Adaptive Algebraic Multigrid for Sequence of Problems with Slowly Varying Random Coefficients

D. Kalchev, C. Ketelsen, P. S. Vassilevski

October 8, 2012

SIAM Journal on Scientific Computing

ADAPTIVE ALGEBRAIC MULTIGRID FOR SEQUENCE OF PROBLEMS WITH SLOWLY VARYING RANDOM COEFFICIENTS

D. KALCHEV [*], C. KETELSEN [*], AND P. S. VASSILEVSKI [*]

Abstract. This paper proposes an adaptive strategy for reusing previously constructed hierarchy of coarse spaces by algebraic multigrid, to construct a multilevel solver for a problem with nearby characteristics. A main target application is the linear problems that appear throughout a sequence of Markov Chain Monte Carlo simulations of subsurface flow with uncertain permeability field. The efficacy of the method is demonstrated with extensive set of numerical experiments.

AMS subject classifications. 65F10, 65N30, 65N55

Key words. iterative methods; multigrid; algebraic multigrid; adaptive algebraic multigrid; Monte Carlo; Markov chain Monte Carlo; subsurface flow

1. Introduction.

We are interested in solving sequences of linear systems originating from finite-element discretization of elliptic partial differential equations (PDEs) where the entries of the matrix change gradually throughout the sequence. Such sequences arise naturally in the iterative solution of nonlinear PDE [7], topology optimization in material design [15] and the quantification of uncertainty in physical systems via the Markov chain Monte Carlo (MCMC) method [18].

Of particular interest is the case of uncertainty quantification in subsurface flow simulation when prior geological models of the hydraulic conductivity field must be conditioned to observed data [10]. The MCMC method is used to sample from the posterior distribution, but requires the numerical solution of many thousands of PDEs with gradually changing highly heterogeneous diffusion coefficient. Furthermore, the solver must be robust enough to handle higher-order discretizations and anisotropic media with coefficients varying several orders of magnitude over small length scales. The necessary linear system solves make up the majority of the computational cost of such simulations [10]. As such, robust and efficient solvers are vital to efficient implementations of simulations.

The proposed method attempts to use a multilevel hierarchy developed for a previous linear system, to quickly build a new hierarchy that can be effectively used to solve a linear system with similar coefficient. It does this by drawing on well established techniques from the multilevel solver literature. As in the element-based algebraic multigrid (AMGe) method [2],[12], the proposed method uses local coarse problems naturally defined by the global finite element space to obtain coarse basis vectors which accurately capture local features of the problem. By incorporating ideas from smoothed aggregation (SA) multigrid

[*]Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, P.O. Box 808, L-560, Livermore, CA 94551. *email:* **{kalchev1, ketelsen1, vassilevski1}@llnl.gov**. This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 (LLNL-JRNL-589132).

([19]),the developed coarse space has good convergence properties independent of the large contrast in the background coefficient [6]. In adapting the old hierarchy for the new problem, the now ubiquitous adaptive AMG methodology (see [3],[4], and in the context of element-based AMG, or AMGe, see [21], [14], [20]) is used to automatically identify precisely the modes that should be incorporated into the new solver. The notion of developing general strategies to solve many realizations from a large parameter-dependent family of problems can be addressed by dimension reduction approaches. Such methods exploit an off-line setup phase to construct a reduced basis for the entire parameter space. Then in the simulation phase, the solver draws on the previously computed reduced basis to quickly produce coarse hierarchies suitable for the current problem realization. For recent studies in this direction, with emphasis on building coarse-scale models, we refer to [11] (and the references therein). Our proposed method differs from such schemes because it produces accurate enough coarse spaces on the fly, without the need for an offline setup. A systematic preliminary study of the presented adaptive solver strategies was performed in the master thesis of the first author, [13].

The remainder of the paper is organized as follows: In Section 2 we describe the model problem that we wish to solve and the finite-element discretization used to arrive at the linear system. We also describe a sequence of linear systems arising from a simplified MCMC subsurface flow simulation. In Section 3 we present background information on multigrid solvers and describe the two-level adaptive methodology that we utilize to efficiently solve the sequence of linear systems. In Section 4 we present a series of numerical experiments based on MCMC simulation which demonstrate the effectiveness of the proposed method. Finally, we make some concluding remarks in Section 5.

2. Background.

2.1. Model Problems. The simplest model problem with close analogy to subsurface flow is the single-phase steady-state flow equations given by

$$\begin{aligned} \mathbf{q}(\mathbf{x},\omega) + k(\mathbf{x},\omega)\nabla p(\mathbf{x},\omega) &= \mathbf{g}(\mathbf{x}) \text{ in } D\times\Omega, \\ \nabla\cdot\mathbf{q}(\mathbf{x},\omega) &= f(\mathbf{x}) \text{ in } D\times\Omega, \end{aligned} \tag{2.1}$$

subject to suitable boundary conditions. Here, $\mathbf{q}$ is the Darcy flux, k is the hydraulic conductivity field, and p is the pressure head. The variation of the PDE coefficient is parametrized by ω, where ω can be a single parameter or a vector of parameters (i.e. $\omega = [\omega_1, \ldots, \omega_M]$). In topology optimization ω may describe the shape of a solid material in a background void. Alternatively, in uncertainty quantification of subsurface flow, ω may represent a stochastic variable from some probability space Ω which describes the geological properties of the subsurface. Notice that since the pressure and flux depend on the conductivity field (nonlinearly through the solution of the PDE) p and $\mathbf{q}$ depend on parameter ω as well.

Re-writing (2.1) as a 2^{nd}-order elliptic equation, and prescribing appropriate boundary conditions, we have

$$\begin{aligned} -\mathrm{div}\left[k\left(\mathbf{x},\omega\right)\nabla p\left(\mathbf{x},\omega\right)\right] &= f\left(\mathbf{x}\right) - \nabla\cdot\mathbf{g}\left(\mathbf{x}\right) \text{ in } D\times\Omega \\ p\left(\mathbf{x}\right) &= g_{_D}(\mathbf{x}) \text{ on } \partial D_D \\ \nabla p\left(\mathbf{x}\right)\cdot\mathbf{n} &= g_{_N}(\mathbf{x}) \text{ on } \partial D_N \end{aligned} \tag{2.2}$$

In practice, the source term and boundary conditions can depend on parameter ω as well. For simplicity we do not consider such cases here but the proposed method can be easily extended to these cases.

Finally, we are particularly interested in the case when the conductivity field is highly anisotropic. In this case $k\left(\mathbf{x},\omega\right)$ can be written as a d-dimensional tensor, i.e. for the case $d=2$, we have

$$k\left(\mathbf{x},\omega\right) = \begin{bmatrix} k_{11}\left(\mathbf{x},\omega\right) & k_{12}\left(\mathbf{x},\omega\right) \\ k_{21}\left(\mathbf{x},\omega\right) & k_{22}\left(\mathbf{x},\omega\right) \end{bmatrix}. \tag{2.3}$$

2.2. Discretization. Consider the polygonal domain $D\subset\mathbb{R}^2$. Let $\mathcal{T}_h$ be a quasi-uniform triangulation of D with characteristic mesh size h and let $\{\tau\}$ be the collection of elements in $\mathcal{T}_h$. Let $\mathcal{V}_h$ be the finite element space associated with Lagrangian degrees of freedom $\mathcal{N}_h$. We assume that each fixed realization of conductivity field $k\left(\mathbf{x},\omega\right)$ is represented component-wise by piecewise-polynomial functions of degree one less than the polynomial basis in $\mathcal{V}_h$. As an example, if we take $\mathcal{V}_h$ to be the space of piecewise linear functions over $\mathcal{T}_h$, then the dofs in $\mathcal{N}_h$ coincide with vertices of the triangular elements, and $k\left(\mathbf{x},\omega\right)$ is represented by piecewise constants over each element.

For fixed ω, the bilinear form corresponding to model problem (2.3) is

$$a\left(u,v\right) = \int_D k\left(\mathbf{x},\omega\right)\nabla u\cdot\nabla v\,d\mathbf{x}. \tag{2.4}$$

Let φ_j be the basis function associated with dof j. The semidefinite fine-grid element stiffness matrix associated with element τ can be expressed as

$$\left[A_\tau\right]_{ij} = \int_\tau k\left(\mathbf{x},\omega\right)\nabla\varphi_j\cdot\nabla\varphi_i\,d\mathbf{x},\ \forall i,j\in\tau. \tag{2.5}$$

Finally, the global fine-grid stiffness matrix A can be defined and assembled according to

$$u^T A v = \sum_{\tau\in\mathcal{T}_h} u_\tau^T A_\tau v_\tau, \tag{2.6}$$

where u_τ represents restriction of global finite-element function u (or rather, its coefficient vector) to element τ.

