 2^{-3}$, $L = 4$, $P = 2$, $\sigma = 0.2$, $L_c = 1$. Each row corresponds to changing one of these parameters. |
|-------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| | Gauss–Seidel | Block GS |
| Grid spacing $h$ | $2^{-4}$ | $2^{-5}$ | $2^{-6}$ | $2^{-7}$ | $2^{-4}$ | $2^{-5}$ | $2^{-6}$ | $2^{-7}$ |
| $\rho_{\text{theo}} = 10^{-1}x$ | 3.226 | 3.251 | 3.253 | 3.253 | 3.226 | 3.251 | 3.253 | 3.253 |
| Random variables $L$ | 1 | 5 | 8 | 10 | 1 | 5 | 8 | 10 |
| $\rho_{\text{theo}} = 10^{-1}x$ | 1.778 | 3.432 | 3.543 | 3.607 | 1.778 | 3.432 | 3.543 | 3.607 |
| Order PC-basis $P$ | 1 | 3 | 4 | 5 | 1 | 3 | 4 | 5 |
| $\rho_{\text{theo}} = 10^{-1}x$ | 1.968 | 4.644 | 6.102 | 7.363 | 1.968 | 4.644 | 6.102 | 7.363 |
| Standard deviation $\sigma$ | 0.1 | 0.4 | 0.6 | 0.7 | 0.1 | 0.4 | 0.6 | 0.7 |
| $\rho_{\text{theo}} = 10^{-1}x$ | 1.553 | 1.057 | > 1 | > 1 | 1.553 | 1.057 | > 1 | > 1 |
| Correlation length $L_c$ | 0.25 | 0.5 | 2.5 | 5 | 0.25 | 0.5 | 2.5 | 5 |
| $\rho_{\text{theo}} = 10^{-1}x$ | 2.475 | 2.955 | 3.381 | 3.414 | 2.475 | 2.955 | 3.381 | 3.414 |
on an element of such a space can be described by regrouping \( \tilde{S}_{h}^{j,k}(\theta) \), as defined in section 4.1, into a \((4Q \times 4Q)\)-matrix \( \tilde{S}_{h}^{j,k}(\theta) \):

\[
\tilde{S}_{h}^{j,k}(\theta) = \text{diag}(\tilde{S}_{h}^{j,k}(\theta^1), \tilde{S}_{h}^{j,k}(\theta^2), \tilde{S}_{h}^{j,k}(\theta^3), \tilde{S}_{h}^{j,k}(\theta^4)).
\]

With \( \tilde{L}_{h}^{j,k}(\theta) \) defined by (4.5), we can represent the action of the differential operator \( L_{h} \) by

\[
\tilde{L}_{h}^{j,k}(\theta) = \text{diag}(\tilde{L}_{h}^{j,k}(\theta^1), \tilde{L}_{h}^{j,k}(\theta^2), \tilde{L}_{h}^{j,k}(\theta^3), \tilde{L}_{h}^{j,k}(\theta^4)).
\]

On the coarse grid, \( L_{2h} \) is constructed by discretizing (4.1) with a standard five-point finite difference scheme on a rectangular grid with grid spacing \( 2h \). Its action can be represented by

\[
\tilde{L}_{2h}^{j,k}(\theta) = -\frac{1}{h^2} (\sin^2(\theta_x) I_{Q} + \sin^2(\theta_y) M_{j,k}).
\]

The prolongation operator maps the mode \( \phi(2\theta, z) \) onto \( \mathcal{H}(\theta, z) \) [32]. It is characterized by

\[
\tilde{P}_{2h}^{h}(\theta) = \tilde{P}_{2h}^{h}(\theta) \otimes I_{Q},
\]

with, in the case of bilinear interpolation, \( \tilde{P}_{2h}^{h}(\theta) \) given by

\[
\tilde{P}_{2h}^{h}(\theta) = \frac{1}{4} \left[ \begin{array}{c} (1 + \cos(\theta_x))(1 + \cos(\theta_y)) \\ (1 + \cos(\theta_x))(1 - \cos(\theta_y)) \\ (1 - \cos(\theta_x))(1 + \cos(\theta_y)) \\ (1 - \cos(\theta_x))(1 - \cos(\theta_y)) \end{array} \right] .
\]

Using standard coarsening, the restriction operator maps the space \( \mathcal{H}(\theta, z) \) onto the single mode \( \phi(\theta^1, z) \). The corresponding Fourier representation satisfies

\[
\tilde{P}_{2h}^{h}(\theta) = (\tilde{P}_{2h}^{h}(\theta))^T.
\]