2.3. Monte Carlo Simulation. In subsurface simulation the goal is to compute moments of some quantity of interest given reasonable assumptions on the uncertainty in the data. In a simple case this could mean assuming that the subsurface properties can be accurately modeled via a log-normal Gaussian field ([16]) and running a simulation to compute moments of the effective conductivity near some retrieval site.

Suppose that the conductivity field k belongs to probability distribution $\pi(k)$ and we wish to compute moments of some quantity of interest Q which, in the case of the model problem, is a function of conductivity field k and pressure p. The mean of the quantity of interest can be approximated using a Monte Carlo estimator

$$\widehat{Q}_{MC} = \frac{1}{N}\sum_{i=1}^{N} Q^{(i)}, \tag{2.7}$$

via Algorithm 2.3.

Algorithm 2.1 General Monte Carlo Estimator

```
**PROCEDURE**: $\widehat{Q}_{MC} \leftarrow \mathbf{MC}(N)$.
**INPUT**: Number of samples $N$.
**OUTPUT**: Mean of quantity of interest $\widehat{Q}_{MC}$.
for $i = 1, \dots, N$ do
  Draw sample $k^{(i)}$ from probability distribution $\pi(k)$.
  Solve model problem for $p^{(i)}$ with $k^{(i)}$ as coefficient.
  Compute quantity of interest $Q^{(i)} \equiv Q^{(i)}(p^{(i)}; k^{(i)})$.
end for
Compute $\widehat{Q}_{MC} = \frac{1}{N}\sum_{i=1}^{N} Q^{(i)}$.
```

The sampling procedure in Algorithm 2.3 depends greatly on the modeling choices made prior to beginning the simulation. A simple and popular choice is to model the conductivity field k as a log-normal random field with two-point correlation structure [16]. In this setting k is expanded in a so-called *truncated Karhunen-Loève Expansion (KLE)* as follows:

$$k(\mathbf{x}, \omega) = \exp\left[Y_M(\mathbf{x}, \omega)\right], \tag{2.8}$$

where

$$Y_M(\mathbf{x}, \omega) = Y_0(\mathbf{x}) + \sum_{j=1}^{M} \sqrt{\lambda_j}\phi_j(\mathbf{x})\,\xi_j(\omega)\,. \tag{2.9}$$

Here, M is the number of terms in the expansion, $Y_0(\mathbf{x})$ is the mean, and (λ_j, ϕ_j) are eigenpairs of the integral equation with the so-called *covariance function* $C(\mathbf{x}, \mathbf{x}')$ as the kernel, where

$$C(\mathbf{x}, \mathbf{x}') = \sigma_k^2 \exp\left(-\frac{|x_1 - x_1'|}{\gamma_1} - \frac{|x_2 - x_2'|}{\gamma_2}\right). \tag{2.10}$$

Here, σ_k^2 is the variance of the stochastic process and γ_j is the correlation length in the j^{th} direction. Larger values of σ_k^2 lead to larger jumps in the conductivity while smaller values of γ lead to higher frequency oscillations. The stochastic basis functions $\xi_j(\omega)$ are normal random variables with zero mean and unit variance.

For the form of covariance function C given in (2.10) there exist analytic expressions for the spatial basis functions ϕ [16]. Then, to draw a random sample from the desired probability distribution only requires generating M i.i.d. (independent and identically distributed) normal random variables ξ_j and evaluating $k(\mathbf{x}, \omega)$ according to (2.9)-(2.10).

Given a random sample $k^{(i)}$ obtained via the KLE, the remaining steps in Algorithm 2.3 are familiar. Next, the governing PDE is discretized based on the supplied conductivity data and the resulting linear system is solved. With solution in hand, the computation $Q^{(i)}$ requires the evaluation of some given functional of $p^{(i)}$ and $k^{(i)}$.

2.4. Markov Chain Monte Carlo Simulation. When distribution $\pi(k)$ is simple (e.g. log-normal with two-point correlation structure), the computation of $\widehat{Q}_{MC}$ is relatively straight-forward. However, major difficulties arise when we wish to incorporate dynamic data (e.g. observed pressure values or flow rates) in the distribution of k. It is in this case that Markov chain Monte Carlo (MCMC) methods become necessary, because they give a computationally feasible way of sampling from such complex (and not necessarily, explicitly given) distributions.

Suppose that, in general, the subsurface can be reliably modeled by some *prior* distribution $P(k)$, which is easy to sample from (e.g. log-normal). Suppose next that we are given some observed data F that we believe to be accurate, up to some level of measurement error. Then, in the Monte Carlo estimator given in Algorithm 2.4, we wish to draw samples of k from the prior distribution *conditioned* to the observed data. That is, we wish to sample from the conditional probability distribution $P(k|F)$. In our tests we will assume F comes from measurements of the pressure head at several fixed points in the domain. From Bayes' Law we have

$$P(k|F) \propto P(F|k)\, P(k)\,. \tag{2.11}$$

Here, $P(k)$ is the prior and $P(F|k)$ is a likelihood function describing the conditional probability that data F is observed given that k is the *true* conductivity field. We model the likelihood function as a Gaussian:

$$P(F|k) \propto \exp\left(-\frac{\|F-F_k\|^2}{\sigma_f^2}\right), \tag{2.12}$$

where σ_f^2 is the likelihood variance and F_k is the model response obtained by solving the forward problem with k as the fixed conductivity field. Finally, we can write the desired *posterior* distribution as

$$\pi(k) \;=\; P(k|F) \propto \exp\left(-\frac{\|F-F_k\|^2}{\sigma_f^2}\right) P(k)\,. \tag{2.13}$$

With the prescribed posterior, we can use any number of flavors of MCMC to generate samples from $\pi(k)$. For this paper we use the standard Metropolis-Hastings algorithm [18]. Additionally, we require a transition probability $q(k'|k)$ that describes a transition from field k to field k'. There are many choices for transition probability q. In this paper we will be primarily concerned with two, namely the independent sampler and the random walker.

The Metropolis-Hastings algorithm for general q is described in Algorithm 2.4.

Algorithm 2.2 Metropolis-Hastings MCMC

```
**PROCEDURE**: $k_{n+1} \leftarrow \mathbf{MH}(k_n)$.
**INPUT**: Current conductivity sample $k_n$.
**OUTPUT**: New conductivity sample $k_{n+1}$.
Generate proposal $k'$ from transition probability $q(k'|k_n)$.
Compute acceptance probability
    $\alpha(k_n, k') = \min\left(1, \frac{\pi(k')\,q(k_n|k')}{\pi(k_n)\,q(k'|k_n)}\right)$.
Take $k_{n+1} = k'$ with probability $\alpha(k_n, k')$, and $k_{n+1} = k_n$ with probability $1-\alpha(k_n, k')$.
```

After a sufficiently long *burn-in* phase, it can be shown (as is well-known, [18]) that samples $\{k_n\}$ come from the desired posterior distribution $\pi(k)$. These samples are then used in a Monte Carlo estimator to compute moments of quantities of interest. Note again that each evaluation of $\pi(k)$ requires solution of the forward model problem (2.3). For complicated applications this solve phase is the main computational bottleneck in the simulation. Furthermore, in the MCMC setting, the vast majority of proposals are rejected, greatly adding to the computational cost of sampling.

The performance characteristics of the method greatly depend on the choice of transition probability $q(k'|k)$. We will utilize two very different transition probabilities in our numerical experiments. The first is the so-called *random walker.* In this case each random coefficient in the KLE of $k(\mathbf{x}, \omega)$ is perturbed slightly by an i.i.d. normal random variable with zero mean and variance δ_k^2 (i.e., from $\mathcal{N}(0,\delta_k^2)$). That is, for $j = 1,\ldots,M$ draw η_j from $\mathcal{N}(0,\delta_k^2)$ and set

$$\xi_j' = \xi_j + \eta_j. \tag{2.14}$$

The size of the random walk is determined by the step-size parameter δ_k. For small choices of δ_k proposal configuration k' will be fairly similar to configuration k. The trade-off in the choice of step-size is that if k is a highly likely configuration and δ_k is small then it's probable that k' will also be a likely configuration and will be accepted in the Algorithm 2.4. On the other hand, if the step-size is too small then it will take a very long time for the MCMC algorithm to explore the stochastic space. If the step-size is too large, accepted realizations will quickly explore the stochastic space, but proposals are less likely to fit the conditioning data and thus are more likely to be rejected. In practice δ_k is tuned so that the MCMC acceptance rate is around $20-25\%$. There are many other transition probabilities similar to the random walker [18] that use additional gradient information to move in likely stochastic directions, but still take gradual steps. It is this class of algorithms that our adaptive solver methodology is targeted because the coefficients of the linear system should change gradually.