To conclude, the action of the two-grid operator is characterized by

\[
\tilde{T}_{G_{h}}^{j,k}(\theta) = (\tilde{S}_{h}^{j,k}(\theta))^{\nu_2} \left( I_{4Q} - \tilde{P}_{2h}^{h}(\theta) \left( \tilde{L}_{2h}^{j,k}(\theta) \right)^{-1} \tilde{P}_{2h}^{h}(\theta) \tilde{L}_{h}^{j,k}(\theta) \right) (\tilde{S}_{h}^{j,k}(\theta))^{\nu_1},
\]

with \( \nu_1 \) and \( \nu_2 \) the number of presmoothing and postsmoothing steps, respectively, and \( I_{4Q} \in \mathbb{R}^{4Q \times 4Q} \) an identity matrix. Under the assumption that the variation of the coefficients \( a_i \) is sufficiently smooth [15, section 8.2.2], the asymptotic convergence factor of the two-grid scheme is defined as

\[
\rho_{TG} = \max_{x=(j^h,k^h) \in G_{h}} \max_{\theta \in [-\pi/2, \pi/2]} \frac{1}{2} \theta \left( \tilde{T}_{G_{h}}^{j,k}(\theta) \right).
\]

Table 4.6 illustrates the multigrid convergence in case of a standard (3.2) and collective (3.3) Gauss–Seidel smoother. The numerical results demonstrate the accuracy of the analysis in the former case. Numerical results for the \( K_s \)-splitting smoother can be found in [25].

Remark 4.7. The theoretical convergence factors for both smoothers are very similar. The more expensive \( K_s \)-splitting smoother does not yield significant faster convergence. Moreover, the convergence factors are even higher in case of a lognormal field.

Remark 4.8. In contrast to Table 4.1, the results for a Gaussian and a lognormal field differ significantly. The convergence in the latter case is about twice as slow as in the former case.
The theoretical ($\rho_{\text{theo}}$) and numerical ($\rho_{\text{num}}$) convergence factors for the two-grid iteration (4.11) with Gauss–Seidel smoother (3.2) applied to problem (4.2), with $a(x, \omega)$ a Gaussian field $a_k(x, \omega)$. Between brackets, the results are given for the lognormal case $a_k(x, \omega)$. Two pre-smoothing steps and one post-smoothing step are applied. Theoretical ($\rho_{\text{theo}}^\log$) convergence factors for the two-grid iteration with a collective Gauss–Seidel (3.3) smoother. The discretization uses $h = 2^{-5}$, $L = 4$, $P = 2$, $\sigma = 0.2$, $L_c = 1$. In each row, one of these parameters is changed.

<table>
<thead>
<tr>
<th>Grid spacing $h$</th>
<th>$2^{-4}$</th>
<th>$2^{-5}$</th>
<th>$2^{-6}$</th>
<th>$2^{-7}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\rho_{\text{theo}}^L$</td>
<td>1.228 (2.874)</td>
<td>1.228 (2.874)</td>
<td>1.228 (2.874)</td>
<td>1.228 (2.871)</td>
</tr>
<tr>
<td>$\rho_{\text{theo}}^{\log}$</td>
<td>1.860 (2.306)</td>
<td>1.860 (2.306)</td>
<td>1.860 (2.306)</td>
<td>1.860 (2.306)</td>
</tr>
<tr>
<td>$\rho_{\text{num}}^L$</td>
<td>1.742</td>
<td>1.739</td>
<td>1.750</td>
<td>1.752</td>
</tr>
<tr>
<td>Random variables $L$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\rho_{\text{theo}}^L$</td>
<td>1.220 (2.799)</td>
<td>1.230 (2.889)</td>
<td>1.233 (2.903)</td>
<td>1.233 (2.913)</td>
</tr>
<tr>
<td>$\rho_{\text{theo}}^{\log}$</td>
<td>1.746 (2.302)</td>
<td>1.883 (2.309)</td>
<td>1.905 (2.309)</td>
<td>1.922 (2.311)</td>
</tr>
<tr>
<td>$\rho_{\text{num}}^L$</td>
<td>1.505</td>
<td>1.719</td>
<td>1.738</td>
<td>1.786</td>
</tr>
<tr>
<td>Order PC-basis $P$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\rho_{\text{theo}}^L$</td>
<td>1.196 (2.495)</td>
<td>1.288 (3.199)</td>
<td>1.397 (3.488)</td>
<td>1.827 (3.750)</td>
</tr>
<tr>
<td>$\rho_{\text{theo}}^{\log}$</td>
<td>1.307 (2.242)</td>
<td>2.473 (2.370)</td>
<td>3.153 (2.435)</td>
<td>3.905 (2.500)</td>
</tr>
<tr>
<td>$\rho_{\text{num}}^L$</td>
<td>1.247</td>
<td>2.255</td>
<td>2.889</td>
<td>3.597</td>
</tr>
<tr>
<td>Standard deviation $\sigma$</td>
<td>0.1</td>
<td>0.4</td>
<td>0.6</td>
<td>0.7</td>
</tr>
<tr>
<td>$\rho_{\text{theo}}^L$</td>
<td>1.193 (2.394)</td>
<td>1.996 (3.978)</td>
<td>7.704 (5.194)</td>
<td>$&gt;1$ (5.804)</td>
</tr>
<tr>
<td>$\rho_{\text{theo}}^{\log}$</td>
<td>1.216 (2.183)</td>
<td>4.226 (2.810)</td>
<td>9.019 (3.668)</td>
<td>$&gt;1$ (4.212)</td>
</tr>
<tr>
<td>$\rho_{\text{num}}^L$</td>
<td>1.106</td>
<td>3.961</td>
<td>8.524</td>
<td>$&gt;1$</td>
</tr>
<tr>
<td>Correlation length $L_c$</td>
<td>0.25</td>
<td>0.5</td>
<td>2.5</td>
<td>5</td>
</tr>
<tr>
<td>$\rho_{\text{theo}}^L$</td>
<td>1.207 (2.646)</td>
<td>1.219 (2.785)</td>
<td>1.235 (2.930)</td>
<td>1.237 (2.948)</td>
</tr>
<tr>
<td>$\rho_{\text{theo}}^{\log}$</td>
<td>1.530 (2.211)</td>
<td>1.725 (2.265)</td>
<td>1.948 (2.338)</td>
<td>1.977 (2.351)</td>
</tr>
<tr>
<td>$\rho_{\text{num}}^L$</td>
<td>1.177</td>
<td>1.223</td>
<td>1.291</td>
<td>1.302</td>
</tr>
</tbody>
</table>

Remark 4.9. The convergence results are independent of the spatial and stochastic discretization parameters. There is a small dependency on the chaos order, because of the asymmetry of the differential operator (4.1), as explained in [23]. The multigrid methods have an optimal convergence behavior for the basic and the collective Gauss–Seidel smoother.

Table 4.7 presents convergence factors for the two-grid iteration (4.11) with a Collective Gauss–Seidel smoother. The discretization uses $h = 2^{-5}$, $L = 4$, $P = 2$, $\sigma = 0.2$, $L_c = 1$. In each row, one of these parameters is changed.

Remark 4.10. Despite the fast convergence of multigrid with a Gauss–Seidel smoother, the convergence rate decreases substantially for a large chaos order or variance. The robustness of multigrid is lost with the Gauss–Seidel smoother. The multigrid convergence is similar to the convergence of the smoother; see Table 4.3.

4.3. Multilevel methods in the stochastic dimension. For the LFA, we shall consider the two-level variant of Algorithm 1, using a polynomial chaos of order $p$ and $p - 1$. Applying this to mode (4.4) results in $TL\tilde{\phi}_{j,k}(\theta, z) = \tilde{T}L_{p}^{j,k}(\theta)\phi_{j,k}(\theta, z)$, with $\tilde{T}L_{p}^{j,k}(\theta)$ defined as

$$(4.12)\quad \tilde{T}L_{p}^{j,k}(\theta) = (\tilde{S}^{j,k}_{p}(\theta))^{\nu_{2}} \left[ I_{Q_{1}} \quad 0 \right] (\tilde{L}_{h}^{p-1}(\theta))^{-1} \left[ I_{Q_{1}} \quad 0 \right] \tilde{L}_{h}^{p}(\theta) (\tilde{S}^{j,k}_{p}(\theta))^{\nu_{1}}.$$
Theoretical (\(\rho_{\text{theo}}\)) and numerical (\(\rho_{\text{num}}\)) convergence factors for the two-grid iteration (4.11) with a Gauss-Seidel \(C_1\)-splitting iteration as smoother (3.4). Two pre-smoothing steps and one post-smoothing step are applied. The random field \(a(x, \omega)\) is modeled as \(a_{\text{log}}(x, \omega)\). The discretization uses by default \(h = 2^{-5}\), \(L = 4\), \(P = 2\), \(\sigma = 0.2\), \(L_c = 1\). In each row, one of these parameters is changed.