A more extreme transition probability is the so-called *independent sampler*. In this case the current conductivity sample is not used at all in the generation of the proposal. Proposed configuration k' is simply generated by a new random draw from the prior distribution $P(k)$. This method tends to sample the stochastic space very quickly, but has a very low acceptance rate because the chance that a randomly drawn configuration fits the conditioning data is very low. We use the independent sampler in our numerical experiments to demonstrate the difficulty of the linear solves when the elements of the sequence of linear systems are unrelated.

3. Solvers.

3.1. General Multigrid.

Multigrid methods are popular due to their potential to solve $N \times N$ sparse linear systems of equations of the form

$$A\mathbf{x} = b, \tag{3.1}$$

in $O(N)$ work. The efficiency of multigrid methods is due to two complementary processes: *relaxation* and *coarse-grid correction*. Relaxation is a process that efficiently eliminates oscillatory error in the approximate solution exploiting local updates. Coarse-grid correction is a global process which utilizes a coarser grid to eliminate the smooth error left untouched by relaxation. It does so by transferring the fine-grid residual to the coarse grid, and then resolving the smooth error components there by solving a smaller coarse-grid version of the equations. The coarse representation of the error is then interpolated to the fine grid and used to correct the fine-grid solution [7]. A general two-grid algorithm is given in Algorithm 3.1. Note that we have used the so-called *Galerkin* definition of the coarse grid operator, i.e. $A_c = P^T AP$. The TL Algorithm with supplied initial iterate x_0 provides an (SPD) mapping $b \mapsto B_{TG}^{-1} b = x_{TG} - x_0$.

Algorithm 3.1 Two-Level (TL) Algorithm

```
**PROCEDURE**: $x_{TG}$, $\rho_{TG} \leftarrow \mathbf{TL}\,(A, b, x_0, M, P, \nu_{TG})$
**INPUT**: Matrix $A$, vector $b$, initial iterate $x_0$, relaxation operator $M$,
           interpolation operator $P$, number of iterations $\nu_{TG}$.
**OUTPUT**: Two-grid iterate $x_{\mathrm{TG}} = x$.
Initialize: $x = x_0$.
for $i = 1, \ldots, \nu_{TG}$ do
  Pre-relax: $x \leftarrow x + M^{-1}(b - Ax)$.
  Correct: $x \leftarrow x + P\left(P^T AP\right)^{-1} P^T (b - Ax)$.
  Post-relax: $x \leftarrow x + M^{-T}(b - Ax)$.
end for
```

For a prescribed fine-grid linear system $Ax = b$, the TL Algorithm is completely defined by the choice of relaxation operator M and interpolation operator P. Relaxation is generally chosen such that the inverse of M is inexpensive to apply on the fine grid and be A-convergent. That is $\|I - A^{\frac{1}{2}} M^{-1} A^{\frac{1}{2}}\| < 1$ or equivalently $M + M^T - A$ is SPD.

Numerous choices for P exist and they separate the class of multigrid methods into several distinct categories. In the finite-element setting it is convenient to associate the columns of P with basis functions living on the coarse grid. In geometric multigrid [7] these coarse basis functions are simply the usual finite element basis functions associated with a coarser mesh. In classical AMG [1] the coarse basis is defined by associating a subset of fine-grid dofs with the coarse grid (called coarse-grid dofs) and choosing the entries of P such that constant functions are interpolated exactly. In smoothed aggregation (SA) multigrid ([19]) a tentative interpolation operator $\widehat{P}$ is constructed associating coarse-grid dofs with small groups of connected fine-grid dofs or *aggregates* and fixing the entries of $\widehat{P}$ so that constant functions are interpolated exactly. The final interpolation operator P is obtained by smoothing the columns of $\widehat{P}$ by a small number of steps of a suitable relaxation scheme. Finally, the spectral element-based AMG (ρAMGe) method (e.g., [8]-[9], or [20]) uses the element stiffness matrices to form local sub-problems on disjoint patches of elements (called *agglomerates*). In the ρAMGe the coarse space is associated with low-lying eigenmodes of the local operators, extended by zero to the global domain to form a tentative interpolant.

3.2. SA-ρAMGe. As the foundation of the proposed adaptive method, we begin with a two-level version of the algorithm presented and analyzed in [6] (see also [5]), which is a hybrid of the ρAMGe [9] and SA multigrid [19] methods. We briefly describe the method below, but for a thorough explanation and analysis the reader is directed to [6] and the references therein.

We begin by partitioning the set of fine-grid elements in $\mathcal{T}_h$ into a pre-specified number of disjoint non-overlapping sets $\{T\}$ of elements called *agglomerated elements* or simply *agglomerates.* We will sometimes refer to this set as $\mathcal{T}_H$, where the characteristic mesh size (diameter of $T \in \mathcal{T}_H$) is of order H. In addition, the set of fine-grid dofs $\mathcal{N}_h$ are partitioned into a corresponding number of disjoint, non-overlapping sets $\{\mathcal{A}\} = \{\mathcal{A}_i\}_{i=1}^{n_c}$ such that each $\mathcal{A}$ is contained in a unique agglomerate T. These sets $\{\mathcal{A}\}$ are called *aggregates.* Since aggregates must be non-overlapping it is necessary to arbitrate dofs lying on the shared boundary of two agglomerates. This arbitration is accomplished by assigning the shared dof to the neighboring aggregate to which it is most *strongly connected.* To be precise, let $a_{ij} = [A]_{ij}$ and define s_{ij} to be the measure of how strongly connected dof i is to dof j. One definition of such strength measure is given by

$$s_{ij} = \frac{|a_{ij}|}{\sqrt{a_{ii}a_{jj}}}. \tag{3.2}$$

Suppose now that dof i lies on the boundary between two or more agglomerates. Define the *aggregated neighborhood* of dof i as follows:

$$\mathcal{N}(i) = \{\, j \mid a_{ij} \neq 0 \text{ and } j \in \mathcal{A}_\ell \text{ for some } \ell\}.$$

We then find dof $j_{max} \in \mathcal{N}(i)$ such that

$$s_{ij_{max}} = \max_{j \in \mathcal{N}(i)} s_{ij}. \tag{3.3}$$

Finally, we assign dof i to aggregate $\mathcal{A}_\ell$ such that $j_{max} \in \mathcal{A}_\ell$. A schematic of the arbitration process can be seen in Figure 3.1.

For each agglomerate T, we assemble the local agglomerate stiffness matrix A_T, constructed using the fine-grid element stiffness matrices A_τ for $\tau \in T$. The local agglomerate stiffness matrix can be assembled in analogy with the assembly of the global stiffness matrix A. That is,

$$u_T^T A_T v_T = \sum_{\tau \in T} u_\tau^T A_\tau v_\tau. \tag{3.4}$$

For each T, we solve the following local generalized eigenvalue problem

$$A_T \mathbf{q}_k = \lambda_k D_T \mathbf{q}_k, \quad k = 1, \ldots, n_T, \tag{3.5}$$

where D_T is a diagonal matrix and n_T is the number of fine-grid dofs in T. For our experiments, the matrix D_T is taken to be the weighted ℓ_1-smoother corresponding to matrix