<table>
<thead>
<tr>
<th>Grid spacing</th>
<th>2^{-4}</th>
<th>2^{-5}</th>
<th>2^{-6}</th>
<th>2^{-7}</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\rho_{\text{theo}} = 10^{-3}\times)</td>
<td>4.587</td>
<td>4.587</td>
<td>4.587</td>
<td>4.587</td>
</tr>
<tr>
<td>(\rho_{\text{num}} = 10^{-3}\times)</td>
<td>3.669</td>
<td>3.817</td>
<td>3.781</td>
<td>3.864</td>
</tr>
<tr>
<td>Random variables (L)</td>
<td>1</td>
<td>5</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>(\rho_{\text{theo}} = 10^{-5}\times)</td>
<td>4.239</td>
<td>4.939</td>
<td>4.939</td>
<td>5.130</td>
</tr>
<tr>
<td>(\rho_{\text{num}} = 10^{-5}\times)</td>
<td>3.616</td>
<td>4.067</td>
<td>4.123</td>
<td>4.351</td>
</tr>
<tr>
<td>Order PC-basis (P)</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>(\rho_{\text{theo}} = 10^{-3}\times)</td>
<td>0.8779</td>
<td>11.71</td>
<td>22.16</td>
<td>35.61</td>
</tr>
<tr>
<td>(\rho_{\text{num}} = 10^{-3}\times)</td>
<td>0.7372</td>
<td>9.802</td>
<td>19.29</td>
<td>32.09</td>
</tr>
<tr>
<td>Standard deviation (\sigma)</td>
<td>0.1</td>
<td>0.4</td>
<td>0.6</td>
<td>0.7</td>
</tr>
<tr>
<td>(\rho_{\text{theo}} = 10^{-3}\times)</td>
<td>0.3108</td>
<td>54.49</td>
<td>180.3</td>
<td>262.6</td>
</tr>
<tr>
<td>(\rho_{\text{num}} = 10^{-3}\times)</td>
<td>0.2457</td>
<td>49.93</td>
<td>160.2</td>
<td>245.0</td>
</tr>
<tr>
<td>Correlation length (L_c)</td>
<td>0.25</td>
<td>0.5</td>
<td>2.5</td>
<td>5</td>
</tr>
<tr>
<td>(\rho_{\text{theo}} = 10^{-3}\times)</td>
<td>0.8842</td>
<td>2.776</td>
<td>6.298</td>
<td>7.014</td>
</tr>
<tr>
<td>(\rho_{\text{num}} = 10^{-3}\times)</td>
<td>0.8083</td>
<td>2.229</td>
<td>5.497</td>
<td>6.318</td>
</tr>
</tbody>
</table>

Theoretical (\(\rho_{\text{theo}}\)) and numerical (\(\rho_{\text{num}}\)) convergence factors for the two-level (TL) and multi-level (ML) V-cycle variant of Algorithm 1 applied to (4.2), with \(\nu_1 = \nu_2 = 1\). The discretization uses by default \(h = 2^{-5}\), \(L = 4\), \(P = 2\), \(\sigma = 0.2\), and \(L_c = 1\). In each row, one of these parameters is changed.

<table>
<thead>
<tr>
<th>Grid spacing (h)</th>
<th>(\alpha_{\text{log}}(x, \omega))—Hermite chaos</th>
<th>(\alpha_{\text{log}}(x, \omega))—Legendre chaos</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\rho_{\text{theo}} = 10^{-3}\times)</td>
<td>2^{-4}</td>
<td>2^{-5}</td>
</tr>
<tr>
<td>(\rho_{\text{num,ML}} = 10^{-3}\times)</td>
<td>3.551</td>
<td>3.556</td>
</tr>
<tr>
<td>Random variables (L)</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>(\rho_{\text{theo}} = 10^{-3}\times)</td>
<td>2.803</td>
<td>2.871</td>
</tr>
<tr>
<td>(\rho_{\text{num,ML}} = 10^{-3}\times)</td>
<td>2.814</td>
<td>2.904</td>
</tr>
<tr>
<td>Order PC-basis (P)</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>(\rho_{\text{theo}} = 10^{-3}\times)</td>
<td>3.034</td>
<td>3.889</td>
</tr>
<tr>
<td>(\rho_{\text{num,TL}} = 10^{-3}\times)</td>
<td>2.615</td>
<td>3.033</td>
</tr>
<tr>
<td>(\rho_{\text{num,ML}} = 10^{-3}\times)</td>
<td>2.650</td>
<td>3.063</td>
</tr>
<tr>
<td>Standard deviation (\sigma)</td>
<td>0.1</td>
<td>0.4</td>
</tr>
<tr>
<td>(\rho_{\text{theo}} = 10^{-3}\times)</td>
<td>0.2390</td>
<td>43.39</td>
</tr>
<tr>
<td>(\rho_{\text{num,TL}} = 10^{-3}\times)</td>
<td>0.1818</td>
<td>36.47</td>
</tr>
<tr>
<td>(\rho_{\text{num,ML}} = 10^{-3}\times)</td>
<td>0.1803</td>
<td>37.99</td>
</tr>
<tr>
<td>Correlation length (L_c)</td>
<td>0.25</td>
<td>0.5</td>
</tr>
<tr>
<td>(\rho_{\text{theo}} = 10^{-3}\times)</td>
<td>1.034</td>
<td>2.308</td>
</tr>
<tr>
<td>(\rho_{\text{num,TL}} = 10^{-3}\times)</td>
<td>0.5651</td>
<td>1.745</td>
</tr>
<tr>
<td>(\rho_{\text{num,ML}} = 10^{-3}\times)</td>
<td>0.6020</td>
<td>1.679</td>
</tr>
</tbody>
</table>

a \(C_1\)-splitting method, \(\mathcal{S}_p^{j,k}(\theta)\) is given by (4.9). The asymptotic convergence factor is then determined by the spectral radius of (4.12),

\[
\rho_{TL} = \max_{x \in (j_h,k_h) \in G_h} \max_{\theta \in [-\pi, \pi]^2 \sigma} \rho(TL_p^{j,k}(\theta)).
\]

Table 4.8 illustrates the convergence properties for a Gauss-Seidel \(C_1\)-splitting operator \(S_p^{j,k}\).
Table 4.9

Theoretical convergence factors for the two-level variant of Algorithm 1 applied to (4.2). The random field $\alpha(x, \omega)$ is modeled as a Gaussian field $\alpha$, with variance 0.04 and correlation length 1, and is discretized by a KL-expansion. A GSLEX-iteration (3.2), a GS $K_i$-splitting iteration (3.3), or a GS $C_i$-splitting iteration (3.4) is used in steps 1 and 5 of Algorithm 1, with $\nu_1 = \nu_2 = 1$.

<table>
<thead>
<tr>
<th>L</th>
<th>P</th>
<th>Basic GS (3.2)</th>
<th>GS $K_i$-splitting (3.3)</th>
<th>GS $C_i$-splitting (3.4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>2</td>
<td>0.9811</td>
<td>0.9811</td>
<td>4.918 $\cdot 10^{-3}$</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>0.9811</td>
<td>0.9811</td>
<td>26.86 $\cdot 10^{-3}$</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>0.9811</td>
<td>0.9811</td>
<td>7.283 $\cdot 10^{-3}$</td>
</tr>
</tbody>
</table>

Remark 4.11. Algorithm 1 combined with a $C_i$-splitting iteration has the same convergence properties as a $C_i$-splitting iteration (3.4) stand-alone: the convergence is independent of $h$, but depends on $L$, $P$, and $\sigma$. Comparing Table 4.8 to Table 4.3 shows the similar convergence behavior.

Remark 4.12. The convergence results for the multilevel variant of Algorithm 1 are well predicted by the analysis for the two-level case, as illustrated by the numerical convergence factors in Table 4.8.

Table 4.9 shows the theoretical convergence factors of the two-level algorithm for various choices of the iterative solver in steps 1 and 5.

Remark 4.13. As the convergence behavior of Algorithm 1 strongly depends on the method used in its first and last steps, we observe in Table 4.9 that only $C_i$-splitting iterations lead to satisfying convergence results. Note that these results deteriorate for large polynomial chaos orders.

5. Numerical experiments. In this section, we consider two nontrivial examples. The experiments use unstructured finite element meshes for the spatial discretization. The multigrid method is therefore of the algebraic multigrid (AMG) type. The AMG prolongators are built with classical Ruge–Stüben AMG [28]. In all numerical experiments, we iterate until the Euclidean norm of the relative residual, $\|r\|/\|b\|$, is smaller than $10^{-10}$.

The basic iterative methods (3.2) and $K_i$-splitting iterations (3.3) are used as smoothers for the multigrid method. The $C_i$-splitting iterations (3.4) can be used as stand-alones, as preconditioners for CG, or in combination with the multilevel approach of Algorithm 1. The various methods used in our tests are summarized in Table 5.1. An approximate method corresponds to solving the $(N \times N)$-block systems inexactly with one $V(1,1)$-AMG cycle.

5.1. Problem 1: Random diffusion coefficient. We consider a steady-state diffusion problem with random diffusion coefficient, defined on an L-shaped domain $D$ (see Figure 5.1):

\begin{align}
-\nabla \cdot (a(x, \omega)\nabla u(x, \omega)) &= b(x, \omega), & x \in D, \omega \in \Omega, \\
u(x, \omega) &= g(x, \omega), & x \in \partial D, \omega \in \Omega, \\
\frac{\partial u(x, \omega)}{\partial n} &= h(x, \omega), & x \in \partial D_N, \omega \in \Omega.
\end{align}

Dirichlet boundary conditions are imposed on the upper and lower right boundaries, and zero Neumann conditions are imposed elsewhere. We shall model $a(x, \omega)$ as a piecewise random field, consisting of three parts, $a_1(x, \omega)$, $a_2(x, \omega)$, and $a_3(x, \omega)$, defined, respectively, on the domains $D_1$, $D_2$, and $D_3$. 