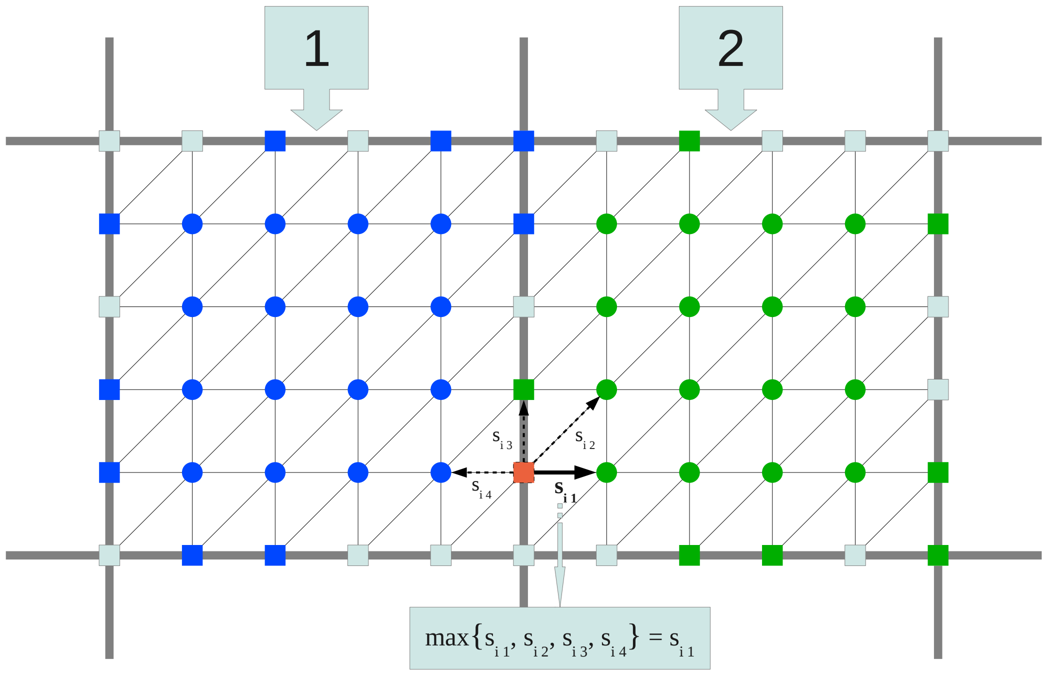

Fig. 3.1: Schematic of arbitration process. Dark grey borders indicate neighboring agglomerates. Filled circles indicate interior dofs naturally assigned to aggregates. Squares indicate dofs lying on boundary of agglomerates, with filled and empty squares representing already aggregated and unaggregated boundary dofs, respectively. Boundary dof i which is currently being arbitrated is highlighted in orange with arrows indicating aggregated dofs that dof i is strongly connected to. The bold arrow indicates that dof i is most strongly connected to dof 1 and thus dof i will be added to aggregate 2.

A_T. That is $D_T = \operatorname{diag}(d_i)_{i=1}^{n_T}$ where $d_i = \sum_{j=1}^{n} |a_{ij}| \sqrt{\frac{a_{ii}}{a_{jj}}}$ and $a_{ij} = [A_T]_{ij}$. Assuming that the eigenpairs are ordered such that $\lambda_1 \le \lambda_2 \le \ldots \le \lambda_{n_T}$ and for some prescribed tolerance θ, we select the first $m_T \le n_T$ eigenvectors such that $\lambda_k \le \theta \|D_T^{-\frac{1}{2}} A_T D_T^{-\frac{1}{2}}\|$ for all $k \le m_T$. We note that with the choice of D_T being the ℓ_1-smoother, we have the simple normalization $\|D_T^{-\frac{1}{2}} A_T D_T^{-\frac{1}{2}}\| \le 1$. For convenience, we collect these local eigenvectors in matrix $Q_T = [\mathbf{q}_1, \ldots, \mathbf{q}_{m_T}]$. We then restrict the columns of Q_T to aggregate $\mathcal{A}$ (where $\mathcal{A} \subset T$) by extracting only the rows of Q_T corresponding to the dofs in $\mathcal{A}$. These restricted vectors are stored in matrix $Q_{\mathcal{A}}$.

The columns of Q_T are linearly independent by construction, but their restriction to $\mathcal{A}$ may not be. By othogonalizing the columns of $Q_{\mathcal{A}}$ we arrive at the local tentative interpolant $\widehat{P}_{\mathcal{A}}$ whose $m_{\mathcal{A}} \le m_T$ columns span the column space of $Q_{\mathcal{A}}$. After computing the local tentative interpolants on each aggregate, the global tentative interpolant can be written as a block-diagonal matrix with the local tentative interpolants as diagonal blocks:

$$\widehat{P} = \begin{bmatrix} \widehat{P}_{\mathcal{A}_1} & 0 & & 0 \\ 0 & \widehat{P}_{\mathcal{A}_2} & & 0 \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & \dots & \widehat{P}_{\mathcal{A}_{n_c}} \end{bmatrix}. \tag{3.6}$$

Note that $\widehat{P}$ is an orthogonal matrix by construction. In order to make the interpolation operator P stable in the energy norm, the columns of $\widehat{P}$ are smoothed using an appropriate polynomial smoother. The final interpolation operator can then be written as

$$P = S\widehat{P}, \tag{3.7}$$

where matrix S is a polynomial in $D^{-1}A$. Here D is some diagonal matrix which is spectrally equivalent to he diagonal of A. In practice we choose the weighted ℓ_1-smoother corresponding to the global stiffness matrix A. The construction of interpolation operator P is summarized in Algorithm 3.2.

Algorithm 3.2 Construction of SA-ρAMGe Interpolation Operator

```
PROCEDURE: $P, \{Q_T\} \leftarrow$ BuildP $(A, \{T\}, \{\mathcal{A}\}, S, \theta)$
INPUT: Matrix $A$, agglomerates $\{T\}$, aggregates $\{\mathcal{A}\}$, and spectral tolerance $\theta$
OUTPUT: Interpolation operator $P$ and local coarse basis $\{Q_T\}$
for each agglomerate $T$ do
  Solve $A_T\mathbf{q}_k = \lambda_k D_T \mathbf{q}_k$ for $k = 1, \dots, n_T$
  Select $\mathbf{q}_k$ s.t. $\lambda_k \le \theta \| D_T^{-\frac{1}{2}} A_T D_T^{-\frac{1}{2}} \|$ for all $k \le m_T$ and set $Q_T = [\mathbf{q}_1, \dots, \mathbf{q}_{m_T}]$
  Form $Q_{\mathcal{A}}$ by restricting columns $Q_T$ to $\mathcal{A} \subset T$
  Orthogonalize columns of $Q_{\mathcal{A}}$ to obtain local tentative interpolate $\widehat{P}_{\mathcal{A}}$
end for
Construct global tentative interpolate $\widehat{P}$
Smooth columns of $\widehat{P}$, i.e. set $P = S\widehat{P}$
```

Remark 3.1. *The performance of the method greatly depends on judicious selection of the spectral tolerance θ. If θ is chosen very small then few vectors will be added to the coarse space. This may be sufficient for easy problems (e.g. constant coefficient Poisson) but for the problems of interest here (e.g. Poisson with high-contrast coefficients or anisotropies) a very small θ will cause the method to underperform. On the other hand, a θ that is too large will result in too many vectors being added to the coarse space. In this case the convergence properties of the TL Algorithm may be very good, but the large dimension of the coarse space* *will result in a very large coarse-grid problem which may be as costly to solve as the fine-grid problem.*

With the construction of P defined for the SA-ρAMGe method it remains to specify the relaxation schemes to be used in the TL Algorithm and for smoothing the columns of the tentative interpolant. It is typical in element-based AMG methods to choose large agglomerates to allow for faster coarsening and thus smaller coarse grid problems. The trade-off is that a powerful relaxation scheme is usually required on the fine grid to get acceptable convergence. For our purposes we choose polynomial smoothers because of their easy parallelization. We first define polynomial $s_\nu(t)$ used in SA-AMG (cf., e.g., [20]), i.e.

$$s_\nu(t) = (-1)^\nu \frac{1}{2\nu+1} \frac{T_{2\nu+1}\left(\sqrt{t}\right)}{\sqrt{t}}, \tag{3.8}$$

where $T_l(t)$ is the degree-l first-kind Chebyshev polynomial on the interval $[-1,+1]$. For the interpolation smoother we choose $S = s_{\nu_p}\left(b^{-1}\, D^{-1}A\right)$ where b is a real number satisfying $\|D^{-\frac{1}{2}}AD^{-\frac{1}{2}}\| \leq b = \mathcal{O}(1)$. For the relaxation smoothing we choose $M^{-1} = p_{\nu_r}\left(b^{-1}D^{-1}A\right)$ where

$$p_\nu(t) = \left(1 - T^2_{2\nu+1}(\sqrt{t})\right) s_\nu(t)\,. \tag{3.9}$$

Note that for interpolant smoothing parameter ν_p, smoothing a single column of $\widehat{P}$ requires approximately ν_p matrix-vector multiplies with global stiffness matrix A. On the other hand, application of the relaxation operator M^{-1} with parameter ν_r requires approximately $3\nu_r+1$ matrix-vector multiplies with A.