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.
Table 5.1
Iterative methods and abbreviations.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>block-GS</td>
<td>Gauss–Seidel $C_r$-splitting iterations (3.4)</td>
</tr>
<tr>
<td>CG-GS</td>
<td>CG with symmetric Gauss–Seidel $C_r$-splitting iterations (3.4)</td>
</tr>
<tr>
<td>CG-Mean</td>
<td>CG with an approximate mean-based preconditioner (3.5)</td>
</tr>
<tr>
<td>CG-GSA</td>
<td>CG preconditioned by approximate symmetric Gauss–Seidel $C_r$-splitting iterations (3.4)</td>
</tr>
<tr>
<td>CG-CircA</td>
<td>CG preconditioned by approximate circulant $C_r$-splitting iterations (3.4)</td>
</tr>
<tr>
<td>MG-Point</td>
<td>CG with W(1,1)-multigrid cycles (section 3.2) and GS-smoother (3.2)</td>
</tr>
<tr>
<td>MG-Block</td>
<td>CG with W(1,1)-multigrid cycles (section 3.2) and collective GS smoother (3.3)</td>
</tr>
<tr>
<td>p-GS</td>
<td>Algorithm 1 with V(1,1)-cycles</td>
</tr>
<tr>
<td>p-GSA</td>
<td>CG preconditioned by one V(1,1)-cycle of Algorithm 1</td>
</tr>
<tr>
<td>p-GS</td>
<td>CG preconditioned by V(1,1)-cycles of Algorithm 1 with approximate GS</td>
</tr>
</tbody>
</table>

First, we discretize each part of $a(x, \omega)$ by a KL-expansion,

\begin{equation}
    a_j(x, \omega) \approx a_{j,1}(x) + \sum_{i=1}^{L_j} a_{j,i+1}(x)\xi_{j,i}(\omega) \quad \text{with} \quad x \in D_j \quad \text{and} \quad j \in \{1, 2, 3\}.
\end{equation}

We assume that all random variables $\xi_{j,i}$, with $i = 1, \ldots, L_j$ and $j \in \{1, 2, 3\}$, are independent and uniformly distributed on $[-1, 1]$. The solution is expanded with Legendre polynomials. The KL-expansions are based on an exponential covariance function $C_{aa}$ with variance $\sigma^2$ and correlation length $L_c$,

\begin{equation}
    C_{aa}(x, x') = \sigma^2 \exp\left(-\frac{\|x - x'\|_1}{L_c}\right).
\end{equation}

From (5.4), it follows that the conditions of Corollaries 2.4 and 2.6 are satisfied. As default configuration, we shall use the following parameters: $a_{1,1}(x) = 30$, $a_{2,1}(x) = 5$, $a_{3,1}(x) = 100$, $l_{c,1} = 1$, $l_{c,2} = 0.5$, $l_{c,3} = 2.5$, $\sigma_1^2 = 100$, $\sigma_2^2 = 2.25$, $\sigma_3^2 = 900$, $L_1 = 3$, $L_2 = 5$, and $L_3 = 2$. Figures 5.1(b) and 5.1(c) show the mean and the variance of the solution for these settings.

Figure 5.2 (top) shows the number of iterations required to solve the discretized system by the methods given in Table 5.1. The left column shows the iteration counts for one particular choice of the discretization parameters, denoted by “default.” In
the other columns, one of these parameters is changed in order to illustrate the effect of, respectively, the number of spatial nodes, the chaos order, the number of random variables, and the coefficient of variation on the convergence. Only the coefficient of variation and the number of spatial nodes slightly influence the iteration counts. This relatively stable convergence behavior is typical for the use of Legendre polynomials, as already reported in the context of Tables 4.3 and 4.8. By inexacty solving the block systems in CG-Mean, CG-GSA, CG-CircA, the theoretical independence, which was reported in section 4.1.3, of the convergence on the size of the spatial discretization is lost. In Figure 5.2 (bottom) relative solution times are given, where “relative” is with respect to the time required by the fastest solver for the particular problem. This shows that the mean-based preconditioning approach (CG-Mean) results in the lowest solution times, despite its larger iteration counts.

In section 3.3, we mentioned that Algorithm 1 can be combined with multigrid. Steps 1 and 5 then consist of one multigrid cycle. This combination does not, however, improve the multigrid convergence significantly, as illustrated by Table 5.2. We consider multigrid $V(1,1)$- and $W(1,1)$-cycles with a Gauss–Seidel smoother (3.2). The multigrid iterations are used stand-alone, as preconditioners for CG (corresponding to “MG-Point”), or in combination with $V(1,1)$-cycles of Algorithm 1. The latter case can also be accelerated by CG. The combination of multigrid with Algorithm 1 leads to a high convergence rate, but suffers from a very large time-per-iteration cost. Multigrid $W(1,1)$-cycles accelerated by CG generally result in the lowest solution times.

Second, we model each part of $a(\mathbf{x}, \omega)$ as a lognormal random field. That is, $a_i(\mathbf{x}, \omega) = \exp(a_{g,i}(\mathbf{x}, \omega))$, with $a_{g,i}(\mathbf{x}, \omega)$ a Gaussian field, given by expansion (5.4)
Table 5.2

Number of iterations required to solve (5.1)–(5.3) by a multigrid based method, with \( a(x, \omega) \) a discontinuous random field discretized by a KL-expansion with 10 random variables, either normally or uniformly distributed. The total solution time in seconds is given between brackets. The finite element mesh consists of 12154 nodes.