3.3. An Adaptive SA-ρAMGe Method. Let $A = A(k)$ be the global stiffness matrix obtained by discretizing the model problem with conductivity field k. Assume that we have constructed a two-level mapping $B = B_{TG}$ based on the SA-ρAMGe procedure described in Section 3.2. Then, in addition to interpolation operator P we also have agglomerates $\{T\}$, and sets of local coarse basis vectors $\{Q_T\}$.

Suppose now that we wish to solve a modified problem with $A' = A(k')$, where k' is similar to k, having come from e.g. random walk MCMC proposal. Rather than perform the costly initial setup phase again for the new matrix A' we wish to reuse the existing hierarchy, if possible. If k and k' are very similar the original hierarchy may be adequate for solving problems involving A'. We test this by replacing A by A' (and by extension, M by M') in the TL Algorithm and then iterating on the homogeneous equation $A'x = 0$ using the unmodified P from the previous hierarchy. That is, we perform ν_a steps of iteration

$$x_k \leftarrow \left(I - B^{-1}_{TG}A'\right) x_{k-1} \tag{3.10}$$

with random x_0. This, in effect, computes an approximation to the minimal eigenvector of $B_{TG}^{-1}A'$. We monitor the convergence $\|x_k\|_{A'} \mapsto 0$ for $k \mapsto \infty$ by computing, e.g.

$$\rho_k^2 = \frac{x_k^T A' x_k}{x_{k-1}^T A' x_{k-1}}, \qquad k = 1, 2, \dots . \tag{3.11}$$

The asymptotic convergence factor ρ_{TG} is approximated by taking the average over the last several values of ρ_k. Let ρ^{target} be some prescribed *acceptable* (or desired) convergence factor for the method. If $\rho_{\mathrm{TG}} \le \rho^{target}$ then we accept $B'_{TG} = B_{TG}$ as the new solver. If the convergence of the method is *not* adequate, then the current coarse space defined by P does not capture some components of the algebraically smooth error (or near null-vector of $B_{TG}^{-1}A'$) with respect to the modified system. Fortunately, the iterative solution of $A'x = 0$ using the current method automatically exposes the smooth error mode that the method cannot resolve, namely the vector x_k after ν_a iterations of the solver. We set $x^{\mathrm{bad}} = x_{\nu_a}$ and form its restriction $x_T^{\mathrm{bad}} = x^{\mathrm{bad}}|_T$ on every agglomerate T.

If the modified conductivity field k' does not differ too much from k then it may be the case that the previous coarse space on each agglomerate T is *nearly* sufficient for solution of the modified problem. We now attempt to build a coarse space based on the old local coarse space and augmented with x_T^{bad}. For each T, we solve the modified local generalized eigenvalue problem

$$A'_T \mathbf{z}_k = \mu_k D'_T \mathbf{z}_k, \quad k = 1, \dots, m_T + 1 \tag{3.12}$$

in the subspace Span $\left\{Q_T, x_T^{\mathrm{bad}}\right\}$. Note that this eigenproblem is solved in a *much* smaller subspace than in the non-adaptive algorithm described in Section 3.2. Then, for spectral tolerance θ, we select the first $m'_T \le m_T + 1$ eigenvectors such that $\mu_k \le \theta \|(D'_T)^{-\frac{1}{2}} A'_T (D'_T)^{-\frac{1}{2}}\|$ for all $k \le m'_T$. These vectors are then stored in matrix form as $Z_T = \left[\mathbf{z}_1, \dots, \mathbf{z}_{m'_T}\right]$.

We now proceed as in the non-adaptive algorithm with the new local eigenvectors Z_T. The next step is to restrict the columns of Z_T to the local aggregate dofs. Recall that in the selection of aggregates it was necessary to arbitrate those dofs lying on agglomerate boundaries. Since the coefficients have changed in A' the dof connections used to compute the strength of connection measure in the arbitration process have also changed. As such it is prudent to re-arbitrate the boundary dofs with respect to the coefficients of A'. We note that this computation is relatively inexpensive and does not greatly affect the data structures in the code. With new aggregates in hand we restrict the columns of Z_T to aggregates, orthogonalize the restricted vectors, build and smooth the global tentative interpolation operator to obtain the modified interpolation operator P'. This, together with the modified matrix A', defines the modified two-level mapping B'_{TG}.

The new hierarchy is tested by iterating on $A'x = 0$ and monitoring $\|x\|_{A'} \to 0$. If measured convergence factor ρ'_{TG} satisfies $\rho'_{TG} \le \rho^{target}$ then the new method is accepted and used in the solution of the forward problem for proposal conductivity field k'. If convergence is still unacceptable then the adaptive process is repeated until a method is obtained with the desired performance properties. If the convergence of the method fails to improve by some prescribed amount (e.g. $\rho'_{TG} \leq \delta_c \rho_{TG}$ with, say, $\delta_c = 0.9$) it may be necessary to increase the spectral tolerance θ for the next adaptive iteration. This will allow more vectors to be added to the coarse space and should improve convergence.

There are many strategies for increasing the spectral tolerance θ during the adaptive process. An attractive naive strategy may be to simply increase θ by some small constant factor (a factor of 2, say). For difficult problems this naive strategy can quickly lead to coarse spaces that are too large to be computationally efficient. The strategy that we employ is as follows. Recall that as a necessary step in the adaptive procedure we compute eigenvalues $\{\mu_k\}$ of the local problems for each agglomerate. Then, on each agglomerate T, we construct the coarse space using the m'_T eigenvectors with eigenvalues satisfying $\mu_k \leq \theta \|(D'_T)^{-\frac{1}{2}} A'_T (D'_T)^{-\frac{1}{2}}\|$ where $m'_T \leq m_T + 1$. The updated spectral tolerance θ' is computed as

$$\theta' = \frac{1}{n_c} \sum_{i=1}^{n_c} \mu^{(T_i)}_{\min(m'_{T_i}, m_{T_i})+1}. \tag{3.13}$$

This quantity is an average over two different values: On agglomerates T_i where some eigenvectors are not accepted into the coarse space, the eigenvalue of the smallest *rejected* mode, namely $\mu^{(T_i)}_{m'_{T_i}+1}$, is used in the average. This tells the algorithm that a spectral tolerance of this size would have allowed another potentially useful vector to be added the coarse space. On agglomerates where all $m_{T_i} + 1$ eigenvectors are accepted, the largest computed eigenvalue, $\mu^{(T_i)}_{m_{T_i}+1}$, is used in the average. This choice is made because if there were only a few agglomerates with rejected vectors the previous value of θ might be acceptable and another similar adaptive iteration will improve solely from the computation of a new smooth error component.

A major advantage of the proposed adaptive procedure is its ability to trim components of the coarse basis (i.e., to achieve *de-refinement*) that are no longer necessary for the current configuration. In the course of a MCMC simulation, thousands of configurations will be proposed. If the adaptive procedure simply augmented the previous coarse space with the new necessary basis vectors at every stage, the complexity of the solver would quickly grow to unacceptable levels. The spectral nature of the proposed method remedies this concern. If a coarse basis vector from the previous hierarchy is no longer necessary for the modified system then it will have an associated eigenvalue in the local eigenproblem greater than the spectral tolerance, and thus will be excluded from the new coarse basis. This keeps the size of the coarse basis in check over the course of the simulation.

It should be clear then that the choice of θ is exceedingly important. If θ is too small then the adaptive update may trim vectors from the coarse basis that would in fact be

Algorithm 3.3 Adaptation of Hierarchy to New Matrix

```
**PROCEDURE**: $P', \{Z_T\}, \theta' \leftarrow \mathbf{AdaptP}(A', \{T\}, \{Q_T\}, \nu_a, \theta, \rho^{target})$
**INPUT**: Matrix $A'$, agglomerates $\{T\}$, local coarse basis $\{Q_T\}$, spectral tolerance $\theta$, and threshold $\delta_c$ used to change $\theta$
**OUTPUT**: Interpolation operator $P'$, local coarse basis $\{Z_T\}$, spectral tolerance $\theta'$
Iterate on $A'x = 0$ with existing solver, i.e. $x, \rho_{TG} \leftarrow \mathbf{TL}(A', 0, x, M', P, \nu_a)$
**if** $\rho_{TG} \geq \rho^{target}$ **then**
  Reassign dofs to aggregates $\{\mathcal{A}\}$ based on $A'$
**end if**
**while** $\rho_{TG} \geq \rho^{target}$ **do**
  Set $x^{\text{bad}} = x_{\nu_a}$
  **for** each agglomerate $T$ **do**
    Set $x_T^{\text{bad}} = x^{\text{bad}}|_T$
    Solve $A'_T\mathbf{z}_k = \mu_k D'_T\mathbf{z}_k$ in Span$\left\{Q_T, x_T^{\text{bad}}\right\}$ for $k = 1, \ldots, m_T + 1$
    Select $\mathbf{z}_k$ s.t. $\mu_k \leq \theta\|(D'_T)^{-\frac{1}{2}}A'_T(D'_T)^{-\frac{1}{2}}\|$ for all $k \leq m'_T$ and set $Z_T = \left[\mathbf{z}_1, \ldots, \mathbf{z}_{m'_T}\right]$
    Form $Z_{\mathcal{A}}$ by restricting columns $Z_T$ to $\mathcal{A} \subset T$
    Orthogonalize columns of $Z_{\mathcal{A}}$ to obtain local tentative interpolate $\widehat{P}'_{\mathcal{A}}$
  **end for**
  Construct global tentative interpolate $\widehat{P}'$
  Smooth columns of $\widehat{P}'$, i.e. set $P' = S\widehat{P}'$
  Iterate on $Ax = 0$ with updated solver, i.e. $x, \rho'_{TG} \leftarrow \mathbf{TL}(A', 0, x, M', P', \nu_a)$
  **if** $\rho'_{TG} \geq \delta_c\,\rho_{TG}$ **then**
    Update spectral tolerance, i.e. compute $\theta'$ according to (3.13)
  **end if**
**end while**
```

useful for improving convergence. Conversely, if θ is too large vectors that were useful for the previous linear system may be retained (thus increasing the cost of the method) without a noticeable benefit to performance. The update scheme given in (3.13) and used in Algorithm 3.3 gives a method for increasing the value of θ when the current method is inadequate. However, this does not allow for the possibility that θ may be too large and could be decreased without hampering performance (in fact, if the coarse space contains many vectors that are not necessary, decreasing θ could improve overall performance.) We propose the following scheme for improving the initial value of θ when the adaptive update is called. Let θ_n be the value of θ at the end of the solve-phase for the n^{th} linear system. Then, if adaptation is necessary for the $(n+1)^{st}$ configuration, the adaptive algorithm is called with spectral tolerance θ_{n+1} given by

$$\theta_{n+1} = \kappa\,\theta_n + (1-\kappa)\,\widetilde{\theta}, \tag{3.14}$$

where $\widetilde{\theta}$ is the average over all $\{\theta_i\}_{i=1}^{n-1}$ and κ is a tuning parameter. This strategy allows θ to effectively remember where it has been. In the case that the sequence of linear systems has some members that are very difficult compared to the rest, it would be unfortunate if the method was automatically tuned to accommodate these outliers, and in the process became more costly than necessary for future typical configuration.

3.4. Practical Considerations. Suppose now that we wish to use the proposed adaptive procedure in a simulation where many thousands of slowly changing linear systems of the form $A(k)\,x = b$ must be solved. If the coefficients do not change too much it should be the case that the current solver is adequate to use in the solution of several consecutive systems in the sequence. According to Algorithm 3.3, rather than immediately attempting to solve a new system $A'x = b$ we should first test the current hierarchy by solving $A'x = 0$. In the case that the solver remains adequate for several systems in the sequence, this initial test will greatly increase the computational cost of the solver. We propose a more pragmatic strategy. For a new system $A'x = b$ we apply ν_0 iterations of the TL-Algorithm with the existing coarse space defined by P and right-hand side vector b. That is

$$x_{TG},\, \rho_0 \leftarrow \mathbf{TL}\left(A', b, x_0, M', P, \nu_0\right). \tag{3.15}$$

In this case we measure the convergence of the solver using the ratio of consecutive residuals. That is, we compute

$$\rho_k = \frac{\|r_k\|_{B_{TG}^{-1}}}{\|r_{k-1}\|_{B_{TG}^{-1}}} \qquad k = 1, \ldots, \nu_0, \tag{3.16}$$

and set ρ_0 equal to the average of the last several ρ_k's. For reasonable choices of ν_0 we can be confident that a good convergence factor measured during this initial test indicates that the solver is still adequate for the current linear system. The solve is then continued until the desired tolerance is met, at which point we have the desired solution and we can move on to the next coefficient. If it is the case that ρ_0 is not adequate the adaptive procedure is called and the current hierarchy is improved.

The final practical modification we make concerns the number of inner adaptive cycles used when adaptation of the hierarchy is required. In Algorithm 3.3 the convention is to continue adapting until the measured convergence falls below ρ^{target} and then return the updated hierarchy. There are a few instances where this rigid practice may hurt the efficiency of the method. In the setting of MCMC simulation it may be the case that a very pathological configuration may be proposed. In this case, the adaptive procedure may require many inner adaptive cycles before the desired performance is reached. After doing substantial work to adapt the method to suit this pathological case, it is likely that the proposal will be rejected by the MCMC algorithm, rendering the effort invested in the adaptation useless for future configurations. This case suggests the utility of setting a maximum number of inner adaptive cycles that the user is willing to commit to any one configuration. On the other hand, in MCMC simulation the situation occasionally arises when adaptation is called and adequate performance is restored by running a single inner adaptive cycle. Then, on the next proposed configuration, it may again be the case that adaptation is necessary and the performance threshold is again satisfied by running a single inner adaptive cycle. Although the overhead involved in the setup of the adaptive algorithm (e.g. re-arbitration of the aggregates) is minor, we would prefer to update a coarse hierarchy and have it remain viable for several more linear systems in the sequence. As such, it may be prudent to perform, say two, inner adaptive cycles, even if the convergence threshold is reached on the first, with the aim that the additional work will pay off in the long term. This scenario suggests the utility of setting minimum number of inner adaptive cycles per adaptation. The compromise between the two is, of course, to prescribe a fixed number of inner adaptive cycles, ν_f, to be performed whenever adaptation is necessary.

4. Numerical Experiments.

We test the proposed method in the setting of MCMC simulation of steady-state single-phase flow with uncertainty in the hydraulic conductivity field. The model problem with a mix of Dirichlet and Neumann boundary conditions is given in (4.1).

$$
\begin{aligned}
-\text{div}\left[k\left(\mathbf{x},\omega\right)\nabla p\left(\mathbf{x},\omega\right)\right] &= 0 && \text{in } [0,1]^2 \times \Omega \\
p\left(\mathbf{x}\right) &= 1 - x_1 && \text{on } x_1 = \{0,1\} \\
\nabla p\left(\mathbf{x}\right)\cdot\mathbf{n} &= 0 && \text{on } x_2 = \{0,1\}
\end{aligned} \tag{4.1}
$$

We test the method on the model problem with three increasingly challenging forms of k, corresponding to scalar, grid-aligned anisotropic, and non-grid-aligned anisotropic conductivity tensors. Furthermore, we test all cases on linear systems arising from both linear and quadratic finite-element discretizations using a uniform triangular fine-grid mesh. The uncertainty in the conductivity field is modeled using the definitions provided in Section 2.3. That is, we let $Y_M\left(\mathbf{x},\omega\right)$ be a Gaussian process with two-point correlation structure (see (2.10)) expanded in an M-term truncated KLE, (2.9). In our experiments we choose $M = 1000$, $Y_0 \equiv 0$, $\sigma_k^2 = 3$, and $\gamma_1 = \gamma_2 = 0.1$. In the case of scalar diffusion we model the conductivity as a log-normal, i.e.

$$
k_S(\mathbf{x},\omega) = \exp\left[Y_M\left(\mathbf{x},\omega\right)\right]. \tag{4.2}
$$

For the chosen set of parameters, realizations of k_S often vary as many as six orders of magnitude over small regions in the domain. Two sample realizations of Y_M with the given parameters are shown in Figure 4.1.

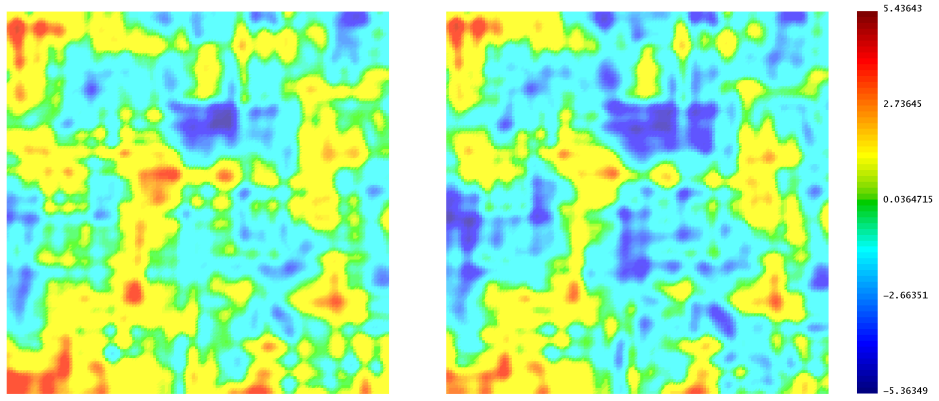

Table 4.1: Two realizations of the $Y_M(\mathbf{x}, \omega)$ based on the given model parameters.