<table>
<thead>
<tr>
<th>Method</th>
<th>Hermite polynomials</th>
<th>Legendre polynomials</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( P = 1 )</td>
<td>( P = 3 )</td>
</tr>
<tr>
<td>V(1,1)-cycle</td>
<td></td>
<td></td>
</tr>
<tr>
<td>stand-alone</td>
<td>129 (74.5)</td>
<td>150 (3033)</td>
</tr>
<tr>
<td>with Algorithm 1</td>
<td>25 (39.8)</td>
<td>27 (1413)</td>
</tr>
<tr>
<td>with CG</td>
<td>30 (17.4)</td>
<td>38 (744)</td>
</tr>
<tr>
<td>with CG and</td>
<td>13 (20.8)</td>
<td>21 (1105)</td>
</tr>
<tr>
<td>Algorithm 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>W(1,1)-cycle</td>
<td></td>
<td></td>
</tr>
<tr>
<td>stand-alone</td>
<td>30 (27.4)</td>
<td>37 (1013)</td>
</tr>
<tr>
<td>with Algorithm 1</td>
<td>8 (16.9)</td>
<td>7 (543)</td>
</tr>
<tr>
<td>with CG</td>
<td>14 (12.8)</td>
<td>18 (497)</td>
</tr>
<tr>
<td>with CG and</td>
<td>7 (14.9)</td>
<td>7 (542)</td>
</tr>
<tr>
<td>Algorithm 1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Method default \( N = 53156 \) \( P = 1 \) \( L = 20 \) \( Q = 11 \) \( Q = 231 \) \( \sigma = 1.4/0.5/2.3 \)

1. block-GS 126 115 30 125 439
2. CG-GS 24 23 13 24 42
3. CG-Mean 184 221 74 203 471
4. CG-GSA 55 64 29 57 99
5. CG-CircA 181 210 68 198 465
6. MG-Block 13 13 13 14 13
7. p-GS 53 48 16 56 199
8. CG-p-GS 17 17 9 17 32
9. CG-p-GSA 31 >500 27 38 90

Fig. 5.3. Number of iterations (table) and relative solution times (figure) required to solve (5.1)–(5.3), with \( a(x, \omega) \) a discontinuous lognormal random field. The column of the table labeled “default” corresponds to \( N = 12154 \), \( P = 2 \), \( L = 10 \), \( Q = 66 \), and \( \frac{\sigma}{\mu} = 0.3 \). The other columns show the effect of modifying one of these parameters.

with normally distributed random variables and \( a_{1,1}^e(x) = 3 \), \( a_{2,1}^e(x) = 1 \), \( a_{3,1}^e(x) = 5 \), \((\sigma_1^e)^2 = 0.81\), \((\sigma_2^e)^2 = 0.09\), and \((\sigma_3^e)^2 = 2.25\). Each part of \( a(x, \omega) \) is now discretized by a Hermite chaos expansion [12]. The chaos order used for \( a(x, \omega) \) equals twice the polynomial chaos order used for the solution. Figure 5.3 illustrates the convergence rates and the relative solution times of the different solvers, analogously presented as in Figure 5.2. Only multigrid with a block smoother shows robust convergence. This robustness is also confirmed by LFA in Table 4.6 and in [23]. The convergence of the
other methods suffers from a high polynomial chaos order and a large coefficient of variation. These dependencies also appeared in the LFA results and were illustrated in Tables 4.3 and 4.5 for the variation. Despite the good LFA results for multigrid with GS smoother in Table 4.6, in this case, it fails to converge to the required tolerance within acceptable time. As a consequence, no results are shown in Figure 5.3 for the method “MG-Point” since its convergence rate deteriorates after only a few iteration steps.

5.2. Problem 2: Random domain. As the second problem, we consider a deterministic problem defined on a stochastic domain,

\begin{align}
-\nabla^2 u(x, \omega) &= 1 \quad \forall x \in \Omega(\omega) = \{ (x_1, x_2) | 0 \leq x_1 \leq 1, 0 \leq x_2 \leq s(x_1, \omega) + t \}, \\
\frac{\partial u}{\partial n} &= 0 \quad \forall x \in \partial \Omega(\omega).
\end{align}

This example is based on a similar problem presented in [39]. The upper boundary of the domain is perturbed by a zero mean random process, $s(x_1, \omega)$, with exponential covariance function. In the computational examples, we discretize $s(x_1, \omega)$, with variance $\sigma^2 = 0.01$ and correlation length $L_c = 1$, by an 8-term KL-expansion $y(x_1, \xi)$. We assume that the random variables are independent and uniformly distributed on $[-1, 1]$. Some samples of the random domain are shown in Figure 5.4 for $t = 1$, together with a cross-section of the mean and variance of the solution at $x_2 = 0.75$.

A deterministic PDE defined on a random domain can be transformed into a stochastic PDE on a deterministic domain by a numerical mapping approach [31] or by an analytical coordinate transformation [39, 38]. Denoting the coordinates of the transformed domain \( \mathbf{E} \) by $z = (z_1, z_2)$ and the coordinates of the original domain $\mathbf{D}(\omega)$ by $x = (x_1, x_2)$, we shall apply the transformation,

\[ z_1 = x_1 \quad \text{and} \quad z_2 = \frac{x_2}{t + y(x_1, \xi)}, \]

to map $\mathbf{D}(\omega)$ to $\mathbf{E} = \{ (z_1, z_2) | 0 \leq z_1 \leq 1, 0 \leq z_2 \leq t \}$. Problem (5.6)–(5.7) is then transformed into finding $v(z, \xi)$ such that

\begin{align}
-\alpha(z_1, \xi) \frac{\partial^2 v}{\partial z_1^2} &= \beta(z_1, \xi) \frac{\partial^2 v}{\partial z_1 \partial z_2} - \gamma(z_1, \xi) \frac{\partial v}{\partial z_2} - \frac{\partial^2 v}{\partial z_2^2} = \alpha(z_1, \xi), \quad z \in \mathbf{E}, \\
v(z, \xi) &= 0, \quad z \in \partial \mathbf{E},
\end{align}

with $\alpha(z_1, \xi) = t^2 + 2ty(z_1, \xi) + y(z_1, \xi)^2$. 

Fig. 5.4. (a) Four realizations of the random domain. (b)–(c) Mean and variance of the steady-state solution of (5.6)–(5.7) at $x_2 = 0.75$. The solution is discretized by a second-order Legendre chaos.
ITERATIVE SOLVERS FOR SFEM DISCRETIZATIONS

<table>
<thead>
<tr>
<th>Method</th>
<th>default</th>
<th>N ≈ 10^5</th>
<th>P = 5</th>
<th>L = 16</th>
<th>σ = 0.25</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.block-GS</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>6</td>
<td>9</td>
</tr>
<tr>
<td>2.CG-GS</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>3.CG-Mean</td>
<td>25</td>
<td>36</td>
<td>27</td>
<td>26</td>
<td>30</td>
</tr>
<tr>
<td>4.CG-GSA</td>
<td>20</td>
<td>28</td>
<td>17</td>
<td>20</td>
<td>21</td>
</tr>
<tr>
<td>5.CG-CircA</td>
<td>26</td>
<td>36</td>
<td>28</td>
<td>27</td>
<td>30</td>
</tr>
<tr>
<td>6.MG-Point</td>
<td>18</td>
<td>19</td>
<td>18</td>
<td>18</td>
<td>20</td>
</tr>
<tr>
<td>7.MG-Block</td>
<td>18</td>
<td>19</td>
<td>18</td>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td>8.p-GS</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>9.CG-p-GS</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
<th>N = 106370</th>
<th>P = 5, (1287)</th>
<th>L = 16, (153)</th>
<th>σ = 0.25</th>
</tr>
</thead>
</table>

Fig. 5.5. Number of iterations (table) and relative solution times (figure) required to solve (5.8).

The partial derivatives in (5.10) and (5.11) can easily be evaluated via the KL-expansion of \( y(z_1, \xi) \). The functions \( \alpha \), \( \beta \), and \( \gamma \) in (5.9)–(5.11) are expressed in terms of \( L \) uniformly distributed random variables \( \xi_i \) and products of the form \( \xi_i \xi_j \).

The stochastic Galerkin FEM is then applied to (5.8), with \( v(z, \omega) \) replaced by a Legendre chaos expansion. Note that for this problem the conditions of Corollaries 2.4 and 2.6 are not fulfilled.

Figure 5.5 illustrates the convergence and efficiency of the solvers of Table 5.1 applied to the discretization of the stochastic PDE (5.8). Method CG-p-GSA is not shown, as it does not reach convergence within 100 iterations for the test problems. The use of Legendre polynomials results in a fast convergence of Algorithm 1, as predicted by LFA in Table 4.8. The multigrid methods maintain their robust convergence, but result in higher computational times than \( C_i \)-splitting based methods.

6. Conclusion. This paper discusses iterative solvers for stochastic finite element discretizations. One-level methods, based on matrix splitting techniques, multigrid methods, and a multilevel algorithm based on a hierarchy in the stochastic discretization are addressed. A local Fourier analysis is presented to provide quantitative convergence results. Of the one-level methods only \( C_i \)-splitting methods yield good convergence rates. The other one-level methods should be used in multigrid context. In that case, both basic smoothers and \( K_i \)-splitting smoothers result in comparable
convergence properties. The convergence of multilevel methods that apply a coarsening in the stochastic dimension is very similar to the convergence of the $C_l$-splitting methods on which the multilevel algorithm is based. The convergence properties were illustrated on two test problems. The experiments compared the efficiency and robustness of the solvers. Multigrid with $K_l$-splitting smoother is the most robust method, but for low-dimensional problems its computational time is substantially higher than, for example, that of a simple mean-based preconditioning approach.

REFERENCES