The model problem becomes significantly more difficult in the case of anisotropic conductivity. As such, we model the component stochastic processes in the diffusion tensors using at most quadratic powers of Y_M. The grid-aligned (k_G) and non-grid-aligned (k_N) conductivity tensors are given as follows: Let $Y_{M,x} \equiv Y_{M,x}(\mathbf{x}, \omega)$ and $Y_{M,y} \equiv Y_{M,y}(\mathbf{x}, \omega)$ be realizations from the stochastic process described above (subscripts x and y indicate only that they are distinct from each other). Then the two anisotropic conductivity tensors are given as follows:

$$k_G(\mathbf{x}, \omega) = \begin{bmatrix} Y_{M,x}^2 + \epsilon & 0 \\ 0 & Y_{M,y}^2 + \epsilon \end{bmatrix}, \tag{4.3}$$

$$k_N(\mathbf{x}, \omega) = \begin{bmatrix} Y_{M,x}^2 + \epsilon & Y_{M,y}Y_{M,x} \\ Y_{M,x}Y_{M,y} & Y_{M,y}^2 + \epsilon \end{bmatrix}, \tag{4.4}$$

where ϵ is some small parameter which controls the strength of the anisotropy.

In the MCMC simulation, sampling is conditioned on pressure data at 9 fixed locations in the domain. For each problem formulation, the simulation is run using fixed likelihood variance σ_f^2 for both the random walker and independent sampler transition probabilities. The value of the random walk step-size δ_k (see (2.14)) is tuned so that the MCMC acceptance rate is between 20−25%. In the case of the random-walk transition probability we run several simulations for a total of 5000 MCMC iterations. For the indepdent sampler test we run one simulation with 2000 MCMC iterations. Each linear solve is considered complete when the SA-ρAMGe-preconditioned relative residual has been reduce by a factor of 10^{-8}.

The solver performance is measured via the average convergence factor $(\overline{\rho})$ over all simulation runs, where the two-level hierarchy is used as a stationary iterative method.

For practical purposes, we also solve each system with the two-level hierarchy used as a preconditioner for the conjugate gradient method (given in parentheses in the tables). As an indication of the cost of the two-level cycle the average operator complexity ($\overline{C_A}$) is reported, where C_A is a measure of the number of nonzeros in the operators of the two-level hierarchy versus the number of nonzeros in the fine-grid operator:

$$C_A = \frac{\mathrm{nnz}\,(A) + \mathrm{nnz}\,\left(P^T A P\right)}{\mathrm{nnz}\,(A)}. \tag{4.5}$$

We also report the percentage of configurations in the run for which adaptation is necessary (N_r) as well as the average number of inner adaptive cycles needed to improve convergence to acceptable levels ($\overline{N_b}$). Finally, the average value of the spectral tolerance ($\overline{\theta}$) over the course of the simulation is reported. In all cases θ is initialized to $\theta = 0.001$ at the beginning of the simulation. To contrast the current method with more traditional AMG methods, we have tested the two-level version of the classical Ruge-Stüben AMG method (as implemented in [17]) on represented samples from each simulation and reported the average convergence factors.

4.1. Scalar Conductivity Field. Table 4.1 displays the solver performance for simulations run with log-normal scalar conductivity field k_S. The likelihood variance was set to $\sigma_f^2 = .009$ for each MCMC simulation. When the random walk transition probability is used the random walk step-size is set to $\delta_k = .085$. The agglomerates are chosen such that the coarsening factor is approximately $H/h = 12$. This is very aggressive coarsening but it is suitable in this case because the coefficient is scalar. The smoothing parameters in the solver are set to $\nu_r = 1$ and $\nu_P = 4$. In the adaptive cycle, the smooth error mode is approximated by performing $\nu_a = 20$ stationary iterations with the current hierarchy on the homogeneous system. The convergence of the solver is considered adequate if it is faster than $\rho^{target} = 0.9$. Finally, when the random walker is used we fix the number of inner adaptive cycles at $\nu_f = 1$ as described in Section 3.4. Since consecutive linear systems in the independent sampler MCMC are unrelated we do not expect the adaptive update to be optimal. As such, we allow the adaptive method to do as many as $\nu_{max} = 5$ inner adaptive cycles, but allow it to exit early if the desired convergence is reached.

It is clear from Table 4.1 that, in the case of scalar conductivity with random walk MCMC, the basic method provides a rich enough course space that adaptation is rarely, if ever, necessary. This is true for both the linear and quadratic finite element test cases. Adaptation is also rarely necessary when the independent sampler is used with linear finite elements. When quadratic finite elements are used in conjunction with the independent sampler adaptation is required quite frequently, but when it is required a single inner adaptive cycle is sufficient to restore good convergence. The scalar conductivity diffusion problem is a fairly simple and can typically be solved with a traditional AMG method, though these methods are not guaranteed to remain adequate as the magnitude of the contrast increases.

Transition	Elem Type	n_f	$\overline{\rho}$	C_A	N_r	$\overline{N_b}$	$\overline{\theta}$
random walk	linear	16641	0.92 (0.53)	1.01	0.9%	1.00	0.001
random walk	linear	66049	0.89 (0.47)	1.01	0.0%	NA	0.001
random walk	quadratic	66049	0.92 (0.57)	1.01	1.2%	1.00	0.005
random walk	quadratic	263169	0.92 (0.57)	1.01	0.8%	1.00	0.005
independent	linear	16641	0.91 (0.52)	1.01	1.2%	1.08	0.002
independent	linear	66049	0.88 (0.48)	1.01	0.0%	NA	0.001
independent	quadratic	66049	0.92 (0.58)	1.02	58%	1.01	0.006
independent	quadratic	263169	0.92 (0.57)	1.02	38%	1.00	0.005

Table 4.2: Average results for simulation with **scalar** conductivity field. The MCMC acceptance rate using the random walker and independent sampler were 23% and 1%, respectively.

The two-level Ruge-Stüben AMG solver attained an average stationary convergence factor of approximately $\rho_{RS} = 0.05$ on both linear test cases. However, the traditional AMG solver performs very poorly in the quadratic case, achieving a convergence factor of only $\rho_{RS} = 0.99$. Clearly the proposed method is the better choice with higher-order elements, and although the proposed method is not directly aimed at simple scalar conductivity problems, it is likely that for large problem sizes and highly varying coefficients the proposed method would out-perform traditional AMG for linear finite elements as well.

4.2. Anisotropic Conductivity Field. Tables 4.2 and 4.3 display the solver performance for simulations with grid-aligned and non-grid-aligned anisotropic conductivity fields, respectively. In all cases the anisotropy parameter is set to $\epsilon = 0.001$. The likelihood variance is again set to $\sigma_f^2 = .009$. When the random walk transition probability is used the random walk step-size is set to $\delta_k = .06$. For the anisotropic problems the agglomerates are chosen with a less aggressive coarsening factor of $H/h = 6$. The smoothing parameters in the solver are set to $\nu_r = 2$ and $\nu_P = 2$. In the adaptive cycle, the smooth error mode is again approximated by performing $\nu_a = 20$ stationary iterations with the current hierarchy on the homogeneous system. Again the target convergence factor is set to $\rho^{target} = 0.9$. Finally, when the random walker is used we fix the number of inner adaptive cycles to $\nu_f = 3$. In the simulations using the independent sampler we again allow the adaptive method to do up to $\nu_{max} = 5$ inner adaptive cycles.

In the case of grid-aligned anisotropy we see that adaptation is again necessary only very rarely when linear finite elements and the random walker are used. The case of quadratic finite elements is more difficult, requiring adaptation approximately 25% of the time for the largest grid. However, when adaptation is necessary three inner adaptive iterations are sufficient to bring the performance back to acceptable levels. When the independent sampler is used it becomes necessary to adapt the hierarchy quite often. Not surprisingly, the simulations using the independent sampler require very frequent adaptation, especially

Transition	Elem Type	n_f	$\overline{\rho}$	C_A	N_r	$\overline{N_b}$	$\overline{\theta}$
random walk	linear	16641	0.87 (0.44)	1.03	0.1%	3.00	0.001
random walk	linear	66049	0.90 (0.47)	1.06	0.8%	3.00	0.005
random walk	quadratic	66049	0.92 (0.56)	1.01	3.4%	3.00	0.002
random walk	quadratic	263169	0.92 (0.58)	1.01	25%	3.00	0.001
independent	linear	16641	0.88 (0.46)	1.04	0.7%	1.54	0.021
independent	linear	66049	0.92 (0.51)	1.05	12%	1.47	0.045
independent	quadratic	66049	0.93 (0.56)	1.02	90%	1.37	0.008
independent	quadratic	263169	0.94 (0.51)	1.18	83%	2.74	0.043

Table 4.3: Average results for simulation with **grid-aligned anisotropic** conductivity field. The MCMC acceptance rate using the random walker and independent sampler were 24% and 1%, respectively.

when quadratic elements are used. However, on average less than two inner adaptive cycles are sufficient to restore good convergence. The traditional AMG method is able to achieve the reasonable convergence factor of $\rho_{RS} = 0.5$ for the linear finite-element case, but again degrades to $\rho_{RS} = 0.99$ for the quadratic elements.

Transition	Elem Type	n_f	$\overline{\rho}$	C_A	N_r	$\overline{N_b}$	$\overline{\theta}$
random walk	linear	16641	0.90 (0.48)	1.19	3.1%	3.0	0.005
random walk	linear	66049	0.89 (0.44)	1.48	7.2%	3.0	0.025
random walk	quadratic	66049	0.92 (0.55)	1.13	14%	3.0	0.004
random walk	quadratic	263169	0.94 (0.58)	1.21	65%	3.0	0.003
independent	linear	16641	0.90 (0.46)	1.23	100%	1.91	0.021
independent	linear	66049	0.89 (0.42)	1.42	100%	2.71	0.036
independent	quadratic	66049	0.96 (0.65)	1.12	100%	4.99	0.005
independent	quadratic	263169	0.98 (0.81)	1.14	100%	5.00	0.004

Table 4.4: Average results for simulation with **non-grid-aligned anisotropic** conductivity field. The MCMC acceptance rate using the random walker and independent sampler were 24% and 1%, respectively.

The non-grid-aligned anisotropic diffusion problem is notoriously difficult to solve. The two-level traditional AMG method used in our numerical experiments did not achieve a converge factor smaller than $\rho_{RS} = 0.99$ in any of the test cases. Even spectral-based AMGe methods typically require large coarse spaces and very powerful relaxation schemes to solve the problem with reasonable speed. The numerical experiments indicate that the proposed method can achieve reasonable convergence factors in the case of the random walk MCMC and requires very little adaptation when linear finite elements are used. When quadratic finite elements are used it must adapt for just over half of the configurations, but three inner adaptive iterations are sufficient to restore good performance. This provides considerable speed-up over the alternative of building new hierarchies from scratch for each realization of the conductivity field. Note that when the independent sampler is used adaptation becomes necessary for nearly every realization, and requires many more inner adaptive cycles to obtain good convergence. In fact, for quadratic finite-elements it would likely be more efficient to build the coarse hierarchy from scratch for each new conductivity field. This should not be surprising since the proposed method is designed specifically for the case when the background coefficient changes gradually, unlike in the independent sampler case when the consecutive configurations are completely independent.

5. Conclusion. In conclusion, we have presented a two-level adaptive framework, based on SA-ρAMGe multigrid, that allows for rapid solution of sequences of linear systems originating from discretization of elliptic PDE with slowly varying conductivity coefficient. For simple problems (e.g. scalar conductivity tensor and linear finite elements) the hierarchy is robust enough that it provides efficient solution of many linear systems where the conductivity may vary many orders of magnitude over small length scales. For difficult problems (e.g. anisotropic conductivity tensors and higher-order finite elements) the solver displays good performance for several consecutive linear systems. When performance degrades our method allows for rapid update of the two-level hierarchy which restores good performance at a fraction of the original setup cost. We demonstrated the effectiveness of the method in the context of Markov Chain Monte Carlo simulation of steady-state subsurface flow.

REFERENCES

[1] A. Brandt, S. McCormick, and J. Ruge, *Algebraic multigrid (AMG) for sparse matrix equations. in sparsity and its applications, D.J, Evans (ed.).*, (1984).

[2] M. Brezina, A. Cleary, R. Falgout, V. Henson, J. Jones, T. Manteuffel, S. McCormick, and J. Ruge, *Algebraic multigrid based on element interplation(AMGe)*, SIAM J. Sci. Comp, 22 (2000), pp. 1570–1592.

[3] M. Brezina, R. Falgout, S. MacLachlan, T. Manteuffel, S. McCormick, and J. Ruge, *Adaptive smoothed aggregation (αSA) multigrid.*, SIAM Rev., 47(2) (2005), pp. 317–346.

[4] ———, *Adaptive algebraic multigrid.*, SIAM J. Sci. Comp., 27(4) (2006), pp. 1261–1286.

[5] M. Brezina, C. Heberton, J. Mandel, and P. Vaněk, *An iterative method with convergence rate chosen a priori.* An earlier version presented at the 1998 Copper Mountain Conference on Iterative Methods, April 1998., April 1999.

[6] M. Brezina and P. Vassilevski, *Smoothed aggregation spectral element agglomeration AMG: $SA-\rho AMGe$*, in "Large-Scale Scientific Computing, 8th International Conference, LSSC 2011, Sozopol, Bulgaria, June 6-10th, 2011. Revised Selected Papers". Lecture Notes in Computer Science, vol. 7116, Springer, 2012, pp. 3–15.

[7] W. Briggs, V. Henson, and S. McCormick, *Multigrid Tutorial*, SIAM, 2000.

[8] T. Chartier, R. Falgout, V. E. Henson, J. Jones, T. Manteuffel, S. McCormick, J. Ruge, and P. S. Vassilevski, *Spectral AMGe (ρAMGe)*, SIAM Journal on Scientific Computing, 25 (2003), pp. 1–26.

[9] ———, *Spectral element agglomerate AMGe*, in Domain Decomposition Methods in Science and Engineering XVI, vol. 55 of Lecture Notes in Computational Science and Engineering, Springer, 2007, pp. 515–524.

[10] Y. Efendiev, A. Datta-gupta, V. Ginting, X. Ma, and B. Mallick, *An efficient two-stage markov chain monte carlo method for dynamic data integration*, Water Resources Research, 41, W12423 (2005).

[11] Y. Efendiev, J. Galvis, and F. Thomines, *A systematic coarse-scale model reduction technique for parameter-dependent flows in highly heterogeneous media and its applications*, Submitted, (2012).

[12] J. Jones and P. Vassilevski, *AMGe based on element agglomeration*, SIAM J. Sci. Comput., 23(1) (2001), pp. 109–133.

[13] D. Kalchev, *Adaptive algebraic multigrid for finite element elliptic equations with random coefficients*, Tech. Rep. LLNL-TR-553254, Lawrence Livermore National Laboratory, Livermore, California, USA, April 30 2012.

[14] I. Lashuk and P. S. Vassilevski, *On some versions of the element agglomeration AMGe method*, Numerical Linear Algebra with Applications, 15 (2008), pp. 595–620.

[15] B. Lazarov, R. Matzen, and O. Sigmund, *Efficient topology optimization in matlab using 88 lines of code*, Structural and Multidisciplinary Optimization, 43(1) (2011), pp. 1–16.

[16] O. Le Maître and O. Knio, *Spectral Methods for Uncertainty Quantification*, Springer: New York, NY, 2010.

[17] *PyAMG: Algenraic Multigrid Solvers in Python.* http://code.google.com/p/pyamg/.

[18] C. Robert and G. Casella, *Monte Carlo Statistical Methods, 2nd Ed.*, Springer: New York, NY, 2004.

[19] P. Vaněk, J. Mandel, and M. Brezina, *Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems.*, Computing, 56 (1996), pp. 179–196.

[20] P. Vassilevski, *Multilevel Block Factorization Preconditioners, Matrix-based Analysis and Algorithms for Solving Finite Element Equations*, Springer: New York, NY, 2008.

[21] P. Vassilevski and L. Zikatanov, *Multiple vector preserving interpolation mappings in algebraic multigrid.*, SIAM Journal on Matrix Analysis and Applications, 27 (2005-2006), pp. 1040–1055.